Beyond SIDH: A survey on countermeasures and new constructions

Valerio Ardizio

School of Computer and Communication Sciences

Semester Project

June 2024

Responsible	**Supervisor**
Prof. Serge Vaudenay	Laurane Marco
EPFL / LASEC	EPFL / LASEC

Contents

Introduction

Cryptography, the science of secure communications in the presence of an adversary, has a long history that dates back to ancient times when people already felt the need to create methods to protect messages from being eavesdropped. Over the course of the last centuries, the field has undergone a profound transformation, since now messages are not written on parchments anymore, but travel from one party to the other through the use of the internet. Indeed, the latter half of the 20th century constituted a pivotal shift with the advent of computers, leading to – much needed – groundbreaking developments in cryptography. Notably, Diffie and Hellman's well renowned work in 1976 – "New directions in cryptography" [DH76] – introduced *public-key cryptography*, revolutionizing the landscape of information security. This advancement enabled secure communication without the need for a pre-shared secret, relying instead on the assumed hardness of computational problems such as the *integer factorization* and the *discrete logarithm problem.*

Public-key cryptography quickly became the backbone of several modern conveniences, such as secure online payments and private messaging applications. It operates on the principle that while anyone can encrypt a message using a recipient's public key, only the intended recipient can decrypt it using their private key. Similarly, *digital signatures* allow the receiver to verify the authenticity of a message, ensuring it originated from the correct sender and remained untampered throughout the transmission over the *insecure channel.* These capabilities are crucial for maintaining the *integrity* and *confidentiality* of information exchanged over untrusted networks like the internet.

However, the field of cryptography is on the brink of another revolution as *quantum computers* make their advent. Unlike classical computers, which operate on binary digits (`0`s and `1`s), quantum computers utilize quantum bits (also called `qubits`) that can represent and process information in new – and in theory faster – ways. In 1994, Peter Shor demonstrated that quantum computers could efficiently solve the *integer factorization problem*, as well as the *discrete logarithm problem*, and thus constitute a major threat to the security foundations of widely-used cryptographic systems [Sho99]. However, these algorithms are still far from being practical since quantum computers are still a long way away from being efficient. Moreover, quantum computers do not represent the end of any hope of achieving secure communications over insecure channels due to the fact that there are still several problems that are potentially *quantum-hard.*

The threat presented by quantum computers has motivated the development of *post-quantum cryptography*, which seeks to create cryptographic protocols resistant to *quantum attacks.* Recognizing the urgency of developing *quantum-secure* cryptographic primitives before quantum computers become efficient and practical, the National Institute of Standards and Technology (NIST) launched a competition in 2016 to identify and standardize the most promising *quantum-resistant* cryptographic algorithms. This competition, now in its fourth round [ST17], has highlighted several families of candidates that show potential for securing our digital future against the capabilities of quantum computers.

Until the Summer of 2022, right after advancing to the fourth round of the competition, among these candidates one could find an *isogeny-based key exchange* called Supersingular Isogeny Key Exchange (SIKE) [Jao+17], a promising approach rooted in the mathematical properties of maps between elliptic curves, that give rise to *isogeny-based cryptography.* At its core, this approach involves *isogenies*, that are algebraic morphisms between elliptic curves that preserve the group structure of the curves, making them a natural fit for cryptographic applications. The primary hard problem upon which isogeny-based cryptography relies is the so-called "*pure isogeny problem*", that consists of computing an isogeny between two *isogenous elliptic curves*. Since this problem is believed to be computationally hard even for quantum computers, it can provide a strong foundation for cryptographic security in a post-quantum world.

The exploration of isogenies in cryptography began in the early 21st century. Initial attempts in promoting isogeny-based cryptography were made by Teske's trapdoor system [Tes06] and the CRS key exchange protocol proposed independently by Couveignes [Cou06] and by Rostovtsev and Stolbunov [RS06]. However, these early protocols were based on isogenies between *ordinary* elliptic curves and faced challenges in terms of efficiency and security. The real breakthrough came with the focus on *supersingular* elliptic curves, which offer enhanced security properties due to the increased difficulty of the isogeny problem in this context.

The most notable development in this field is the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, introduced by Jao and De Feo in 2011 [JD11] and later improved with Plût's contributions in 2014 [DJP14]. SIDH represented a significant advancement in isogeny-based cryptography, providing a practical and efficient key exchange mechanism. The traditional Diffie-Hellman key exchange is mimicked, but SIDH operates within the framework of isogenies on supersingular elliptic curves, rather than simple group elements. The SIDH protocol, that forms the basis of the Supersingular Isogeny Key Exchange (SIKE) mechanism [Jao+17], was believed to be a very promising candidate for post-quantum cryptographic applications, due to its very compact public keys and reasonable efficiency. Indeed, among all post-quantum alternatives for cryptographic protocols, isogeny-based cryptography often offers reasonable execution times and requires smaller bandwidth compared to its *lattice-based* counterparts, which are likely the most efficient option.

On a very high-level, in SIDH two parties, Alice and Bob, operate by exchanging enough information (the so-called "public keys" of the participants) to combine with their respective "secret keys" (i.e. Alice holds a piece of information that is only known to her, and analogously for Bob) in order to establish a shared secret key. This is done, of course, through the use of isogenies between supersingular elliptic curves. The difficulty of finding a specific isogeny between two given supersingular elliptic curves is widely leveraged, since this is a problem (i.e. the pure isogeny problem or the "*path finding problem*") that is believed to remain hard even in the presence of a quantum-enabled adversary. From a very similar approach to the one presented in SIDH, several other inspired schemes were published, such as k-SIDH [AJL18], B-SIDH [Cos20a], SÉTA [De +21], SHealS [FP21], and SIKE [Jao+17], which heavily relied on the security analysis provided for SIDH in [DJP14, §6].

Overall, the rapid advancements in isogeny-based cryptography highlights the importance of this field in the attempt of reaching quantum-resistant cryptographic solutions.

This report stems from the series of devastating attacks to SIDH, published during the Summer of 2022 [CD23; Mai+23; Rob23a]. Indeed, since then, several countermeasures have been developed, leading to the emergence of more secure variants of SIDH-inspired cryptosystems. Specifically, we will discuss countermeasures such as M-SIDH and MD-SIDH [FMP23], as well as new constructions like FESTA [BMP23] and POKE [Bas24]. However, it's important not to overlook other countermeasures against SIDH attacks, such as binSIDH and terSIDH [BF23], along with additional newly-derived constructions such as QFESTA [NO23] and IS-CUBE [Mor23].

Outline of the report

The report is divided in two main parts, for a total of seven chapters.

In Part I we present the SIDH protocol and discuss how its security was completely broken in the Summer of 2022. In Part II, we present two countermeasures that were published to make the SIDH framework resistant to the attacks, together with two new constructions that originated from the SIDH attacks.

Part I is made up of three chapters:

Chapter 1 introduces the necessary preliminaries on elliptic curves, isogenies between elliptic curves and isogeny graphs. All the important results for these mathematical objects are stated in this chapter.

Chapter 2 is dedicated to the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange protocol [DJP14]. After presenting the mathematical idea at the core of the scheme, some interesting aspects on the algorithmic implementation of the protocol are presented and, lastly, the assumptions regarding the hardness of the problems upon which SIDH relies are discussed.

Chapter 3 presents three devastating "SIDH attacks" that were able to completely break SIDH (in chronological order). The first attack [CD23] is described more meticulously than the others: we show how it is possible to break SIDH in *heuristic* polynomial time in case of known *endomorphism ring* of the starting curve. Subsequently, we present the second attack [Mai+23], that improves on the efficiency and shows that SIDH can be broken in *proven* polynomial time in case of known *endomorphism ring* of the starting curve. The chapter concludes by presenting the last attack [Rob23a], that is the most devastating of the three, as it allows to break SIDH in polynomial time without requiring the knowledge of the *endomorphism ring* of the starting curve.

Part II is made up of four chapters:

Chapter 4 details M-SIDH & MD-SIDH [FMP23]: two countermeasures to SIDH that avoid the application of the SIDH attacks, showing that the SIDH framework can still be used – if in a very careful way.

Chapter 5 presents one of the new constructions that stem from the application of the SIDH attacks in a constructive manner to define a trapdoor function, from which a public-key encryption scheme was obtained [BMP23].

Chapter 6 describes another new construction, that applies the SIDH attacks in a constructive way to obtain a public-key encryption scheme as well. In this case, the central focus is placed on the fact that the SIDH attacks allow for representations of higher-dimensional isogenies that allow to obtain an extremely compact and efficient scheme.

Chapter 7 concludes the report by summarizing what was presented. It also proposes future work, and finally acknowledges the people that the author wishes to thank and is deeply grateful to.

Objectives and contributions

The primary goal of this report is to provide additional clarity on how the various topics presented throughout the chapters interconnect and complement each other. For instance, significant effort has been dedicated in clarifying the attacks on SIDH to offer a solid starting point for less experienced readers who wish to explore isogeny-based cryptography.

This journey has been complex and demanding, culminating in the creation of this report. The author's main contribution lies in clarifying challenging aspects of the papers and sources reviewed. The ideas presented are not claimed as original contributions, since the objective was to introduce existing knowledge in a comprehensible manner, acknowledging the intricate nature of isogeny-based cryptography and the time required to thoroughly understand it.

The completion of this report marks a highly enriching learning experience for the author. Moreover, the report aims to share this newly acquired knowledge and serves as a reference that the author would have wished to have at his disposal during his learning journey, even though overcoming unclear passages in the sources deeply enhanced the learning experience. Regardless, the author hopes that this work will be valuable to others who wish to approach isogeny-based cryptography, particularly through the first three chapters, which focus on achieving clarity for a reader with no prior knowledge on the subject.

Part I

Supersingular Isogeny Diffie-Hellman (SIDH)

Chapter 1

Isogeny-based cryptography

In order to present the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, we first need to introduce what isogenies between elliptic curves are and, for the sake of this report being as self-contained as possible, we shall also introduce what an elliptic curve is. Hence the present chapter presents the most relevant background underlying the remainder of the survey.

Therefore, let us introduce elliptic curves in Section 1.1, and postpone the formal presentation of isogenies to Section 1.2, concluding with a presentation of isogeny graphs in Section 1.3. Some recommended references for a reader that is interested in diving deeper into the concepts presented throughout the chapter are the book "The Arithmetic of Elliptic Curves" by Joseph Silverman [Sil09], Lorenz Panny's PhD thesis titled "Cyptography on Isogeny Graphs" [Pan21], Luca De Feo's "Mathematics of isogeny based cryptography" [De 17] and Antonin Leroux's PhD thesis "Quaternion Algrebras and isogeny-based cryptography" [Ler22]. All of them should be accessible to the interested reader, but throughout the present chapter we will strongly reference the latter. For further references to a much more in-depth background on elliptic curves and their applications in cryptography, we refer the interested reader to Washington's "Elliptic Curves: Number Theory and Cryptography" [Was08], and Hartshorne's "Algebraic Geometry" [Har13].

Before delving into the details of the chapter, let us mention that, as is commonly done in mathematics, several equivalent definitions for most of the concepts presented in the chapter might be valid. We attempt to minimize the required prior knowledge to read through the report, by always choosing the most concrete and simple alternative, sometimes sacrificing generality or abstraction in exchange for a more accessible report.

1.1 Elliptic curves

Throughout the chapter, let k be a field and denote its algebraic closure by $\overline{k}$. Let us present the basic theory of elliptic curves, that are projective curves of genus 1 having a specified base point defined over k (i.e. not defined over the extension field $\overline{k}$). The history of projective spaces starts from the study of the *geometry of projections* (also known as *perspective* in classical painting) and initially appeared through the interest placed towards the *points at infinity.*

Definition 1 (Elliptic curve)**.** *Let k be a field. An* elliptic curve *over k is a pair $(E, \mathcal{O})$, where E is a smooth projective genus 1 curve over k and $\mathcal{O}$ is a distinguished k-rational point on E, called the* base point*.*

Let us suppose that the field k has characteristic such that $\operatorname{char}(k) \notin \{2, 3\}$. This will greatly simplify the representation of an elliptic curve, but the reader can find a more general definition in [Sil09, Chapter 3]. Throughout the chapter, E, E' will be elliptic curves over the field k, unless specified otherwise. The base point $\mathcal{O}$, also called "the point at infinity" will usually be omitted. We write E/k to indicate "an elliptic curve E defined over k". Definition 1 is hiding the fact that elliptic curves are, in simpler terms, identified by an homogeneous polynomial in three variables, that is usually called the *projective equation of the curve.* The set of points of an elliptic curve E over k is denoted by $E(k)$ and is simply made up of all the solutions to the *equation of the curve* over k. A point $P = (x, y)$ of the elliptic curve is a *k-rational* point if $P \in E(k)$, i.e. its coordinates are elements of the base field k. On the other hand, a point $P' = (x', y')$ is *not* a *k-rational* point if $P \in E(\overline{k})$, i.e. its coordinates are solution to the *equation of the curve* over the extension field $\overline{k}$.

The equations defining elliptic curves have very specific "shapes" and for this reason we can use the term "elliptic curve model" to mention a family of polynomials, of similar shapes, that define elliptic curves. The most well known family of polynomials are the so-called "Weierstrass curves". The Weierstrass model can be considered the canonical way of representing elliptic curves because every elliptic curve admits such a representation. Indeed it is possible to manipulate, through the use of arithmetic operations, any equation tied to an elliptic curve so that it is expressed in Weierstrass form.

Definition 2 (Weierstrass curve)**.** *A* short Weierstrass curve E/k *is the locus in* $\mathbb{P}^2(\overline{k})$ *defined by the projective equation*

$$E_{a,b} : Y^2Z = X^3 + aXZ^2 + bZ^3, \tag{1.1}$$

with $a, b \in k$ *such that* $\Delta = -16(4a^3 + 27b^2) \neq 0$. *The point* $[0 : 1 : 0]$ *is the point at infinity of* $E_{a,b}$ *and we will refer to it as* $\mathcal{O}$.

Equation (1.1) *is often written in its* affine form, *rather than its* projective form. *Let* $x = X/Z$ *and* $y = Y/Z$ *and equivalently define the same elliptic curve as the locus of points that satisfy the following equation:*

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

to which we add the point at infinity $\mathcal{O}$. *Therefore we have*

$$E_{a,b}(k) = \{(x, y) \in k^2 : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

Note that, for ease of notation, the subscript indicating the constants $a, b \in k$ *that define the curve will be often omitted when it is clear or irrelevant.*

Remark 1. *If the field k has characteristic such that* $\mathrm{char}(k) \notin \{2, 3\}$, *it is possible to show that any projective curve of genus 1 with a distinguished point* $\mathcal{O}$ *is isomorphic to a Weierstrass equation by mapping* $\mathcal{O}$ *onto the point at infinity* $[0 : 1 : 0]$.

Remark 2. *Note that for all models, each point* $P \in E(k)$ *can be represented by its* projective coordinates $(X_P : Y_P : Z_P) \in \mathbb{P}^2(k)$, *but also through its* affine coordinates $(x_P, y_P) \in k^2$ *since the projective equation can be mapped to an affine equation representing the model of* E. *Such a mapping is defined for all points of* $E(k)$, *except for the point at infinity* $\mathcal{O}$, *that cannot be represented in affine coordinates. Therefore we consider* $\mathcal{O}$ *as an abstract point of* E *and "add it by hand" to the set of points of the elliptic curve. We implicitly define the coordinate-projection functions, call them* x *and* y *for any elliptic curve* E, *and write* $x(P)$, $y(P)$ *to indicate the* x*-coordinate of the point* P *and the* y*-coordinate of the point* P, *respectively. The former is explicitly defined in* (1.3) *as* $x := \mathrm{pr}_1$.

Weierstrass models are vastly used for theoretical results, but are not the most efficient models for implementing algorithms, since the operations are more expensive than other alternatives such as *Montgomery* models or *Edwards* models. Indeed, these last two models are preferred for faster implementations, as we will see in Section 2.3, but not every elliptic curve admits a rational Montgomery or an Edward model[1]. Let us present below a definition (taken from [Ler22]) for the notion of *Montgomery curves*, introduced in [Mon87] to speed-up Lenstra's ECM integer factorization method [Len87], as well as the notion of *Edwards curves*, that provide *complete addition formulas* so that every pair of points can be added by evaluating the same rational function with no case distinction. Note that using Edwards form, rather than Weierstrass form, also helps in avoiding side-channel attacks in cryptographic implementations, since they do not require inconvenient case distinctions when evaluating the addition formulas that will be presented in Definition 5. We refer the reader to [BL17; CS18] and [BL17; BL07], whether interested in a deeper understanding of Montgomery curves and Edwards curves, as well as the related efficient algorithms.

[1] Contrarily to what happens for equations in Weierstrass form, not all elliptic curve equations can be transformed into Montgomery or Edwards forms through the use of arithmetic operations.

Definition 3 (Montgomery curve)**.** *A Montgomery curve E over k is defined by the equation*

$$E_{A,B} : BY^2Z = X(X^2 + AXZ + Z^2) \tag{1.2}$$

with $A, B \in k$ such that $B \neq 0$ and $A^2 \neq 4$. The point $[0 : 1 : 0]$ is the point at infinity *of $E_{A,B}$.*

Equation (1.2) *is often written in its* affine form*, rather than its* projective form*. Let $x = X/Z$ and $y = Y/Z$ and equivalently define the same elliptic curve as the locus of points that satisfy the following equation:*

$$E_{A,B} : By^2 = x(x^2 + Ax + 1),$$

to which we add the point at infinity $\mathcal{O}$. Therefore we have

$$E_{A,B}(k) = \{(x, y) \in k^2 : By^2 = x(x^2 + Ax + 1)\} \cup \{\mathcal{O}\}.$$

Note that, for ease of notation, the subscript indicating the constants $A, B \in k$ that define the curve will be often omitted when it is clear or irrelevant.

Remark 3. *Even though Montgomery curves do not solve the problem of treating exceptional cases differently in the addition law, thus are vulnerable to side-channel attacks just like Weierstrass curves, they offer extremely efficient formulas for computations on the x-line, also known as the* Kummer line *of a Montgomery curve, that is described by the surjective projection morphism $(x(P), y(P)) \mapsto x(P)$ from the elliptic curve to the projective X-line. It might be interesting to notice that the Kummer line can be also be viewed as the quotient $X = E/\{\pm 1\}$, since the points P and $-P$ are identified as the same point by the projection morphism*

$$\begin{aligned} \mathrm{pr}_1 : \quad & E && \longrightarrow \quad X \\ P = & \begin{pmatrix} x(P) \\ y(P) \end{pmatrix} && \longmapsto x(P) \end{aligned} \tag{1.3}$$

Keeping this in mind note that, even though we cannot perform the addition of two points anymore over the Kummer line, the scalar-multiplication operation can be implemented very efficiently using the ladder step

$$\texttt{DBLADD} : (P, Q, P - Q) \mapsto ([2]P, P + Q)$$

enhanced by the differential addition[2] *between P and Q. This mapping is well defined on the Kummer line $X = E/\{\pm 1\}$ and can be used as a building block for algorithms that perform scalar-multiplication of points such as the* Montgomery ladder*.*

Definition 4 (Edwards curve)**.** *A* (twisted) Edwards curve *E over k is defined by the equation*

$$E_{a,d} : (aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2 \tag{1.4}$$

with $a, d \in k$ such that $ad(a - d) \neq 0$. The point $[0 : 1 : 0]$ is the point at infinity *of E.*

Equation (1.4) *is often written in its* affine form*, rather than its* projective form*. Let $x = X/Z$ and $y = Y/Z$ and equivalently define the same elliptic curve as the locus of points that satisfy the following equation:*

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2,$$

to which we add the point at infinity $\mathcal{O}$. Therefore we have

$$E_{a,d}(k) = \{(x, y) \in k^2 : ax^2 + y^2 = 1 + dx^2y^2\} \cup \{\mathcal{O}\}.$$

Note that, for ease of notation, the subscript indicating the constants $A, B \in k$ that define the curve will be often omitted when it is clear or irrelevant.

[2] A differential addition is a sum $A + B$ such that the value $C = A - B$ is known.

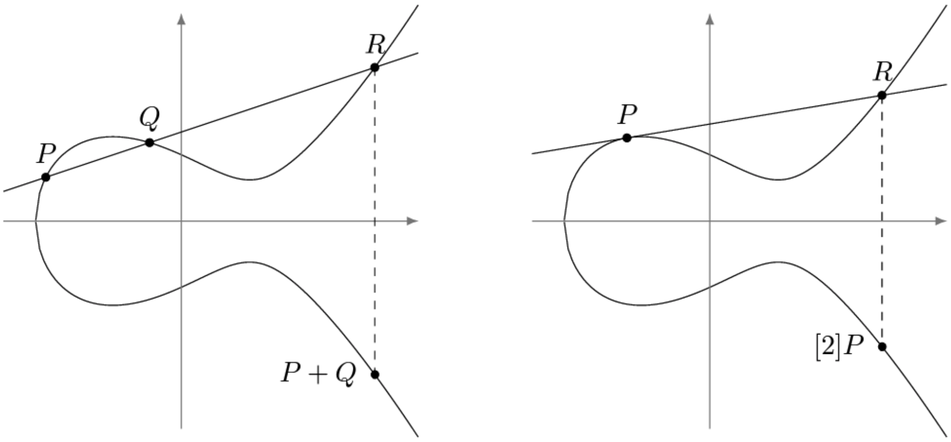

Figure 1.1: An elliptic curve defined over $\mathbb{R}$, and the graphical representation of its group law. Taken from [De 17, Figure 1].

Remark 4. *Edwards curves, just like Montgomery curves, allow to perform some arithmetic operations in a very efficient manner on the Kummer line. Additionally, they also have* complete addition formulas*, i.e. any two points can be added by evaluating the very same rational functions, with no exceptional points and no case distinction. Therefore, the big advantage of using Edwards curves is that they permit to build efficient side-channel resistant implementations of cyptosystems based on elliptic-curves that need to compute point additions, rather than scalar multiplications.*

Due to Remark 1, we know that any elliptic curve is defined by a cubic equation. Moreover, Bezout's theorem states that any line in $\mathbb{P}^2$ intersects the cubic curve in exactly three points, taken with multiplicity. Therefore, it is possible to define a group law by requiring that three co-linear points sum to zero.

Definition 5 (Group law)**.** *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on E different from the point at infinity, then we define an addition law $\oplus$ on E as follows:*

- *$P \oplus \mathcal{O} = \mathcal{O} \oplus P = P$ for any point $P \in E$;*
- *If $x_1 = x_2$ and $y_1 = -y_2$, then $P_1 \oplus P_2 = \mathcal{O}$;*
- *Otherwise, set*

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \textit{if } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \textit{if } P = Q \end{cases}$$

then the point $P_3 = P_1 \oplus P_2 = (x_3, y_3)$ is defined by

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= -\lambda x_3 - y_1 + \lambda x_1. \end{aligned}$$

It can be shown that the law described above defines an Abelian group, thus we will simply write $+$ instead of $\oplus$ and indicate it as $(E(k), +)$. The n-th scalar multiple of a point P will be denoted by $[n]P$. When E is defined over k, the subgroup of its k-rational points over k is denoted as $E(k)$. A graphical representation of the group law on an elliptic curve defined over $\mathbb{R}$ is shown in Figure 1.1.

Since E is an Abelian group, it has a free part and a torsion part. The former is of limited interest for the report, so we just mention that it is made up of all elements of infinite order. The latter subgroup of E, instead, is crucial for the description of SIDH, the main topic of the report, and is the subgroup of E that consists of all elements that have finite order. Let us define this concept in a more precise way.

Definition 6 (Torsion subgroup). *For any $m \in \mathbb{Z}$, we write*

$$\begin{array}{rccc} [m]\colon & E(k) & \longrightarrow & E(k) \\ & P & \longmapsto & [m]P \end{array} \tag{1.5}$$

for the scalar multiplication-by-m morphism. *The kernel of $[m]$ over the field $\overline{k}$ is the m-*torsion subgroup *and is denoted as $E[m]$. In other words, $E[m]$ is made up of all points over $\overline{k}$ whose order is a multiple of m and can be defined as follows:*

$$E[m] := \{P \in E(\overline{k}) : [m]P = \mathcal{O}\}.$$

For the sake of completeness, let us define the torsion subgroup *of E, denoted by E_{tors}, as the set of all points over $\overline{k}$ of finite order:*

$$E_{tors} = \bigcup_{m=1}^{\infty} E[m].$$

*However, it is worth mentioning that throughout the report we will always refer to the m-*torsion *even when writing just "torsion".*

Note that torsion subgroups are very important for the theory of elliptic curves: if the base field is a finite field $k = \mathbb{F}_q$, then the finite Abelian group $E(\mathbb{F}_q)$ is entirely determined by its intersection with all the m-torsion subgroups $E[m]$ [Sut21]. The structure of $E[m]$ is easy to determine in most cases and can be characterized by the following proposition. Its relevance will be clearer when discussing SIDH, as we will be in the first case of the characterization.

Proposition 1 (Characterization of the m-torsion). *Let E be an elliptic curve defined over a field k, and let m be the power of a prime. The m-torsion subgroup of E, denoted as $E[m]$, has the following structure.*

$$E[m] \cong \begin{cases} \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z} & \textit{if } \operatorname{char}(k) = 0 \textit{ or } \operatorname{char}(k) \nmid m \\ \mathbb{Z}/m\mathbb{Z} \textit{ or } \{\mathcal{O}\} & \textit{otherwise} \end{cases}$$

If $\operatorname{char}(k) = p$, *then $E[p]$ could be either $\mathbb{Z}/p\mathbb{Z}$ or $\{\mathcal{O}\}$. The former case is called* ordinary *and the latter is called* supersingular.

To conclude the section, let us introduce the notions of *isomorphism class* and *j-invariant.* Two curves E, E' are *isomorphic* over a field k if there exists a k-rational isomorphism between them, i.e. an isomorphism defined by coefficients taken from the field k. This creates an equivalence relation on the set of elliptic curves over k, and we will often consider isomorphism classes of elliptic curves over $\overline{k}$. Indeed, we will often say "an elliptic curve" while really meaning "an isomorphism class of elliptic curves". The *j-invariant* gives a unique representative for a class of isomorphic curves. For Weierstrass and Montgomery curves, for instance, this quantity is equal respectively to

$$j(E) = 1728 \cdot \frac{4a^3}{4a^3 + 27b^2} \qquad \text{and} \qquad j(E) = 256 \cdot \frac{(A^2-3)^3}{A^2-4}.$$

Proposition 2. *Two curves E, E' defined over k are isomorphic over $\overline{k}$ if and only if $j(E) = j(E')$.*

Remark 5. *Note that* isogenies, *that will be introduced in Section 1.2, will provide another equivalence relation for elliptic curves, i.e.* being isogenous. *The* isomorphic *relation is a specific case of the equivalence relation introduced by isogenies, since an isomorphism is simply an isogeny of degree 1.*

Before concluding the present section, let us provide one additional results that involves supersingular elliptic curves over finite fields. Since we consider an elliptic curve over a finite field $k = \mathbb{F}_q$, the group of k-rational points is finite, hence the group $E(\overline{k})$ is made up of exclusively torsion elements. Indeed, Hasse's theorem is useful for bounding the number of k-rational points of an elliptic curve and is taken from [De 17, Corollary 10].

Theorem 1 (Hasse's theorem). *Let E/k be an elliptic curve defined over a finite field $k = \mathbb{F}_q$, then*

$$\mid \#E(k) - q - 1 \mid \leq 2\sqrt{q}.$$

Proof. We refer the interested reader to [Sil09, Chapter V, Theorem 1.1]. □

1.2 Isogenies between elliptic curves

After having presented some basic concepts from the theory of elliptic curves, let us now introduce the main topic of the report: isogenies, that are specific maps between elliptic curves. Indeed, we are particularly interested in studying maps that preserve both natures of elliptic curves: as projective curves, and as groups.

Since preserving the former is really equivalent to preserving its projective (thus also affine) equation, let us introduce *invertible algebraic maps*. These maps are linear changes of coordinates that preserve the Weierstrass form of the equation. Since it is known that linear maps preserve lines, it is intuitive to believe that they also preserve the group law (since the latter relies on the tracing lines between points, c.f. Figure 1.1). As is stated in [De 17], it can be verified that all such maps are of the form

$$(x, y) \mapsto (u^2 x', u^3 y')$$

for some $u \in \overline{k}$. Note that such map defines an isomorphism $E_{a,b} \cong E_{au^4,bu^6}$ between the Weierstrass curves defined by the equations

$$E_{a,b} : (y')^2 = (x')^3 + ax' + b \qquad \text{and} \qquad E_{au^4,bu^6} : y^2 = x^3 + au^4 x + bu^6.$$

Isogenies are introduced when focusing on preserving the group structure, as they are surjective group morphisms (not necessarily invertible) between two elliptic curves. As is noted in [De 17], isogenies are algebraic maps as well. Throughout the report, we will focus on *cyclic separable isogenies*. Moreover, unless differently specified, we will simply say “isogeny” when referring to cyclic separable isogenies. Let us now present a series of definitions and results concerning isogenies between elliptic curves.

Definition 7 (Isogeny). *Let E, E' be two elliptic curves. An* isogeny *ϕ from E to E' is a morphism*

$$\phi : E \to E'$$

such that $\phi(\mathcal{O}_E) = \mathcal{O}_{E'}$, where $\mathcal{O}_E$ and $\mathcal{O}_{E'}$ are the points at infinity of E and E', respectively. Two curves are said isogenous *if there is an isogeny from E to E' such that $\phi(E) \neq \{\mathcal{O}_{E'}\}$. This constitutes an equivalence relation.*

Definition 8 (Rational or irrational isogeny). *Let E, E' be two elliptic curves defined over the field k. Let $\phi : E \to E'$ be an isogeny between the two curves. We distinguish between two cases:*

1. *The isogeny ϕ is* rational *if its kernel is defined over the field k;*
2. *The isogeny ϕ is* irrational *if its kernel is defined over the extension field $\overline{k}$.*

Let us present a theorem taken from [De 17, Theorem 7] that attempts to summarize what was presented so far, by characterizing isogenies between elliptic curves.

Theorem 2 (Characterization of isogenies). *Let E, E' be two elliptic curves defined over k. Let $\phi : E \to E'$ be a map between them. The following are equivalent:*

1. *ϕ is an isogeny;*
2. *ϕ is a surjective group morphism;*
3. *ϕ is a group morphism having finite kernel;*
4. *ϕ is a non-constant algebraic map between projective curves mapping the point at infinity of E onto the point at infinity of E'.*

Definition 9 (Degree and separability). *Let $\phi : E \to E'$ be an isogeny defined over a field k. Let $k(E), k(E')$ be the function fields of E, E'*[3]*. By composing ϕ with the functions of $k(E')$, we obtain a subfield of $k(E)$ that we denote by $\phi^*(k(E'))$*[4] *and the following hold true:*

1. *The degree of ϕ is defined as $\deg \phi = [k(E) : \phi^*(k(E'))]$ and is always finite;*
2. *ϕ is said to be* separable *if the extension of the function fields is* separable. *In particular, ϕ can be separable for the following two reasons:*
   (a) *If $\mathrm{char}(k) = 0$, then all isogenies are separable;*
   (b) *If $\mathrm{char}(k = p)$ and $\gcd(\deg \phi, p) = 1$, then ϕ is separable;*
3. *If ϕ is separable, then $\deg \phi = \# \ker \phi$;*
4. *If ϕ is separable, then ϕ is* cyclic *if and only if $\ker \phi$ is* cyclic.

Throughout the report, we will often refer to an isogeny having degree equal to $d \in \mathbb{Z}$ by calling it a "*d-isogeny*". Furthermore, since we will be mostly considering *cyclic separable isogenies*, we can take $\deg \phi = \# \ker \phi$ as the definition of the degree of the isogeny ϕ.

Moreover, let us consider separable isogenies and state the following. For any elliptic curve E, there is a unique correspondence between the separable isogeny $\phi : E \to E'$ itself and a finite subgroup of E (defined over $\overline{k}$), since we can associate ϕ to its kernel $\ker \phi$. Indeed, separable isogenies are entirely determined by their kernel. In other words, if we have a separable isogeny ϕ, we can uniquely determine a subgroup of the domain curve, this being the kernel $\ker \phi$.

Conversely, there is also a unique correspondence between finite subgroups G of E (defined over $\overline{k}$) and separable isogenies $\phi : E \to E'$. Indeed, separable isogenies can be entirely determined by a subgroup G of E (defined over $\overline{k}$) that acts as the kernel of the isogeny. In other words, we can take a domain curve E and a subgroup G of E and obtain, through an application of Vélu's formals [Vél71], an isogeny $\phi : E \to E' := E/\langle G \rangle$. The following proposition, taken from [De 17, Proposition 25], summarizes what we just discussed.

Proposition 3. *Let E be an elliptic curve, and let G be a finite subgroup of E. There are a* unique *elliptic curve $E' := E/\langle G \rangle$, and a unique separable isogeny ϕ, such that $\ker \phi = G$ and $\phi : E \to E'$.*

Proof. We refer the interested reader to [Sil09, Proposition III.4.12]. □

An extremely interesting and useful property of isogenies is that every isogeny naturally has a complementary isogeny in the opposite direction, called the *dual isogeny*. Note that, even though the dual behaves similarly to the inverse of the isogeny, modulo a scalar multiplication by the degree, the two are not the same in general. Moreover, the existence of the dual proves that *being isogenous* is actually an equivalence relation. Let us present the definition of dual isogeny, taken from [Ler22, Definition 1.1.13], as well as an important theorem to better understand this mathematical object.

Definition 10 (Dual isogeny). *For any isogeny $\phi : E \to E'$ of degree d, there exists a unique isogeny $\hat{\phi} : E' \to E$ of the same degree such that $\phi \circ \hat{\phi} = [d]$. We call $\hat{\phi}$* the dual *of ϕ. It is interesting to notice that when $\gcd(d, \mathrm{char}(k)) = 1$, then we have $\ker \hat{\phi} = \phi(E[d])$.*

[3] The *function field* $k(E)$ is the field of all functions over the elliptic curve E. A function over an elliptic curve is a rational map $f(x, y) \in k(x, y)$ that is defined for at least one point. The set $k(x, y)$ is made up of all rational bivariate functions with coefficients in k. Let us present two examples to make matters clearer. Let $E/k : y^2 = x^3 + ax + b$ be a curve in Montgomery form.

- $f(x, y) = \frac{1}{y^2 - x^3 - ax - b}$ is *not* a function over E because it is not defined $\forall\ P \in E(k)$.
- $g(x, y) = y^2 - x^3 - ax - b$ is a function over E. It is the [0]-function over E.

[4] Given $\phi : E \to E'$, define the *pull-back* of ϕ as $\phi^*(f) = f \circ \phi$:

$$\begin{array}{rccc} \phi^* \colon & k(E') & \longrightarrow & k(E') \\ & g & \longmapsto & f \circ \phi. \end{array}$$

Therefore, we have that $\phi^*(k(E'))$ is a subfield of $k(E)$.

Theorem 3. *Let $\phi : E \to E'$ be an isogeny of degree d. There is a* unique *isogeny $\hat{\phi} : E' \to E$ such that*

$$\hat{\phi} \circ \phi = [\deg \phi] = [d]_E : E \to E \qquad \text{and} \qquad \phi \circ \hat{\phi} = [\deg \hat{\phi}] = [d]_{E'} : E' \to E'.$$

The isogeny $\hat{\phi} : E' \to E$ is called the dual isogeny *of ϕ and has the following properties:*

1. *$\hat{\phi}$ is defined over k if and only if ϕ is defined over k;*
2. *$\widehat{\psi \circ \phi} = \hat{\phi} \circ \hat{\psi}$ for any isogeny $\psi : E' \to E''$;*
3. *$\widehat{\psi + \phi} = \hat{\psi} + \hat{\phi}$ for any isogeny $\psi : E \to E'$;*
4. *$\deg \phi = \deg \hat{\phi}$;*
5. *$\hat{\hat{\phi}} = \phi$.*

Considering isogenies with some added requirements on the degree or on the codomain curve, we can define three additional mathematical objects.

Definition 11 (Isomorphism, automorphism and endomorphism)**.** *Let us define the three following objects, that are particular isogenies:*

1. *An* isomorphism *is defined as an isogeny $\phi : E \to E'$ of degree* $\deg \phi = 1$*;*
2. *An* automorphism *is defined as an isomorphism $\phi : E \to E$ from a curve to itself;*
3. *An* endomorphism *is defined as an isogeny $\theta : E \to E$ from a curve to itself, having degree* $\deg \theta = t > 1$.

The latter will be the most relevant to the scope of the report.

Remark 6. *A typical example of endomorphism is the scalar* multiplication-by-m *endomorphism defined in* (1.5)*, whose degree is* $\deg[m] = m^2$*. Recall that its kernel over the field $\overline{k}$ is exactly the m-torsion subgroup $E[m]$. As stated in [De 17], for most elliptic curves this is the only type of endomorphisms that they* come equipped with*, but for other curves one can find several* non-trivial *endomorphisms, such as the Frobenius endomorphism.*

Definition 12 (Frobenius endomorphism)**.** *Let E/k be an elliptic curve defined over a finite field $k = \mathbb{F}_q$. The curve E comes equipped with an endomorphism that is typically, but not always, non-scalar and is the* (q-power) Frobenius endomorphism *π defined as*

$$\begin{array}{rccc} \pi : & E & \longrightarrow & E \\ & P = (x, y) & \longmapsto & P' = (x^q, y^q) \end{array}$$

Given the notion of endomorphism, let us introduce the notion of the *endomorphism ring.* Let E/k be an elliptic curve, then the endomorphism ring of E is the ring $(\mathrm{End}(E), +, \circ)$ defined from the set of all endomorphisms over the curve E, together with the multiplication-by-0 map, with respect to the addition and composition of endomorphisms. The most relevant operation of the two, for the sake of the report is the *composition of endomorphisms*, that serves the role of “multiplication” between elements of the ring.

Let us also introduce the following definition, taken from [Wat69], in the attempt of demystifying what a *supersingular* elliptic curve is, by providing a clearer definition than the one presented in Proposition 1.

Definition 13 (Supersingular elliptic curve)**.** *An elliptic curve is* supersingular *if its endomorphism ring over k is non-commutative.*

Let us mention that the degree behaves multiplicatively with respect to the composition of isogenies, as discussed in Remark 7. For instance, if E is d-isogenous to E' (i.e. the connecting isogeny is of degree d) and E' is isomorphic to E'' (i.e. the connecting isogeny is of degree 1), then E is also d-isogenous to E''.

$$\begin{tikzcd}
 & E_1 \arrow[dr, "\phi'_2 = \phi_{1*}\phi_2"] & \\
E_0 \arrow[ur, "\phi_1"] \arrow[rr, "\phi = \phi'_2 \circ \phi_1 = \phi'_1 \circ \phi_2"] \arrow[dr, "\phi_2"'] & & E_3 \\
 & E_2 \arrow[ur, "\phi'_1 = \phi_{2*}\phi_1"'] &
\end{tikzcd}$$

Figure 1.2: Configuration of a commutative isogeny diagram, taken from [Ler22, Figure 1.1].

Remark 7 (Degree of the composition of isogenies)**.** *Note that, by convention, we set* $\deg[0] = 0$*. As stated in [Sil09, §III.4], this ensures that*

$$\deg \psi \circ \phi = \deg \psi \cdot \deg \phi$$

for all chains (i.e. compositions) of isogenies $\psi \circ \phi : E_1 \xrightarrow{\phi} E_2 \xrightarrow{\psi} E_3$*.*

Before concluding the present section, with strong reference to [Ler22, §1.1.3], let us present the two very relevant notions of *isogeny decomposition* and *isogeny commutative diagram*. Introducing the concept of commutative diagrams of isogenies is crucial because they will be widely employed throughout the entire report. Any isogeny ϕ of degree $d = \prod_{i=1}^{n} d_i$, with d_i being (not necessarily coprime) integers, can be decomposed as a chain of n isogenies such that $\phi = \phi_n \circ \cdots \circ \phi_1$, where each ϕ_i is an isogeny of degree d_i. Let us expand a bit more on the notation adopted for Figure 1.2, where we have a commutative isogeny diagram for the case of $n = 2$. Let d_1, d_2 be two coprime integers and ϕ a $d_1 d_2$-isogeny, i.e. an isogeny of degree $d_1 d_2$. We decompose (or factor) the isogeny as $\phi = \phi'_2 \circ \phi_1 = \phi'_1 \circ \phi_2$, where each ϕ_i, ϕ'_i is a d_i-isogeny. Moreover, it can be shown that $\ker \phi'_1 = \phi_2(\ker \phi_1)$ and, analogously for $\ker \phi'_2$, $\ker \phi'_2 = \phi_1(\ker \phi_2)$. This underlies the concept of *push-forward*: $\phi'_1 = \phi_{2*}\phi_1$ is the pushforward of ϕ_1 through ϕ_2 and $\phi'_2 = \phi_{1*}\phi_2$ is the pushforward of ϕ_2 through ϕ_1, as formalized in Definition 14. Conversely, one can also view ϕ_1 as the *pull-back* (previously introduced in Footnote 4) of ϕ'_1 by ϕ_2 (that is the same as the push-forward of ϕ'_1 through $\hat{\phi}_2$, up to isomorphism) such that $\phi_1 = {\phi_2}^* \phi'_1 = \hat{\phi}_{2*}\phi'_1$.

Definition 14 (Push-forward)**.** *If two isogenies* $\phi_1 : E \to E'$ *and* $\phi_2 : E \to E_2$ *share the same domain, we can express the* push-forward *of* ϕ_2 *under* ϕ_1 *as* $\phi_{1*}\phi_2$*. This yields an isogeny* $\phi_{1*}\phi_2 = \phi'_2 : E_1 \to E_3$ *such that* $\ker \phi'_2 = \phi_1(\ker \phi_2)$*. This is depicted in the commutative isogeny diagram of Figure 1.2.*

Lastly, let us introduce the notion of two isogenies being *parallel* with respect to a third one. This concept will be recalled in Chapter 6.

Definition 15 (Parallel isogenies)**.** *Two isogenies* ϕ *and* ϕ' *are called* parallel *with respect to an isogeny* ψ *if* $\ker \phi' = \psi(\ker \phi)$*. Indeed, one would have the configuration presented in Figure 1.3, that would explain the nomenclature.*

$$\begin{tikzcd}
E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \\
E_2 \arrow[r, "\phi'"] & E_3
\end{tikzcd}$$

Figure 1.3: Configuration when ϕ and ϕ' are parallel isogenies with respect to ψ.

1.3 Isogeny graphs

Let us now turn our attention to the graph structure induced by isogenies on the set of j-invariants of curves defined over a finite field. We have already mentioned that the set of all isogenous curves is an equivalence class, that is called *isogeny class*. Even when restricting ourselves to isogenies of degree ℓ, being *ℓ-isogenous* then constitutes an equivalence relation. Furthermore, it is not possible for an isogeny class to contain *ordinary* curves as well as *supersingular* curves, as stated in [De 17, §9]. This preamble is important because when discussing isogeny graphs, we are interested exclusively in elliptic curves up to isomorphism, i.e. we are interested in their j-invariant. Thus, we say that two j-invariants are isogenous if their corresponding curves are isogenous. Let us define what an *isogeny graph* is, taking the definition from [De 17, §9].

Definition 16 (Isogeny graph)**.** *An* isogeny graph *is a (multi)-graph*[5] *whose nodes are the j-invariants of isogenous curves and whose edges are isogenies between them.*

Due to the existence of *dual isogenies*, we know that for every edge representing the isogeny $E \to E'$, there will also be an edge representing the isogeny $E' \to E$ of the same length as the former. Therefore, the isogeny graph is usually drawn as an undirected graph.

Proposition 4 (Isogeny graphs)**.** *Let E/k be an elliptic curve defined over the finite field $k = \mathbb{F}_{p^2}$ of characteristic p. Let $\ell \neq p$ be a prime. Then the following hold true:*

1. *There are $\ell + 1$ distinct isogenies of degree ℓ with domain E* defined over the algebraic closure *$\overline{k}$;*
2. *There are $0, 1, 2$ or $\ell + 1$ isogenies of degree ℓ having domain E* defined over *k.*

Note that we will be interested, almost exclusively, in *supersingular isogeny graphs*, that are isogeny graphs whose set of vertices corresponds to *supersingular* elliptic curves. Thus, unless differently specified, we will simply say "isogeny graph" when referring to a supersingular isogeny graph.

Figure 1.4 taken from [De 17, Figure 15] shows two simple examples of supersingular isogeny graphs induced from the same vertex set, that is the set of all supersingular j-invariant defined over $\mathbb{F}_{p^2}$, where $p = 97$. Note that it is possible to obtain two completely different graphs from the same vertex set, as the graph on the left (in blue) is made of ℓ_A-isogenies and the graph on the right (in red) is made of ℓ_B-isogenies, with $\ell_A = 2$ and $\ell_B = 3$.

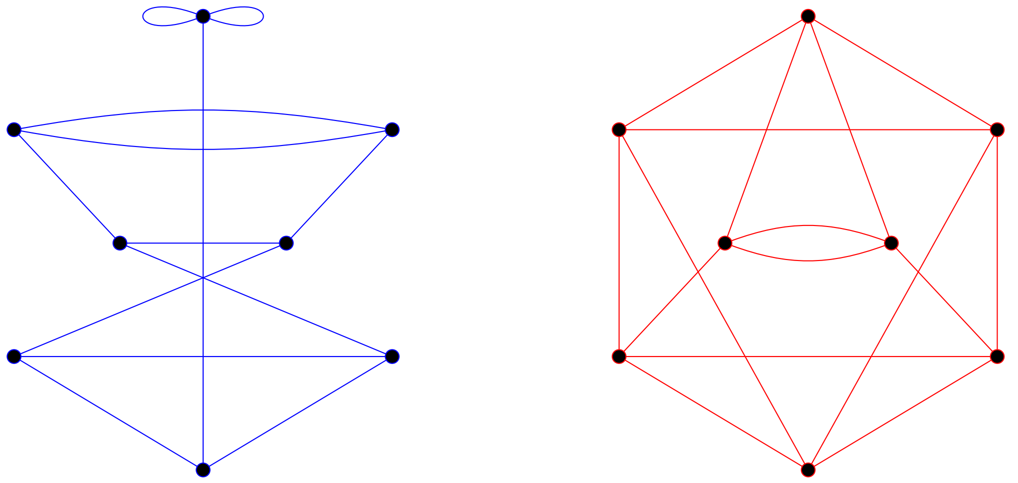

Figure 1.4: Supersingular isogeny graphs of degree 2 (left, blue) and 3 (right, red) on $\mathbb{F}_{97^2}$.

Of course, just like any other type of graph, *supersingular* isogeny graphs can also be "walked": it is possible to start at a given node and reach another node of the graph by following the edges. This translates to computing an isogeny between two curves, represented by the start and end nodes, with the isogeny having a degree that is a power of the degree of a single isogeny in the graph. So, for instance, walking 3 steps in a 2-isogeny graph (e.g. blue graph on the left in Figure 1.4) is equivalent to having a 2^3-isogeny having kernel of size 2^3. As we will see, this concept will be of extreme relevance through the entire report.

Therefore, let us tie together two of the most relevant concepts that we have encountered so far in the theory of isogenies: *kernels* and *isogeny graphs*, particularly *walks on isogeny graphs*.

[5] A multi-graph is a graph in which multiple edges can connect the same pair of vertices.

Remark 8 (Walk on the isogeny graph from a kernel)**.** *Let ϕ_A be a separable $\ell_A^{e_A}$-isogeny. Since a separable isogeny is uniquely defined by its kernel, and* $\deg \phi = \# \ker \phi$, *we have that a walk of length e_A in the ℓ_A-isogeny graph corresponds to an isogeny of degree $\ell_A^{e_A}$, because it is a composition of e_A sequential ℓ_A-isogenies. Thus, a walk of length e_A in the ℓ_A-isogeny graph corresponds to a kernel of size $\ell_A^{e_A}$. Such kernel is cyclic if and only if the walk does not backtrack*[6].

To conclude the section, let us present the concept of *ℓ-th modular polynomial* taken from [GV18, §6], that will be relevant for the upcoming chapters.

Definition 17 (ℓ-th modular polynomial)**.** *Let ℓ be an integer with $\ell > 2$. The ℓ-*th modular polynomial $\Phi_\ell(x, y) \in \mathbb{Z}[x, y]$ *is such that a pair $(j, j') \in \overline{\mathbb{F}}_q^2$ satisfies $\Phi_\ell(j, j') = 0$ if and only if there exist the following two:*

1. *Two elliptic curves E, E' over the extension field $\overline{\mathbb{F}}_q$ having j-invariants, respectively, j and j';*
2. *An isogeny $\phi : E \to E'$ of degree ℓ.*

Due to the existence of the dual *isogeny, it follows that $\Phi_\ell(x, y) = \Phi_\ell(y, x)$.*

Remark 9. *Given an elliptic curve E/k over the finite field $k = \mathbb{F}_q$, one can find all the curves E' that are ℓ-isogenous to E by computing the kernel of the univariate map $\Phi_\ell(j(E), y) \in \mathbb{F}_q[y]$ over the extension field $\overline{k} = \overline{\mathbb{F}}_q$.*

Furthermore, applying an algorithm due to Elkies [Elk+98], we can also compute the kernel of the isogeny connecting E to E', given E and the j-invariant of E', in exponential time in ℓ.

[6] A walk *backtracks* if a dual isogeny is used to go back to a previously visited curve of the graph. Cycles are not backtracking walks.

Chapter 2

SIDH – Supersingular Isogeny Diffie-Hellman

Supersingular Isogeny Diffie-Hellman (SIDH) is a post-quantum isogeny-based key exchange protocol initially proposed in [JD11], but was further improved in [DJP14]. The protocol underlies a key encapsulation mechanism called SIKE (Supersingular Isogeny Key Encapsulation) [Jao+17] that advanced to the fourth round of NIST's post-quantum standardization competition in May 2022 [ST17].

The SIDH protocol, that focuses on isogenies between *supersingular* elliptic curves, was mainly motivated by (at the time) recent developments in the field of constructing isogenies between *ordinary* elliptic curves with subexponential-time quantum algorithms. Thus, given the deteriorated security of protocols based on *ordinary* curves, the choice of the *supersingular* case, apart for a technical detail, stems from the fact that the fastest known quantum algorithm to compute isogenies between *supersingular* elliptic curves is exponential. Indeed the non-commutativity of the endomorphism ring in the *supersingular* case does not allow to extend the attack for the *ordinary* case. However, if on one hand this might help to preserve the security of the scheme, on the other hand it made achieving a successful key exchange more complicated and forced the authors in [DJP14] to resort to a trick. Indeed, to allow the parties Alice and Bob to reach a shared secret, the idea at the core of the SIDH protocol is to publish the images of the torsion bases under the action of the secret isogenies. In this way they will be able to construct a shared commutative diagram despite the non-commutativity of the endomorphism ring.

The key hardness assumption of many isogeny based protocols is based on the difficulty of recovering a large degree isogeny $\phi : E \to E'$ between two ordinary or supersingular elliptic curves, the so-called *isogeny path problem.* This problem has exponential quantum security for supersingular curves, if no additional information is provided on E and E'. However, we will see that the security of SIDH relies on the *Supersingular Isogeny with Torsion (SSI-T) problem*, and not on the more general and harder problem of supersingular *isogeny path problem*, due to the fact that it is necessary to publish the images of the torsion bases under the action of the secret isogenies in order to make the scheme work correctly.

After reaching the fourth round of the competition its security was completely broken in August of 2022 through the publication of three devastating attacks, that exploited the trick that made the key exchange possible, as we will see in Chapter 3.

After providing some background knowledge on what a key exchange protocol is in Section 2.1, we present the SIDH key exchange protocol in Section 2.2, of which we discuss some algorithmic aspects in Section 2.3. To conclude the chapter, we briefly present the hardness assumptions upon which SIDH relied in Section 2.4.

2.1 Key exchange protocol

The goal of SIDH is to achieve a successful *key exchange* between two parties, known as Alice and Bob. In other words, they aim to *establish a shared secret over an insecure communication channel.* Here, "insecure" means that an adversary, named Eve, could intercept and (depending on the threat model) potentially alter communications between Alice and Bob. Indeed, Eve aims at extracting or influencing the shared secret as much as possible.

This cryptographic primitive is crucial for secure communications because the shared secret key enables Alice and Bob to encrypt messages using a *secret-key encryption scheme*, that requires a secret key to be shared between the parties before they begin communicating. The components of the cryptographic primitive are the public parameters (the security parameter λ and a definition of the domain for the secretly shared key), the two parties (Alice and Bob), and the protocol made up of two distinct probabilistic algorithms (one for each party, so that the output is the public key). The required functionality of the key exchange primitive is that after both parties have executed their algorithms, and exchanged the outputs, they can reach a commonly agreed secret. So, slightly formalizing, we have that Alice executes $\mathcal{A}(\lambda, r_A)$, Bob executes $\mathcal{B}(\lambda, r_B)$, with respective randomness r_A and r_B, and in the end both can obtain the shared key $K = K_A = K_B$ by interacting with each other.

Let us provide an intuition on the security notions required by the cryptographic primitive. Against an *active* adversary, we should be aware of the fact that it is unavoidable to have the configuration where Eve interacts with both Alice and Bob, impersonating Bob while interacting with Alice and vice versa when interacting with Bob. This will lead to Eve knowing K_A and K_B. However, in this setting we must avoid the more devastating scenario where Eve can force $K_A = K_B$ and, thus, infer the secret key K. In the case of a *passive* adversary, we want to protect against *key recovery*, and also against *key distinguisher*. In simple terms, the former means that if Eve is provided with the transcript, i.e. the exchanges between Alice and Bob, and with the security parameter, then it must be hard for her to compute the shared secret K established between the two parties. The latter, instead, can be interpreted as the property such that if Eve is provided with the transcript, a key $\tilde{K}$ that she does not know whether it is the commonly established key between Alice and Bob or not, and the security parameter, it must be hard for her to determine whether $\tilde{K}$ is something random or if it is the actual secret key established between Alice and Bob.

Note that a key exchange protocol is called this way because the parties exchange so-called "public keys" (*public* because they are made publicly available as they could be intercepted by Eve, and thus by Kerckhoffs' principle the security of the scheme should not rely on them being secret). However, one must keep in mind that what the parties exchange are not actually keys, but simply pieces of information that allow to derive a common secret key with no prior secretly-shared knowledge between the parties.

2.2 The SIDH key exchange

At the core of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol there is the idea of making Alice and Bob go on random walks in two distinct isogeny graphs having the same vertex set, consisting of the j-invariants of the supersingular curves defined over $\mathbb{F}_{p^2}$. If we let p be a large prime, and ℓ_A and ℓ_B be two small primes, Alice's graph will be made of ℓ_A-isogenies, while Bob's graph will be made of ℓ_B-isogenies.

However, this is not enough for defining a key exchange protocol, since the labeling of these edges cannot be done in any canonical way: Alice and Bob would not know how to tell each other the walk that they performed to reach a given node in the graph. Therefore, in order to represent these walks on the graphs, SIDH will be based upon a specific construction that leverages the group structure of supersingular elliptic curves. Recall that a separable isogeny is uniquely defined by its kernel, and that $\deg \phi = \# \ker \phi$. More precisely, recall that a walk of length e_A in the ℓ_A-isogeny graph corresponds to an isogeny of degree $\ell_A^{e_A}$, because it is a composition of e_A sequential ℓ_A-isogenies, as stated in Remark 8. Hence, it also corresponds to a kernel of size $\ell_A^{e_A}$, that is cyclic if and only if the walk does not backtrack.

Alice will perform a secret walk of length e_A by obtaining an isogeny ϕ_A of degree $\ell_A^{e_A}$ through a secret cyclic subgroup $\langle A \rangle \subset E[\ell_A^{e_A}]$ of size $\ell_A^{e_A}$. Similarly, also Bob will choose a secret subgroup $\langle B \rangle \subset E[\ell_B^{e_B}]$ of size $\ell_B^{e_B}$ and use it to compute an isogeny ϕ_B of degree $\ell_B^{e_B}$. Since $\ell_A \neq \ell_B$ and they are also prime numbers, $\langle A \rangle + \langle B \rangle = \langle A, B \rangle$ is a cyclic subgroup of order $\ell_A^{e_A} \ell_B^{e_B}$. Therefore, it is possible to obtain an isogeny to $E/\langle A, B \rangle$. This is described in Figure 2.1.

$$\begin{aligned} \ker\phi_A &= \langle A\rangle \subset E[\ell_A^{e_A}] \\ \ker\phi_B &= \langle B\rangle \subset E[\ell_B^{e_B}] \\ \ker\phi'_A &= \langle \phi_B(A)\rangle \\ \ker\phi'_B &= \langle \phi_A(B)\rangle \end{aligned}$$

$$\begin{tikzcd} E \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E/\langle A\rangle \arrow[d, "\phi'_B"] \arrow[dl, dashed] \\ E/\langle B\rangle \arrow[r, "\phi'_A"'] \arrow[ur, dashed] & E/\langle A, B\rangle \end{tikzcd}$$

$$\begin{array}{c} \phi'_A : E/\langle B\rangle \to \dfrac{E/\langle B\rangle}{\langle \phi'_B(A)\rangle} \\ \cong \\ \phi'_B : E/\langle A\rangle \to \dfrac{E/\langle A\rangle}{\langle \phi'_A(B)\rangle} \end{array}$$

Figure 2.1: Commutative diagram constructed for the SIDH protocol, taken from [De 17, Figure 16]. Quantities colored in blue are known to Alice and quantities colored in red are known to Bob. The dotted blue (resp. red) line represents that Alice (resp. Bob) "moves" to the curve $E/\langle B\rangle$ (resp. $E/\langle A\rangle$). From there she (resp. he) will compute the last part of the diagram, ϕ'_A (resp. ϕ'_B).

From this idea, let us see how it is possible to define a protocol where Alice and Bob choose $\langle A\rangle$ and $\langle B\rangle$ to be random cyclic subgroups of some large enough torsion subgroup, and are able to both compute $E/\langle A, B\rangle$ (up to isomorphism) after exchanging enough information and keeping their respective secrets private. However, first we have two problems to tackle:

1. The points in the subgroups $\langle A\rangle$ (or $\langle B\rangle$) may not be rational as they may be defined over a large field extension up to degree $\ell_A^{e_A}$ (or $\ell_B^{e_B}$), that is the order of the subgroup $\langle A\rangle$ (or $\langle B\rangle$) by which we take the quotient $E/\langle A\rangle$ (or $E/\langle B\rangle$). This is a problem since an exponential amount of information would be required to explicitly represent all these points. Indeed, points in a field extension of degree m can be uniquely represented as elements of a vector space of dimension m over the base field.

2. It is not clear, at the moment, how Alice and Bob can both compute $E/\langle A, B\rangle$, and therefore close the commutative diagram, without revealing their secret isogenies ϕ_A and ϕ_B.

The first problem is addressed by exploiting the group structure of the supersingular curves[7] that are taken into account for the SIDH key exchange. Indeed, if we let p be a prime and E be a supersingular elliptic curve defined over the finite field $\mathbb{F}_{p^2}$, then $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p\pm1)\mathbb{Z})^2$ for a theorem on the group structure of supersingular elliptic curves presented in [De 17, Theorem 54]. Given that the prime p can be chosen freely, it is chosen such that $E(\mathbb{F}_{p^2})$ contains two large subgroups $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$ of coprime order. Therefore, after choosing $\ell_A^{e_A}$ and $\ell_B^{e_B}$, we take $p = \ell_A^{e_A}\ell_B^{e_B}f \mp 1$, where f is a small cofactor taken such that p is prime. In the majority of cases, $f = 1$. In conclusion, the choice of a prime of the form $p = \ell_A^{e_A}\ell_B^{e_B} \mp 1$ leads to a supersingular elliptic curve $E(\mathbb{F}_{p^2})$ that contains $\ell_A^{e_A-1} \cdot (\ell_A + 1)$ cyclic subgroups of order $\ell_A^{e_A}$. This number stems from the following reasoning driven by combinatorics. Recall that Proposition 4 states that given an elliptic curve E defined over the finite field $\mathbb{F}_{p^2}$, there are $\ell + 1$ distinct isogenies of degree ℓ with domain E defined over the algebraic closure $\overline{\mathbb{F}}_{p^2}$, where $\ell \neq p$ is a prime. Thus, if we view an $\ell_A^{e_A}$-isogeny as a composition of e_A sequential ℓ_A-isogenies that form a walk in the isogeny graph starting from the curve E, we have what follows: for the first isogeny, there are $\ell_A + 1$ possible choices (all the possible outgoing edges from the node representing E) and for the remaining $e_A - 1$ isogenies there are ℓ_A possibilities to choose from (since backtracking walks are not allowed).

This solves the first issue since now we are sure that a single point $A \in E(\mathbb{F}_{p^2})$ is enough to represent an isogeny of degree $\ell_A^{e_A}$, or equivalently a walk of length e_A. Indeed, we have that $\#E(\mathbb{F}_{p^2}) = \ell_A^{e_A}\ell_B^{e_B}$. Thus, $E(\mathbb{F}_{p^2})$ is generated by two points of order $\ell_A^{e_A}\ell_B^{e_B}$, let us say $E(\mathbb{F}_{p^2}) = \langle P, Q\rangle$. Taking linear combinations of P and Q, it is possible to obtain all the points of order a divisor of $\ell_A^{e_A}\ell_B^{e_B}$ (due to Lagrange's theorem). To conclude, by considering the subgroup generated by one single point R, obtained as a linear combination of P and Q, we can represent any isogeny of degree any divisor of $\ell_A^{e_A}\ell_B^{e_B}$, even an isogeny of degree $\ell_A^{e_A}$.

[7] Note that this is more complicated to do with ordinary curves, providing an additional motivation for why supersingular curves are preferred in isogeny-based cryptography.

The second problem is solved by a trick that is specific to SIDH and not used in other isogeny-based protocols such as CSIDH [Cas+18] or SQISign [De +20], that consists of making Alice and Bob reveal some additional information about the images of the torsion points under the secret isogeny to help each other compute the shared secret.

Let us describe how this is done in the SIDH key exchange protocol. To set up the scheme, Alice and Bob have publicly agreed on a prime $p = \ell_A^{e_A}\ell_B^{e_B}f \mp 1$ and a supersingular elliptic curve E such that $\#E(\mathbb{F}_{p^2}) = \ell_A^{e_A}\ell_B^{e_B}f$. The bases $\langle P_A, Q_A\rangle$ of $E[\ell_A^{e_A}]$ and $\langle P_B, Q_B\rangle$ of $E[\ell_B^{e_B}]$ are fixed[8]. Focusing on Alice, she will choose a random secret subgroup $\langle A\rangle = \langle [m_A]P_A + [n_A]Q_A\rangle \subset E[\ell_A^{e_A}]$ of order $\ell_A^{e_A}$ and compute the isogeny $\phi_A : E \to E/\langle A\rangle$. Bob will act analogously and in the end Alice will publish $E_A = E/\langle A\rangle$ and Bob will publish $E_B = E/\langle B\rangle$.

In order to obtain the shared secret $E/\langle A, B\rangle$, Alice will have to compute the isogeny $\phi'_A : E/\langle B\rangle \to E/\langle A, B\rangle$, whose kernel is $\phi_B(A)$. Thus, we note that Alice needs Bob's help to perform this computation and here is where the *SIDH trick* comes into play. Bob will publish the values $\phi_B(P_A)$ and $\phi_B(Q_A)$ and Alice will compute $\phi_B(A) = [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)$ to complete the protocol and obtain the shared secret curve $E_{AB} = \phi'_B \circ \phi_A(E) = \phi'_A \circ \phi_B(E) = E/\langle A, B\rangle$, as Bob will act analogously with Alice's published values. A convincing argument for the correctness of the protocol is provided in [Cos20b, §5].

We hereby describe the SIDH protocol and illustrate it in Figure 2.2:

Public parameters: Let ℓ_A, ℓ_B be primes. Let $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ be a prime. Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ of order $(p \pm 1)^2$. Let $\langle P_A, Q_A\rangle$ be a basis of $E[\ell_A^{e_A}]$ and $\langle P_B, Q_B\rangle$ be a basis of $E[\ell_B^{e_B}]$.

Public key (Alice): Alice samples two random elements $m_A, n_A \in_\$ \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A and computes the isogeny $\phi_A : E \to E_A = E/\langle A\rangle$ such that $A = [m_A]P_A + [n_A]Q_A$. Alice also computes the images $\phi_A(P_B)$ and $\phi_A(Q_B)$. Alice sends these points to Bob, together with the curve E_A.

Public key (Bob): Analogously, Bob samples two random elements $m_B, n_B \in_\$ \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B and computes the isogeny $\phi_B : E \to E_B = E/\langle B\rangle$ such that $B = [m_B]P_B + [n_B]Q_B$. Bob also computes the images $\phi_B(P_A)$ and $\phi_B(Q_A)$. Bob sends these points to Alice, together with the curve E_B.

Shared key (Alice): From $\phi_B(P_A)$ and $\phi_B(Q_A)$, Alice computes the isogeny $\phi'_A : E_B \to E_{AB} = E_B/\langle\phi_B(A)\rangle$ having kernel equal to $\langle[m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)\rangle$. The shared key is the value $j(E_{AB})$.

Shared key (Bob): Analogously, from $\phi_A(P_B)$ and $\phi_A(Q_B)$, Bob computes the isogeny $\phi'_B : E_A \to E_{BA} = E_A/\langle\phi_A(B)\rangle$ having kernel equal to $\langle[m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B)\rangle$. The shared key is the value $j(E_{BA}) = j(E_{AB})$.

Let us discuss the sizes of the parameters, by mentioning how the well known *claw problem* in complexity theory[9] can apply to our case and be solved by a meet-in-the-middle approach. Intuitively, the key space of SIDH depends on the size of the torsion subgroups $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$. Therefore, taking $\ell_A^{e_A} \sim \ell_B^{e_B}$ is recommended: trying to recover Alice's secret isogeny ϕ_A is equally difficult as trying to recover Bob's secret isogeny ϕ_B.

To select the size for p, let us analyze an easy adaptation of the meet-in-the-middle approach to find the secret $\ell_A^{e_A}$-isogeny $\phi_A : E_0 \to E_A$ in $O(\ell_A^{e_A/2})$ iterations (and $O(\ell_A^{e_A/2})$ storage). The attack starts by storing all possible end-points of walks of length $\lfloor e_A/2\rfloor$ starting from E_0. After that, perform walks of length $\lceil e_A/2\rceil$ starting from E_A, until a collision is found. The same result can be obtained

[8] Note that, since ℓ_A, ℓ_B and p are all primes, they are pairwise coprime. This implies, by Proposition 1, that the respective torsion subgroups are of dimension 2. Therefore, a basis must be made up of two elements.

[9] The *claw problem* is defined as follows: given two functions $f : A \to C$ and $g : B \to C$ with domains of equal sizes, find a pair (a, b) such that $f(a) = g(b)$. On a classical computer, it can be solved in $O(|A| + |B|)$ time and $O(|A|)$ space by building a hash table containing all values $f(a)$ for all $a \in A$ and looking for collisions with $g(b)$ for $b \in B$. On a quantum computer, it is possible to improve this strategy, obtaining the complexity $O\left(\sqrt[3]{|A||B|}\right)$ through the algorithm presented in [Tan09].

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\begin{array}{c} E_0, \\ \{P_A, Q_A\} \subset E_0[\ell_A^{e_A}] \\ \{P_B, Q_B\} \subset E_0[\ell_B^{e_B}] \end{array} \arrow[r, "\phi_A", "\ker(\phi_A) = \langle [m_A]P_A + [n_A]Q_A \rangle"'] \arrow[d, "\phi_B"', "(\star)"] & \begin{array}{c} E_0/\langle A \rangle \\ \phi_A(P_B), \phi_A(Q_B) \end{array} \arrow[d, "(\star\star)"', "\phi'_B"] \arrow[ld, dashed, shift left] \\
\begin{array}{c} E_0/\langle B \rangle \\ \phi_B(P_A), \phi_B(Q_A) \end{array} \arrow[r, "\ker(\phi_A) = \langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle", "\phi'_A"'] \arrow[ru, dashed, shift left] & \frac{E/\langle A \rangle}{\langle \phi'_A(B) \rangle} \cong E_0/\langle A, B \rangle \cong \frac{E/\langle B \rangle}{\langle \phi'_B(A) \rangle}
\end{tikzcd}$$

$(\star)$ $\ker(\phi_B) = \langle [m_B]P_B + [n_B]Q_B \rangle$

$(\star\star)$ $\ker(\phi'_B) = \langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$

Figure 2.2: Commutative representing the SIDH protocol, inspired from [DJP14, Figure 1]. Alice is represented in blue and Bob is represented in red. The dotted blue (resp. red) line represents that Alice (resp. Bob) "moves" to the curve $E_0/\langle B \rangle$ (resp. $E_0/\langle A \rangle$). From there she (resp. he) will compute the last part of the diagram, ϕ'_A (resp. ϕ'_B).

in $O(\ell_A^{e_A/3})$ queries to a quantum oracle [Tan09]. Since the size of the supersingular isogeny graph is $O(p)$, but the secret isogenies ϕ_A and ϕ_B are of degree $O(\sqrt{p})$, Alice and Bob take random walks that are shorter than the diameter of the graph. Therefore, the number of collisions that we expect to find when trying to recover Alice's secret isogenies is exactly one, i.e. the targeted isogeny $\phi_A : E_0 \to E_A$.

For this reason, if we let λ be the security parameter, $p \approx 2^\lambda$ offers a classical security of $\approx \lambda/4$ bits and a quantum security of $\approx \lambda/6$ bits. Thus, to obtain 128-qubit and 192-bit security, SIDH requires a prime p of 768 bits of the form $p = \ell_A^{e_A} \ell_B^{e_A} f \mp 1$ with $\log \ell_A^{e_A} \sim \log_2 \ell_B^{e_B} \sim 384$. In practice, concrete instantiations of SIDH had $\ell_A = 2$ and $\ell_B = 3$ (or vice versa), for efficiency reasons, and a possible prime for this choice of ℓ_A and ℓ_B would be $p = 2^{387}3^{242} - 1$.

2.3 Algorithmic aspects

This section's goal is to provide an overview of all the optimizations required to achieve a compact and fast implementation of the SIDH protocol. In order to make the SIDH key exchange competitive compared to the other post-quantum candidates, the authors in [DJP14, §4] and [Jao+17] focused on each of the following:

- The arithmetic over $\mathbb{F}_p$ takes advantage of primes of the form $p = 2^a 3^b - 1$, as explained in [CLN16];
- The arithmetic over $\mathbb{F}_{p^2}$ takes advantage of the fact that -1 is not a square in $\mathbb{F}_p$, since $p \equiv -1 \mod 2$;
- The arithmetic of elliptic curves takes advantage of Montgomery models. Optimized formulas for doublings, triplings, computing scalar multiplications and evaluating isogenies are employed [DJP14; CLN16];
- One can avoid to perform field inversions through the use of projective coordinates and projectivized curve equations [CLN16];
- To compute and evaluate the secret isogenies given a generator of the kernel, a quasi-linear algorithm must be employed [DJP14].

Even though the SIDH scheme was made quite practical, as shown in [DJP14; CLN16], it was still one or two orders of magnitude slower than other post-quantum alternatives. However, the scheme really excelled in the sizes of the keys, the shortest among all post-quantum candidates at the time of the NIST competition, that could have been reduced even more through several key compression techniques [Aza+16; Zan+18; Cos+17].

Let us provide some specific algorithms to implement what has been discussed so far in an efficient manner, starting from the parameter generation, then discussing how to compute the kernel and the isogenies themselves. Lastly, we discuss how the choice of a specific model can allow to perform these computations more efficiently. These topics are presented in more detail in [DJP14, §4].

To generate public parameters, firstly pick a prime number $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ and find a supersingular curve E over $\mathbb{F}_{p^2}$ such that $\#E(\mathbb{F}_{p^2}) = (p \mp 1) = (\ell_A^{e_A}\ell_B^{e_B}f)^2$ through Bröker's algorithm [Brö09]. These first two steps are easy to perform. Ideally one would want to move from E to E_0 to ensure that the elliptic curve we start the protocol from is randomly chosen. This can be performed through a random walk on an isogeny graph. However, one can also take $E_0 = E$. Regardless, the curve E_0 will have the same desired group structure as E. At this point, to select a basis $\{P_A, Q_A\}$ for the torsion subgroup $E_0[\ell_A^{e_A}]$, Alice will proceed as follows. Alice chooses a random point $P \in_\$ E_0(\mathbb{F}_{p^2})$ and computes $P' = P \cdot (\ell_B^{e_B}f)^2$, that will have order $\ell_A^{e_A}$ with overwhelming probability. Alice can verify that this is the case by checking that $\ell_A^{e_A} \cdot P' = \mathcal{O}$. The point Q_A can be generated analogously. Alice checks that P_A and Q_A are independent by computing a Weil pairing and verifying that the point $R_A = e(P_A, Q_A) \in E_0[\ell_A^{e_A}]$ has order $\ell_A^{e_A}$. If this is not the case, Alice chooses another Q_A, until she has a basis $\{P_A, Q_A\}$ of $E_0[\ell_A^{e_A}]$. As a concluding remark, the choice of the basis does not affect the security of the scheme since it is possible to move from one basis to another using extended discrete logarithm computations[10], that are easy to compute in $E_0[\ell_A^{e_A}]$, with ℓ_A being a small prime [Tes99].

Once the public parameters are generated, the key exchange is performed in two rounds, during which Alice and Bob perform the following steps on each side:

1. Compute $\langle R \rangle = \langle [m]P + [n]Q \rangle$ for some points P and Q;
2. Compute the isogeny $\phi : E \to E/\langle R \rangle$ for a given curve E;
3. (Only in the first round) compute $\phi(U)$ and $\phi(V)$ for some points U and V.

Note that the curve E, the points P, Q, R, U, V, and the integers m, n depend on the round and the party involved, as shown Figure 2.2.

Let us see how each of the three steps can be implemented efficiently, starting by discussing how to obtain the generating kernel and, briefly, how to compute and evaluate a smooth degree isogeny. For what concerns the kernel, we first observe that we are looking for any generator of the subgroup $\langle R \rangle = \langle [m]P + [n]Q \rangle$. Without losing in generality, we assume that m is invertible modulo the order of the group and therefore we consider $R' = P + [m^{-1}n]Q$ to be a generator of the kernel. The computation of $R' = P + [m^{-1}n]Q$ by a standard double-and-add method is two times faster than computing $R = [m]P + [n]Q$ naively, but it has the major issue of being vulnerable to a side-channel attack called "simple power analysis" (SPA) [KJJ99]. Thus, in order to avoid SPA, a Montgomery ladder [Mon87] could be used to compute $[m^{-1}n]Q$ and then add it to P. However, this process is significantly slower than what the authors propose in [DJP14, Algorithm 1], that leverages the very efficient differential additions on Montgomery curves [Mon87], being as efficient as the naive double and add on twisted Edwards curves.

Let us mention how Alice and Bob can compute and evaluate the isogenies. Let E be a supersingular elliptic curve, and let R be a point of order ℓ^e, for some exponent e. We have the goal of describing how it is possible to compute the image curve $E/\langle R \rangle$ and to evaluate the ℓ^e-isogeny $\phi : E \to E/\langle R \rangle$ at some points of E. Firstly, we notice that we can decompose ϕ as a chain of e sequential ℓ-isogenies, that will be easier to handle since $\deg \phi$ is smooth. Indeed, such chain would be defined as follows: set $E_0 = E$, $R_0 = R$ and let

$$E_{i+1} = E_i/\langle \ell^{e-i-1}R_i \rangle, \quad \phi_i : E_i \to E_{i+1}, \quad R_{i+1} = \phi_i(R_i), \quad \forall\, i \in \{0, \dots, e-1\}.$$

[10] Given $\alpha, \beta \in G$, determine the least positive integer y such that $\beta^y \in \langle \alpha \rangle$. Output the pair (x, y), where $x = \log_\alpha \beta^y$. This is a generalization of the regular discrete logarithm problem.

With this construction, we would have $E/\langle R\rangle = E_e$ and $\phi = \phi_{e-1} \circ \cdots \circ \phi_0$.

Naively, the curve E_{i+1} and the isogeny ϕ_i can be computed through the use of Vélu's formulas [Vél71] if the ℓ-torsion subgroup $\langle R_i\rangle$ of E_i is known. Therefore, we have methods with quadratic complexity in e, as described in [JD11]. However, the authors notice that it is possible to do much much better through the use of a formalization of the problem that leverages the structure provided by a discrete equilateral triangle. The details can be found in [DJP14, §4.2.2].

Furthermore, to obtain better performances, at each step of the protocol it is important to use specific models to represent elliptic curves that offer the fastest formulas for doublings, triplings, computing scalar multiplications and evaluating isogenies. To measure efficiency, we count the number of elementary operations over $\mathbb{F}_{p^2}$ and consider the quantities I, M, S for the costs of inversion, multiplication and squaring, respectively. We assume that $S \leq M \leq I$, we neglect additions, subtractions and comparisons and account for multiplications by a constant just as if they were ordinary multiplications. In order to make the computations of every step as efficient as possible, we move between different models of elliptic curves. Note that this is possible because any curve that is used in SIDH has group structure $(\mathbb{Z}/(p \pm 1)\mathbb{Z})^2$ and is thus isomorphic to both a twisted Edwards curve and a Montgomery curve [Ber+08]. Therefore, points in projective coordinates can be moved from one model to the other by performing a few multiplications (and no inversions) as stated by the authors in [DJP14, §4.3.1].

Recall that twisted Edwards curve is defined as $E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$ and there are very efficient methods to compute addition and doubling formulas (using projective coordinates) with $11M + S$ and $3M + 4S$ operations respectively when one of the points is scaled to have Z-coordinate equal to 1. A Montgomery curve is defined as $M_{B,A} : By^2 = x^3 + Ax^2 + x$ and there are very efficient arithmetic operations on their Kummer line, where we represent the points by the coordinates $(X : Z)$ where $x = X/Z$ and scale them to have Z-coordinate equal to 1. In this case, doubling can be done using $2M + 2S$ and differential addition on the Kummer line can be performed using $3M + 2S$, assuming that one of the points is scaled to have Z-coordinate equal to 1.

Let us present an idea of how twisted Edwards curves are used for computing the kernel and how Montgomery curves are used to compute and evaluate isogenies. To compute the kernel subgroup $\langle [m]P + [n]Q\rangle$, express the points P and Q in projective Edwards coordinates and perform a "double-and-add method" followed by an addition. If we take care of scaling Q to have Z-coordinate equal to 1, this computation can be performed in $9.5M + 4.5S$ per bit on average.

This can also be done through Kummer coordinates over a Montgomery curve by using a "ladder method" and, if we scale P, Q and $P - Q$ to have Z-coordinate equal to 1, we can perform the computation in $9M + 6S$. Note that, since the cost of one squaring is similar to the cost of one multiplication ($S \sim M$), the "ladder algorithm" is slightly slower than the "double-and-add" method, but the "ladder method" offers the two advantages of providing SPA resistance and simplifying the implementation by not using Edwards coordinates at all.

For what concerns the isogeny evaluation, explicit formulas for isogenies between Montgomery curves are provided in [DJP14, §4.3.2], where the authors optimize the cases of isogenies of degree 2 and 3, that are the ones typically used for SIDH. With the intent of obtaining the most efficient formulas for evaluating isogenies, the authors attempt to avoid inversions and square root computations as much as possible.

Let G be a subgroup of the Montgomery curve E of odd cardinality ℓ. Through Vélu's formulas we can obtain an isogeny $\phi : E \to E/G$, that can be evaluated on a point in full projective coordinates using $11M + 2S$, or $4M + 2S$ using Kummer coordinates. If, instead, ℓ is even, an isogeny can be evaluated on a point in full projective coordinates using $5M+3S$ operations, or $2M+S$ using Kummer coordinates, avoiding the computation of expensive square roots, only if an 8-torsion point is known. On the other hand, if ℓ is even and we don't know an 8-torsion point, a properly crafted isogeny can be evaluated on a point in full projective coordinates using $10M + 4S$ operations, or $4M + S$ operations using Kummer coordinates. Unfortunately, the application of this latter method twice sequentially yields the scalar multiplication-by-4 isogeny.

It might be interesting to mention that after a generator R of the kernel of the isogeny has been computed, the authors notice that the algorithm does not use its y-coordinate at all and therefore they completely remove it and only use scalar multiplication and isogeny evaluation formulas for points on the Kummer line.

Moreover, isogenies of composite smooth degree are computed by composing small degree isogenies, but more attention must be placed when computing cyclic isogenies of degree 2^e. The authors present two strategies to evaluate this type of isogenies: either use 2-isogenies as much as possible and in the end use one 4-isogeny for the last two steps, or use one degree 2 isogeny if e is odd and then use only 4-isogenies composed with isomorphisms.

The two approaches should take roughly the same computational effort and both, if implemented as described in [DJP14, §4.3.2], leak two bits of security. Even though this could be easily mitigated by taking a random change of coordinates, this makes the process more costly than simply adding two extra bits of security and letting the information leak.

The complexities of evaluating an isogeny and computing a multiplication are compared in Table 2.3, where it is estimated that $S = 0.8M$ in $\mathbb{F}_{p^2}$, since squaring requires 2 multiplications in $\mathbb{F}_p$ instead of 3. The most outstanding difference is between evaluating 2^e-isogenies and 3^e-isogenies, suggesting that degree 2^e-isogenies are preferable for constraint devices.

ℓ	2	3	4
Isogeny	$2M + S$ 2.8	$4M + 2S$ 5.6	$6M + S$ 6.8
Multiplication	$3M + 2S$ 4.6	$7M + 4S$ 10.2	$6M + 4S$ 9.2

Table 2.1: Comparison of computational costs for isogeny evaluation and multiplication in projective Kummer coordinates, assuming $S = 0.8M$. Taken from [DJP14, Table 1].

2.4 Hardness assumptions

The security of SIDH can be formally stated as a hardness assumption on a problem called SSDDH, whose best solving algorithm was believed to have exponential complexity, even on a quantum computer. This assumption will be proven wrong by the SIDH attacks [CD23; Mai+23; Rob23a] that will be covered throughout Chapter 3. Nevertheless, the problems presented within the present section are of general interest for isogeny-based cryptography, not necessarily only restricted to the scope of SIDH.

Let p be a prime of the form $p = \ell_A^{e_A}\ell_B^{e_B} f \mp 1$. Let E_0 be a supersingular curve over $\mathbb{F}_{p^2}$. Let $\{P_A, Q_A\}$ be a basis of $E_0[\ell_A^{e_A}]$ and $\{P_B, Q_B\}$ be a basis of $E_0[\ell_B^{e_B}]$. Let us present the following computational problems for the case of isogenies over supersingular elliptic curves.

Problem 1 (Decisional Supersingular Isogeny (DSSI) problem)**.** *Let E_0 be a supersingular curve over $\mathbb{F}_{p^2}$ and let E_A be a different supersingular curve defined over $\mathbb{F}_{p^2}$. Decide whether E_A is $\ell_A^{e_A}$-isogenous to E_0.*

Problem 2 (Computational Supersingular Isogeny (CSSI) problem)**.** *Let $\phi_A : E_0 \to E_A$ be an isogeny such that* $\ker \phi_A = \langle [m_A]P_A + [n_A]Q_A \rangle$ *for $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and not both divisible by $\ell_A^{e_A}$. Given E_A and the images $\phi_A(P_B), \phi_A(Q_B)$, find a generator R_A of the subgroup $\langle [m_A]P_A + [n_A]Q_A \rangle$*

The authors also highlight that given a generator $R_A = [m_A]P_A + [n_A]Q_A$, it is easy to solve for (m_A, n_A), due to the fact that E_0 has smooth order and extended discrete logarithms can be computed easily over E_0 [Tes99]. Furthermore, since recovering the generator of the kernel of the $\ell_A^{e_A}$ isogeny $\phi_A : E_0 \to E_A$ is equivalent to actually recovering the isogeny itself, Problem 2 can be re-formulated as done in Problem 3, for a slightly more specific instance. Most importantly, notice that Problem 3 focuses on recovering Bob's secret isogeny ϕ_B, while Problem 1 and Problem 2 focus on recovering Alice's secret isogeny. Defining the problem as follows is useful for two main reasons: we will recall this notation when discussing the attack to SIDH presented in [Mai+23], and we will be able to better compare the securities of the SIDH scheme and the M-SIDH variant, discussed in Section 4.1.1.

Problem 3 (Computational Supersingular Isogeny with Torsion (CSSI-T) problem)**.** *Let d_A and d_B be two coprime integers. Let $E_0/\mathbb{F}_{p^2}$ and $E_B/\mathbb{F}_{p^2}$ be two supersingular curves with $p = d_A d_B - 1$. Let $E_0[d_A] = \langle P_A, Q_A \rangle$. Let $\phi_B : E_0 \to E_B$ be an isogeny of degree d_B and let $P' = \phi_B(P_A), Q' = \phi_B(Q_A)$. Given $E_0, P_A, Q_A, E', P', Q'$, compute the d_B-isogeny ϕ_B.*

The security of SIDH was formalized through a computational and a decisional problem called Supersingular Computational Isogeny Diffie-Hellman (SSCDH) and Supersingular Decision Isogeny Diffie-Hellman (SSDDH) [DJP14, Problem 5.3, Problem 5.4] respectively. However, since the security of SIDH was badly broken by the aforementioned SIDH attacks, introducing these problems is of limited interest. Instead, Problem 1 and Problem 2 are introduced in the present section because all the schemes that built on top of the framework introduced by the SIDH protocol relied on their assumed hardness. Indeed schemes such as k-SIDH [AJL18], B-SIDH [Cos20a], SÉTA [De +21], SHealS [FP21] and SIKE [Jao+17], that reveal the images of the torsion basis under a secret isogeny of known degree, heavily relied on the security analysis provided for SIDH in [DJP14, §6]. Ideally, the authors in [DJP14] hoped that the best algorithm for solving Problem 1 and Problem 2 was the one previously mentioned, that uses the meet-in-the-middle paradigm to solve the specific instance of the *claw problem.* However, this was not the case since the additional information provided by the images of the torsion points, as well as the degree of the isogeny was exploited by the SIDH attacks [CD23; Mai+23; Rob23a], definitely showing that Problem 2 is simpler to solve that the *pure isogeny problem* (i.e. the so-called "*path finding problem*") as had already been discussed by [Pet17] before the publication of the devastating SIDH attacks [CD23; Mai+23; Rob23a].

We will conclude the present section by briefly mentioning that in the context of cryptography, the hard problem of computing an isogeny between two supersingular curves (the so-called *path finding problem*) was first taken into account in [Gal99] and the first cryptographic primitive that used supersingular isogeny graphs was the CGL hash function proposed in [CLG09], that relies on the hardness of the *path finding problem.* Indeed, this problems is of general interest for several constructions in isogeny-based cryptography and the fastest algorithm for finding isogenies between supersingular curves has a run-time of $O(\sqrt{p}\log^2 p)$ operations [CLG09, §5.3.1]. The latter algorithm refers to the case where no additional information, such as the images of the torsion points or the degree of the isogeny, is provided. One must keep in mind that in the case of SIDH, the degree of the isogeny, that is smooth, and the images of the torsion points are known in advance by an attacker and, although the authors in [DJP14] stated that it did not appear to be any way to use such information to determine one of the secret isogenies, the SIDH attacks [CD23; Mai+23; Rob23a] were indeed able to take advantage of it.

Chapter 3

Breaking SIDH in polynomial time

The present chapter serves as a high-level exposition of the series of devastating attacks to the SIDH protocol published during the Summer of 2022 [CD23; Mai+23; Rob23a], that were able to break any instance of SIDH in polynomial time. Throughout the rest of the survey, we will refer to these attacks as the "SIDH attacks" and they will be used as black-boxes when arguing the security of the countermeasures, such as M-SIDH & MD-SIDH [FMP23], and also when presenting ways in which these attacks were used in a constructive manner to obtain new protocols, like the cases of FESTA and POKE [BMP23; Bas24]. Therefore, the actual (gory) details are not necessary to understand the final chapters of the survey and are also outside its scope.

Before the Summer of 2022, the SIDH protocol and the derived key encapsulation mechanism SIKE were considered very promising since the most significant attacks against SIKE were variants of the meet-in-the-middle approach (that require exponential complexity). Moreover, attacks to SIDH that exploited the information provided by the torsion points, the so-called "torsion point attacks", applied exclusively to unbalanced sets of SIDH parameters (and not to SIKE) [Pet17]. Therefore, the SIKE algorithm was considered very promising, especially given its very compact secret and public keys (typical of most isogeny-based schemes).

However, in August 2022 several new attacks exploiting the torsion point information, published by Alice to help Bob close the commutative diagram, were published and the security picture drastically changed. At the core of these attacks there was the idea of embedding the targeted secret isogeny into a higher dimensional isogeny, that allowed to break the case of balanced parameters and forced the authors of the SIKE algorithm to declare it as fully broken both in theory and in practice[11].

As already mentioned towards the end of the previous chapter, the security of SIDH does not rely on the *pure isogeny problem* (i.e. the *isogeny path finding problem*), but on some variant of it. This is due to the fact that revealing some additional non-trivial information, such as the degree of the isogeny and the images of the torsion points, is necessary for the protocol to work. Moreover, in SIDH the starting curve E_0 is chosen such that its endomorphism ring is publicly available. By exploiting this information, it had already been possible to design an attack that recovered the secret isogeny in polynomial time in the case of unbalanced parameters for which the degree of one isogeny is much greater than the other even before the Summer of 2022 [Pet17]. Even though this attack was further improved in [Que+21], it still did not apply to the case of SIDH, where the degrees of the isogenies are approximately the same.

Therefore, in the Summer of 2022, the SIDH attacks presented several methods to recover the secret key in SIDH and SIKE, instantiated with the NIST parameters (not unbalanced), in a few hours [CD23] and then in a few minutes after vaious follow-ups to the first attack [Mai+23]. Therefore, the complexity of attacking SIDH went from exponential time, as the authors of SIKE thought, to subexponential time, with further improvement to polynomial time when the endomorphism ring of the starting curve is available (as it was for SIKE) [CD23; Mai+23]. Briefly after the publication of these two attacks, a third improved attack with polynomial time complexity, that applied for arbitrary starting curves, was published [Rob23a] and made SIKE (and all SIDH-variants) irreversibly insecure.

However, it is worth mentioning that the devastating SIDH attacks do not apply to other isogeny-

[11] The statement made by the authors to NIST can be found here.

based schemes in which no torsion point information is revealed, such as: CRS [Cou06; RS06], CSIDH [Cas+18] and CSIDH-based signatures (such as SeaSign [DG19] and CSI-FiSh [BKV19]), CGL [CLG09], GPS [GPS17] and SQISign [De +20]. Therefore, SIDH being completely broken does not mean that the whole isogeny-based cryptography is insecure, but only that the protocols that publish the image of the torsion points should be avoided.

In what follows, let $d_A = 2^a$ and $d_B = 3^b$, as in SIDH, be the degrees of the secret isogenies $\phi_A : E_0 \to E_A$ and $\phi_B : E_0 \to E_B$. However, in cases where context makes it clear, we will omit the subscripts, since any attack can target either of the two secret isogenies. On the other hand, if the subscripts are not omitted, the reader must keep in mind that the same attack could be performed analogously to recover the other secret isogeny, since in SIDH Alice and Bob have symmetric roles.

Before diving into a brief description of the attacks, it is worth mentioning the so-called "lollipop attack", first presented in [Pet17], since it was the first attack exploiting torsion point information. The main idea of the attack that allowed to recover the secret isogeny $\phi_B : E_0 \to E_B$ was to consider the endomorphism

$$\psi = [t] + \phi_B \circ \theta \circ \hat{\phi}_B$$

over E_B, where $t \in \mathbb{Z}$ and θ is a non-scalar endomorphism over E_0 such that

$$\deg \psi = t^2 + d_B^2 \cdot \deg \theta$$

divides d_A. After retrieving torsion point information about ψ through the torsion point information on ϕ_B, it is possible to recover $\ker \psi$ via a discrete logarithm computation (that is efficient since the degrees are smooth), and finally obtain ϕ_B. However, as already mentioned, crucial assumptions for this attack to work are the (at least partial) knowledge of the endomorphism ring of E_0 (so that the non-scalar endomorphism θ is known) and having degrees such that $d_B \ll d_A$ (in particular, $d_B < \sqrt{d_A}$). Due to the latter requirement, this attack could not work for SIKE, where $d_B \approx d_A$.

The new SIDH attacks published during the Summer of 2022 can be seen as generalizations of previous torsion point attacks, with the additional idea of embedding the targeted secret isogeny into a higher-dimensional isogeny, since in higher dimensions more endomorphisms can be used more easily.

3.1 Key recovery knowing the endomorphism ring

The authors in [CD23] present a method to recover one of the participants' secret isogeny by exploiting the information provided by the torsion point images that Alice and Bob exchange. The presented method is based on Kani's "reducibility criterion" and is able to achieve classical polynomial runtime if the endomorphism ring of the starting curve is known, and classical subexponential runtime otherwise. In particular, the attack is fast and easy to implement if one of the parties uses 2-isogenies and if there exists a non-scalar endomorphism of very small degree over the starting curve. These two assumptions were verified in the case of SIKE. Thus, it was possible to break `SIKEp751` (that aims at 256-bit security) in approximately 3h15m on a single core through the Magma [BCP97] implementation that can be found at

https://homes.esat.kuleuven.be/~wcastryc/.

3.1.1 Background on higher-dimensional isogenies

Before presenting some details of the attack presented in [CD23], let us take a step back and mention that it is possible to generalize the concept of elliptic curves to higher dimensions.

Elliptic curves are *abelian varieties* of dimension 1, but in dimension 2 this translates to the concept of *abelian surfaces*, that have a group structure and are equipped with a commutative algebraic group operation. In particular, while we are interested in supersingular elliptic curves in dimension 1, in the dimension-2 case we deal with *superspecial principally polarized abelian surfaces*[12], that will often be called *abelian surfaces* for simplicity.

[12] Principally polarized abelian surfaces can be written through the equation $C : y^2 = x^5 + Ax^3 + Bx^2 + Cx + D$.

A given abelian surface A can be either a product of two elliptic curves $E \times E'$ or a *Jacobian of a genus-2 curve* C, that is a specific type of abelian surface that will be denoted as $\mathrm{Jac}(C)$ (cf. [FS19] for more details).

The generalization also extends to the concept of isogenies: a (N, N)-isogeny $\Phi : A \to A'$ is an isogeny such that $\ker \Phi \cong \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ and, at the same time, $\ker \Phi$ is maximal isotropic with respect to the N-Weil pairing, i.e., $\forall\ P, Q \in \ker \Phi,\ e_N(P, Q) = 1$. Note that the latter condition ensures that the codomain abelian surface is principally polarized. This isotropic condition is crucial for maintaining the principal polarization on the abelian surface after the isogeny. Isogenies between higher-dimensional abelian varieties, also referred to as "higher-dimensional isogenies", introduce a crucial structure for various applications in isogeny-based cryptography.

Furthermore, there are four types of isogenies between abelian surfaces:

1. $\mathrm{Jac}(C) \to \mathrm{Jac}(C')$: the *generic* (N, N)-isogeny;

2. $\mathrm{Jac}(C) \to E_1' \times E_2'$: the *split* (N, N)-isogeny;

3. $E_1 \times E_2 \to \mathrm{Jac}(C')$: *gluing* elliptic curves along their (N, N)-torsion;

4. $E_1 \times E_2 \to E_1' \times E_2'$: (N, N)-isogeny between products of elliptic curves.

When referring to (N, N)-isogenies *from* products of elliptic curves, that is of types 3 and 4, we will consider the (N, N)-isogeny

$$\Phi : E_1 \times E_2 \to A'$$

having non-diagonal kernel[13] and state that it is not an (N, N)-gluing, i.e. $A' \cong E_1' \times E_2'$, with probability $\approx \frac{10}{p}$ for a superspecial abelian surface. Therefore, when considering a higher-dimensional isogeny having domain a product of two elliptic curves, we expect its codomain to be a Jacobian of a genus-2 curve with overwhelming probability. This is due to the fact that there are $\approx \frac{p^3}{2880}$ superspecial Jacobians of genus-2 curves, while there are only $\approx \frac{p^2}{288}$ superspecial products [Bro93].

3.1.2 Strategy through Kani's criterion

The attack presented in [CD23] is based on a "reducibility criterion" due to Kani [Kan97, Theorem 2.6] for determining whether an isogeny having domain a product of two elliptic curves has also as codomain a product of elliptic curves, rather than a Jacobian of a genus-2 curve as one would expect. In other words, we want to check whether Φ, that is the higher-dimensional embedding of the secret isogeny ϕ, is of the type $\Phi : E_1 \times E_2 \to E_1' \times E_2'$. We have already mentioned that this occurs with incredibly small probability, but Kani's criterion helps us understand under which circumstances this happens.

Let us introduce Kani's criterion in Theorem 4 by first introducing the notion of *isogeny diamond configuration* in Definition 18. Note that these concepts are presented in a very informal manner, since further formalizing them is beyond the scope of the survey.

Definition 18 (Isogeny diamond configuration, very informal)**.** *An* isogeny diamond configuration *of order N is a tuple (ψ, G_1, G_2) where $\psi : E \to E'$ is an isogeny and $G_1, G_2 \subset \ker(\psi)$ are two subgroups such that we have the following three:*

1. $G_1 \cap G_2 = \{\mathcal{O}_E\}$;

2. $\deg \psi = \#G_1 \cdot \#G_2$;

3. $N = \#G_1 + \#G_2$.

Theorem 4 (Kani's criterion, very informal)**.** *An (N, N)-gluing* fails*, i.e. the codomain of the isogeny is a product of elliptic curves, if and only if it comes from an isogeny diamond configuration, i.e. the kernel of the isogeny is of the form $\langle (P, [x]\psi(P)), (Q, [x]\psi(Q)) \rangle$ for some $x \in \mathbb{Z}$.*

[13] The kernel of an isogeny $\Phi : E_1 \times E_2 \to A'$ is said to be diagonal if it can be written as $\ker \Phi = \langle (P, \mathcal{O}_{E_2}), (\mathcal{O}_{E_1}, Q) \rangle$. Note that the kernel of an (N, N)-isogeny between products of elliptic curves (type 4) is always maximal isotropic with respect to the N-Weil pairing and it is always diagonal.

Examples of "failed gluing" between the domain and the codomain of an higher-dimensional isogeny are the following:

- A $(2,2)$-isogeny $\Phi : E_1 \times E_2 \to A'$ with non-diagonal kernel can only have $A' \cong E_1' \times E_2'$ if $E_1 \cong E_2$;
- A $(3,3)$-isogeny $\Phi : E_1 \times E_2 \to A'$ with non-diagonal kernel can only have $A' \cong E_1' \times E_2'$ if there exists a 2-isogeny $\psi : E_1 \to E_2$;
- A $(5,5)$-isogeny $\Phi : E_1 \times E_2 \to A'$ with non-diagonal kernel can only have $A' \cong E_1' \times E_2'$ if there exists a 4- or 6-isogeny $\psi : E_1 \to E_2$;
- A $(7,7)$-isogeny $\Phi : E_1 \times E_2 \to A'$ with non-diagonal kernel can only have $A' \cong E_1' \times E_2'$ if there exists a 6- or 10- or 12-isogeny $\psi : E_1 \to E_2$.
- ...

The attack targets Bob's secret 3^b-isogeny $\phi : E_0 \to E$ that connects two supersingular elliptic curves. Moreover, recall that the curve E_0, the points $P_0, Q_0 \in E_0$ of order 2^a, and the exponents a and b are public parameters. Note that the authors in [CD23] mainly focus on the cases where E_0 is one of the curves defined by the equations

$$y^2 = x^3 + x, \qquad y^2 = x^3 + 6x^2 + x, \tag{3.1}$$

that are supersingular in characteristic $p \equiv 3 \mod 4$ and are the curves proposed in the first and last three rounds of the NIST standardization effort, respectively. However, in view of the work of Love and Boneh [LB20] combined with the KLPT algorithm [Koh+14], any starting curve whose endomorphism ring is known can be broken in polynomial time. In the case of unknown endomorphism ring, more work needs to be done to apply the attack and therefore it runs in subexponential time. Additionally, recall that Bob's public key is composed of the codomain curve E and the image of the torsion points $\phi(P_0), \phi(Q_0)$.

The idea at the core of the attack is that landing on a product of elliptic curves (i.e. having a higher-dimensional isogeny of type 4) is extremely unlikely if $E, \phi(P_0), \phi(Q_0)$ do *not* constitute a valid public key triple. In this sense, Kani's criterion (Theorem 4) is used as a a decision tool: if the codomain of the higher-dimensional isogeny is a product of elliptic curves, then we say that the test *passed* and we can gather some information on the secret isogeny ϕ. On the contrary, if the codomain of the higher-dimensional isogeny is a Jacobian of a genus-2 curve, we say that the test *failed.*

The attack will proceed iteratively by guessing a portion of the secret isogeny at every step and applying the same test several times until it passes, determining whether the guess was correct or not. For the correct guess, we will be able to retrieve some information about the secret isogeny ϕ. After gathering enough information, i.e. applying the test to several parts of the secret isogeny, it will be possible to recover the entire secret isogeny ϕ.

However, in order to use Kani's criterion as a test for determining whether the guessed fraction of the secret isogeny is correct or not, it is necessary to resort to a trick to force an *isogeny diamond configuration*, by considering an auxiliary isogeny γ.

3.1.3 The attack

Let us describe the actual attack performed by Castryck and Decru [CD23] by first delineating the actual set-up considered, then adding some details on how to use Kani's criterion as a decision tool and lastly by detailing the algorithm followed by the attack to recover Bob's secret isogeny $\phi : E_0 \to E$.

Setup and overview

The set-up required for considering the algorithm that returns the secret isogeny ϕ is the following:

1. An SIDH prime p of the form $p = 2^a 3^b f - 1$ for integers $a \geq 2, b, f \geq 1$ such that $2^a \approx 3^b$;

2. An elliptic curve $E_0/\mathbb{F}_{p^2}$ such that $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (2^a 3^b f)^2$;
3. A basis $\{P_0, Q_0\}$ of $E_0[2^a]$;
4. The codomain $E/\mathbb{F}_{p^2}$ of a secret cyclic 3^b-isogeny $\phi : E_0 \to E$;
5. A basis $\{P, Q\}$ of $E[2^a]$, where $P = \phi(P_0)$ and $Q = \phi(Q_0)$.

Moreover, two further technical conditions are imposed:

- Assume $2^a > 3^b$;
- Let $c = 2^a - 3^b$ and assume it is possible to compute $P_c = \gamma(P_0)$ and $Q_c = \gamma(Q_0)$ under the action of a well crafted c-isogeny $\gamma : E_0 \to C$.

Let $x \in \mathbb{Z}$ denote a multiplicative inverse of $3^b \mod 2^a$ and $-x$ be the multiplicative inverse of c mod 2^a. Further discussion on these technical details can be found in [CD23]. We have the following configuration:

$$\begin{tikzcd}
\{P_0, Q_0\} & \{P, Q\} \\[-2em]
E_0 \arrow[r, "\phi"] \arrow[d, "\gamma"'] & E \\
C & \\[-2em]
\{P_c, Q_c\} &
\end{tikzcd}$$

Figure 3.1: Diagram representing the setup for the attack.

Through Kani's criterion, we want to decide whether the following holds true or not:

$$\text{There is a } 3^b\text{-isogeny } \phi : E_0 \to E \text{ such that } \phi(P_0) = P \text{ and } \phi(Q_0) = Q. \tag{3.2}$$

In order to do this, the authors consider the $c3^b$-isogeny

$$\psi = [-1] \circ \phi \circ \hat{\gamma} : C \to E,$$

where $\psi(P_c) = -cP$ and $\psi(Q_c) = -cQ$. For reasons tied to Weil-pairing computations, we have that the subgroup

$$\langle (P_c, [x]\psi(P_c)), (Q_c, [x]\psi(Q_c)) \rangle = \langle (P_c, P), (Q_c, Q) \rangle \tag{3.3}$$

for some $x \in \mathbb{Z}$ is maximally isotropic with respect to the 2^a-Weil pairing on the product $C \times E$. Therefore, the subgroup $\langle (P_c, P), (Q_c, Q) \rangle$ concerns the kernel of a $(2^a, 2^a)$-isogeny between principally polarized abelian surfaces and, as stated by the authors in [CD23, §4.1], it can be viewed as a walk of length a in the $(2,2)$-isogeny graph of *superspecial principally polarized abelian surfaces over* $\overline{\mathbb{F}}_p$. This is due to the fact that it is possible to write the $(2^a, 2^a)$-isogeny as a composition of several $(2,2)$-isogenies, just like for the analogous 1-dimensional case.

As already mentioned, the vertices of the $(2,2)$-isogeny graph of *superspecial principally polarized abelian surfaces over* $\overline{\mathbb{F}}_p$ can be of two types: products of supersingular elliptic curves (there are $\approx \frac{p^2}{288}$ of them) or Jacobians of superspecial genus-2 curves (there are $\approx \frac{p^3}{2880}$ of them) [Bro93].

Therefore, most isogenies in the chain are between Jacobians of genus-2 curves, that can be computed through efficient formulae due to Richelot [Smi+05]. However, the authors state that the first step is an exception as it will necessarily result in a "gluing" step, mapping the product $C \times E$ to a Jacobian, with overwhelming probability[14].

[14] It is stated that this can only fail if $C \cong E$, due to a corollary of Theorem 4.

Remark 10. *The isogeny $\gamma : C \to E$ plays the role of forcing us into the exceptional situation where the* last *step of the chain is split, i.e. the codomain of the higher-dimensional $(2^a, 2^a)$-isogeny is again a product of elliptic curves.*

At this point, the decision strategy consists of testing whether the codomain of the higher-dimensional isogeny Φ having domain $C \times E$ and kernel (3.3) is a product of elliptic curves or not, as illustrated in Figure 3.2. If (3.2) does *not* hold, i.e. there is not an isogeny of the desired degree connecting E_0 to E, then the test fails, i.e. the codomain is a Jacobian of a genus-2 curve, *with overwhelming probability.* Equivalently, if the test passes, i.e. the codomain is a product of elliptic curves, then (3.2) is true, i.e. there is an isogeny of the desired degree connecting E_0 to E, *with overwhelming probability.*

We say that these implications are true *with overwhelming probability* because the fact that the fraction of *products of elliptic curves* among all vertices of the $(2, 2)$-isogeny graph is only $\approx \frac{10}{p}$, combined with Kani's criterion, tell us that if the codomain of Φ is a product of elliptic curves, then this was most likely not a coincidence. Indeed, this means that the probability that the test returns a false positive is negligible and we can feel confident with the conclusions that we derive from it.

Therefore, we can conclude that the guess made for a part of the secret isogeny, that is embedded in the higher-dimensional $(2^a, 2^a)$-isogeny Φ, is correct with overwhelming probablity if the test passed, i.e. the $(2^a, 2^a)$-gluing failed. Therefore, to summarize, we have the following implications:

- If the isogeny exists, then the test surely passes;
- Conversely, if the test passes, then the isogeny exists with overwhelming probability.

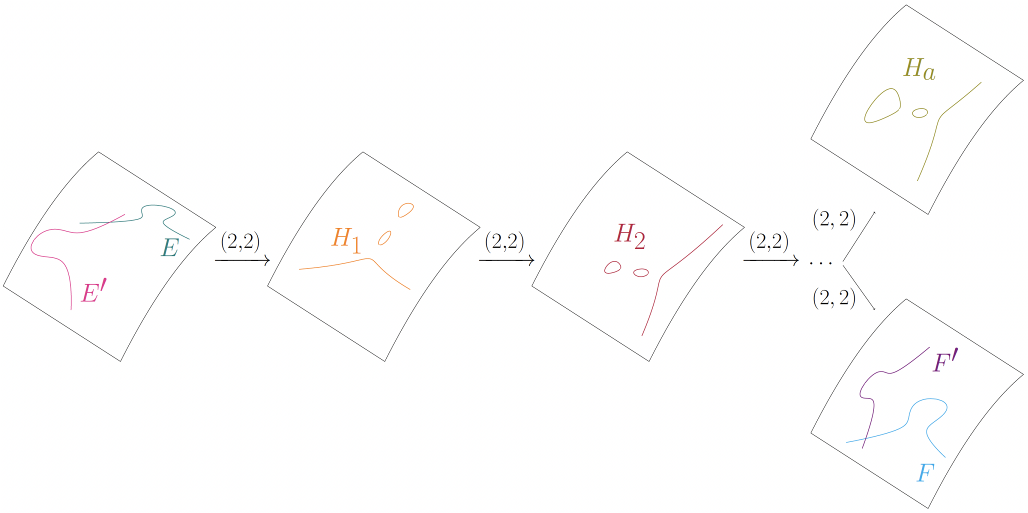

Figure 3.2: Decision strategy based on Kani's reducibility criterion, taken from [CD23, Figure 1].

Iteration

Let us describe the iterative approach to full key recovery adopted by the authors in [CD23, §6].

For the first iteration, choose an integer $\beta \geq 1$ minimal such that there exists $\alpha \geq 0$ for which

$$c_1 = 2^{a-\alpha_1} - 3^{b-\beta_1}$$

is of the form $u_1 + 4v_1^2$. Consider the 3^{β_1}-isogeny $\kappa_1 : E_0 \twoheadrightarrow E_1$ such that $\phi = \phi_1 \circ \kappa_1$. We have the configuration depicted in Figure 3.3.

To an attacker, there are $O(3^{\beta_1})$ a priori options for κ_1. For each of these options, run the decision algorithm having as input the following:

2. The curve $E_1 = \kappa_1(E_0)$;

$$\phi :\ \underbrace{E_0 \xrightarrow{\kappa_1} E_1}_{\substack{\kappa_1 : E_0 \twoheadrightarrow E_1 \\ \deg \kappa_1 = 3^{\beta_1}}} \underbrace{\xrightarrow{\qquad\quad \phi_1 \qquad\quad} E}_{\substack{\phi_1 : E_1 \to E \\ \deg \phi_1 = 3^{b-\beta_1}}}$$

Figure 3.3: Diagram representing the setup for the first iterative step of the attack.

3. A basis $\{P_1, Q_1\}$ of $E_1[2^{a-\alpha_1}]$, where $P_1 = \kappa_1(2^{\alpha_1}P_0)$ and $Q_1 = \kappa_1(2^{\alpha_1}Q_0)$;
4. The codomain E: if the guess is correct, then it is connected to E_1 via the unknown isogeny ϕ_1 of degree $3^{b-\beta_1}$;
5. A basis $\{P_2, Q_2\}$ of $E[2^{a-\alpha_1}]$, where $P_2 = 2^{\alpha_1}P$ and $Q_2 = 2^{\alpha_1}Q$;
6. The 3^{β_1}-isogeny $\hat{\kappa}_1 : E_1 \to E_0$.

Note that the numbering is chosen to be consistent with what was previously mentioned in the setup description, and the additional input consists of the guess made by the attacker. According to what was previously described, among all possible choices of κ_1, only the correct guess for κ_1 is expected to pass the test (this is true with overwhelming probability).

Remark 11. *Let us attempt to explain in simpler terms what happens in the first iterative step of the attack. We make a guess for part of the secret isogeny and call this 3^{β_1}-isogeny $\kappa_1 : E_0 \twoheadrightarrow E_1$, for some intermediate curve E_1 that should be on the path from E_0 to E. We want to know if this choice of κ_1 is correct or not, thus we apply Kani's criterion to verify whether there exists an isogeny $\phi_1 : E_1 \to E$ of the correct degree. Indeed, if the choice of κ_1 is correct, then* $\deg \phi_1 = 3^{b-\beta_1}$. *We have the configuration depicted in Figure 3.4 when applying Kani's criterion to this case.*

$$\begin{tikzcd}
\{P_1, Q_1\} & & \{P_2, Q_2\} \\
E_1 \arrow[rr, "\phi_1"] \arrow[d, "\gamma"'] & & E \\
C & & \\
\{P_c, Q_c\} & &
\end{tikzcd}$$

Figure 3.4: Applying Kani's criterion to the first iterative step of the attack.

At this point, we apply Kani's criterion and conclude that if the codomain of the higher-dimensional isogeny having kernel (3.3) *is a product of elliptic curves, then the choice of the isogeny $\kappa_1 : E_0 \twoheadrightarrow E_1$ is correct with overwhelming probability.*

Therefore, we can assume that we have found the first piece $\kappa_1 : E_0 \twoheadrightarrow E_1$ of the secret isogeny $\phi : E_0 \to E$. Now we continue from E_1 and apply this procedure iteratively. That is, let $\beta_2 > \beta_1$ be the minimal integer such that there exists some $\alpha_2 \geq 0$ for which $c_2 = 2^{a-\alpha_2} - 3^{b-\beta_2}$ is of the form $u_2^2 + 4v_2^2$. Through an analogous process to the previous one, one tries to recover the $3^{\beta_2-\beta_1}$-isogeny $\kappa_2 : E_1 \to E_2$ such that $\phi_1 = \phi_2 \circ \kappa_2$. By continuing in this way, the entire isogeny ϕ can be eventually retrieved.

Let us briefly attempt to rephrase the attack in terms of Bob's secret key $\phi_B : E_0 \to E$, as it is done in [CD23, §6.3]. First notice that, as already mentioned in Section 2.3, Bob's secret isogeny can be encoded as the integer $\mathrm{sk}_B \in [0, 3^b)$ for which $\ker \phi_B = \langle P_B + [\mathrm{sk}_B]Q_B \rangle$. Let us expand

$$\mathrm{sk}_B = k_1 + k_2 3^{\beta_1} + \cdots + k_r 3^{\beta_{r-1}}, \quad \text{with } k_i \in [0, 3^{\beta_i - \beta_{i-1}} - 1)$$

(where $\beta_0 = 0$ without loss of generality). The attack will proceed by guessing one digit at a time. Indeed, intuitively we will have that if $\{P_B, Q_B\}$ is the public basis of $E_0[3^b]$, then

$$\begin{aligned} \ker \kappa_1 &= \langle 3^{b-\beta_1} P_B + k_1 3^{b-\beta_1} Q_B \rangle, \\ \ker \kappa_2 &= \langle 3^{b-\beta_2} P_B + (k_1 + k_2 \cdot 3^{\beta_1}) 3^{b-\beta_2} Q_B \rangle, \\ &\dots \end{aligned}$$

and at each step a trial-and-error procedure is adopted to determine the value of k_i (using Kani's criterion). Subsequently, the same thing is done for k_{i+1}, but with the newly acquired knowledge of k_i. Therefore, after each iteration we determine a digit of sk_B. If $\beta_i = 2$ for all iterations of the algorithm, then we basically determine one base-9 digit of sk_B at a time.

Computational details

Additional insights are provided regarding the construction and evaluation of the auxiliary isogeny γ, the degrees β_i of the isogenies κ_i, the computations of the $(2,2)$-isogenies, and the various methods deployed to speed up key recovery [CD23, §5, §6.2, §8, §7].

The assumption that the points P_c and Q_c can be efficiently computed under a degree c-isogeny is non-trivial and this is where we need the factorization of $c = 2^a - 3^b$. Furthermore, we also rely on the knowledge of the endomorphism ring of E_0, as both curves previously mentioned in (3.1) come equipped with the endomorphism $2\mathbf{i}$ such that $2\mathbf{i} \circ 2\mathbf{i} = [-4]$, upon which the construction of γ relies. This is also the reason for assuming that c is of the form $c = u^2 + 4v^2 = (u + 2\mathbf{i}v)(u - 2\mathbf{i}v)$. Even though there are many other ways for constructing the isogeny γ that do not make use of the endomorphism $2\mathbf{i}$, at least partial knowledge of the endomorphism ring is required in order to compute and evaluate γ in an efficient way [CD23, §5]. Indeed, if the endomorphism ring is unknown, we have to hope that c is smooth and work with arbitrary isogenies over extension fields. Otherwise, it is possible to introduce a limited amount of leeway at the expense of making the attack's runtime subexponential and not polynomial anymore.

A further interesting point of discussion regards the degrees β_i of the isogenies κ_i, in particular the gaps between the consecutive integers $0, \beta_1, \dots, \beta_r = b$. The authors in [CD23, §6.2] state that those differences should be as small as possible, since in this way we reduce the number of possible guesses for each iteration. It is stated that the expected number of $(2^a, 2^a)$-isogenies that needs to be computed is

$$\frac{1}{2}\left(3^{\beta_1} + 3^{\beta_2 - \beta_1} + 3^{\beta_3 - \beta_2} + \cdots + 3^{b - \beta_{r-1}}\right). \tag{3.4}$$

The authors also notice that it is necessary that each $b - \beta_i$ is odd, except for the last one where we have $\beta_r = b$, since otherwise c_i cannot be of the form $u_i^2 + 4v_i^2$. So the optimal situation is obtained for a sequence that grows by steps of two, for which (3.4) becomes $\approx \frac{9b}{4}$. Even though this assumption is indeed too optimistic, the authors show how it is possible for an attacker to create more leeway for it to occur in [CD23, §11.3].

In [CD23, §8], the authors explain how to determine whether a subgroup $\langle (P_c, P), (Q_c, Q) \rangle$ of $C \times E$, obtained like kernel (3.3), is the kernel of a higher-dimensional $(2^a, 2^a)$-isogeny whose codomain is a product of elliptic curves. Additionally, they also mention:

- How to compute the "gluing" of the product $C \times E$ into the Jacobian of a genus-2 curve in [CD23, Proposition 1]. Note that the first gluing step does not fail since otherwise this would mean that $C \cong E$, from Kani's theorem [Kan97, Theorem 2.6]. If this unlucky and unlikely case occurs, then we could just compute another auxiliary isogeny $\gamma : E_0 \to C$ to another codomain curve and solve the issue.

- How to compute $(2,2)$-isogenies between Jacobians of genus-2 curves (called Richelot isogenies);

- How to verify whether the last step of the $(2,2)$-isogeny chain has as codomain a product of elliptic curves. This last computation is fairly simple, as it amounts to calculating the determinant δ of a 3×3 matrix and checking whether $\delta \overset{?}{=} 0$ or not.

Therefore, all the tools for computing the $(2^a, 2^a)$-isogeny $\Phi : C \times E \to A'$, that factors as a chain of $(2, 2)$-isogenies, are provided. Moreover, we also have a way to determine whether our guess of κ_i was correct by assessing if $A' \overset{?}{=} E_1' \times E_2'$. Overall, the situation one expects to have with a successful guess of κ_i is depicted in Figure 3.5.

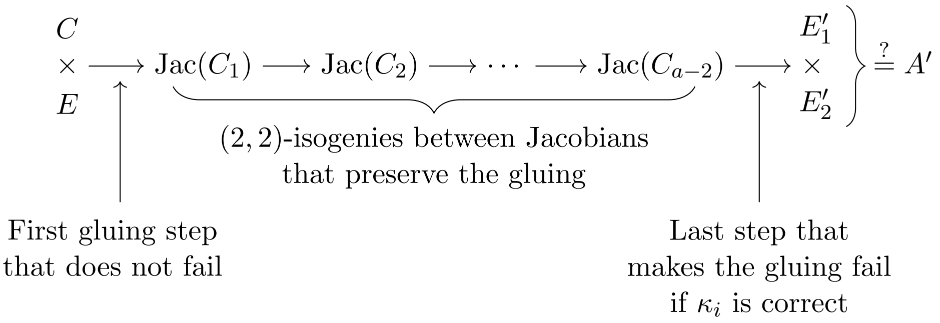
$C \times E \longrightarrow \mathrm{Jac}(C_1) \longrightarrow \mathrm{Jac}(C_2) \longrightarrow \cdots \longrightarrow \mathrm{Jac}(C_{a-2}) \longrightarrow E_1' \times E_2' \overset{?}{=} A'$
(2, 2)-isogenies between Jacobians
that preserve the gluing
First gluing step
that does not fail
Last step that
makes the gluing fail
if κ_i is correct

Figure 3.5: Diagram representing the situation one expects to have with a successful guess of κ_i though the "glue-and-split" attack. The arrows between abelian varieties represent the $(2, 2)$-isogenies that compose the $(2^a, 2^a)$-isogeny $\Phi : C \times E \to A' = E_1' \times E_2'$.

Since discussing the details of the three main speed-up methods adopted in [CD23, §7] is of limited interest for the survey, let us present them on a very high-level. Firstly it is mentioned to take the α_i's as large as possible, since the larger each α_i is, the smaller will be the length $a - \alpha_i$ of the chain of $(2, 2)$-isogenies. Secondly, the authors have pre-computed a large table to retrieve the values of u_i and v_i without the need for performing expensive factoring. On a side-note, this method also ensures that α_i is chosen as large as possible. Lastly, the authors propose to extend Bob's secret isogeny ϕ with an additional 3-isogeny ϕ' where useful and treat $\phi' \circ \phi$ as the new secret isogeny. This might be useful if, for instance, some candidate β_i does not admit an integer $\alpha_i \geq 0$ such that $2^{a-\alpha_i} - 3^{b-\beta_i}$ is of the desired form, but $\beta_i - 1$ does. We therefore prolong the secret isogeny, considering $P' = \phi'(P), Q' = \phi'(Q)$ and have the relevant expression be $2^{a-\alpha_i} - 3^{b+1-\beta_i}$ now. At this point, the attack can be extended to recover Bob's secret key.

Remark 12. *If 2^a is considerably smaller than 3^b, then one might want to attack Alice's secret key instead of Bob's. Recall that the reason why we prefer to attack Bob's secret isogeny is that in this way, when applying Kani's criterion, we compute a higher-dimensional $(2^a, 2^a)$-isogeny, that factors into a chain of $(2, 2)$-isogenies. Such a chain of isogenies can be computed efficiently through Richelot's formulae [Smi+05]. However, when attacking Alice's secret key, we must compute a higher-dimensional $(3^b, 3^b)$-isogeny, that factors into a chain of $(3, 3)$-isogenies, every time we apply Kani's criterion.*

Therefore, this remark serves the purpose of pointing out that it might be more efficient to recover Alice's secret 2^a-isogeny since there will be less guessing steps to perform compared to recovering Bob's secret 3^b-isogeny, even though computing $(3^b, 3^b)$-isogenies is way less efficient than computing $(2^a, 2^a)$-isogenies. Of course, if 2^a is much smaller than 3^b, then one can apply one of the first versions of the torsion-point attacks that required unbalanced parameters [Pet17; Que+21] and the attacks mentioned in the present chapter lose their value.

For what concerns the discussion on the runtime of the algorithm for the setting of known endomorphism ring [CD23, §10], let us mention that the simple case of considering $c_1 = 2^{a-\alpha_1} - 3^{b-\beta_1}$ of the form $u_1^2 + 4v_1^2$ does not allow to achieve polynomial runtime since, in view of Landau's theorem [SS66], we expect to test $O(\sqrt{a})$ pairs (α_1, β_1) before finding c_1 of the desired form. Therefore, to achieve polynomial runtime, we look for values c_1 of the form $c_1 = u_1^2 + nv_1^2$, for a prime number $n \leq a$. The authors argue that this occurs with overwhelming probability, reducing significantly the runtime of the attack, compared to the simpler description, and making it polynomial.

Remark 13. *Note that the first iteration dominates the overall runtime. Indeed, after finding suitable α_1, β_1, the expression $2^{a-\alpha_1} - 3^{b-\beta_1}$ can be re-used in the remaining iterations by extending Bob's secret isogeny as previously mentioned.*

Regarding the case of unknown endomorphism ring of the base curve E_0, the attack can be applied analogously (as described in [CD23, §11.3]), but it is required to find $c = 2^{\alpha-a} - 3^{b-\beta}$ that is smooth, since we do not know any path to E_0 and cannot exploit the endomorphism $2\mathbf{i}$. Unfortunately, the likelihood of finding a smooth c of the required form is very small, but the authors propose two different methods for creating more leeway for the attacker, at the cost of increasing the attack's runtime to subexponential. Indeed, we can increase the likelihood of finding the pair (α, β) required to apply the attack by either extending Bob's secret isogeny or by the aid of a sub-routine algorithm.

The combination of these two tweaks leads us to searching values c of the form $c = d2^{a-\alpha} - e3^{b-\beta}$, allowing for more valid pairs (α, β) hence increasing the likelihood of succeeding in the attack, since e and d are under the attacker's control. The subexponential attack is expected to run in $L_p(1/2, 0)$[15].

Remark 14. *The attack also applies to instances of SIDH where other torsion choices for Alice and Bob are made, i.e.* $\ell_A \neq 2$ *and* $\ell_B \neq 3$ *[CD23, §11.1]. Indeed, the secret* ℓ_B^b*-isogeny can still be recovered from information on the* ℓ_A^a*-torsion points for any small primes* ℓ_A, ℓ_B*, as long as* $\ell_B^b \approx \ell_A^a$*. Furthermore, the runtime remains polynomial, but if* $\ell_A \neq 2$ *the implementation of the attack requires more effort since one can no longer rely on fast Richelot isogenies [Smi+05].*

3.1.4 Improvements of the attack

Subsequently, several improvements and extensions of the attack were published and, in the attempt of providing a rapid overview, one must mention the works by Maino-Martindale [MM22][16], Oudompheng-Pope [OP22], and Wesolowski [Wes22] who observed how Kani's criterion allowed for a direct key recovery, without needing to perform any iterative steps thus improving on the runtime of the attack. Moreover, Wesolowski showed how the auxiliary isogeny γ could be constructed in a more efficient way without requiring the factorization of $c = 2^a - 3^b$ [Wes22].

More specifically, the work by Maino, Martindale, Panny, Pope and Wesolowski [Mai+23] is a merge of [MM22] and [Wes22] and has the main advantage of practicality, as the key can be recovered directly (i.e. the iterative decision strategy presented in [CD23] can be avoided entirely), and provability. Indeed, the methods developed for key recovery are so fast that they will be used in constructive applications, such as FESTA [BMP23]. This is due to the fact that the secret isogeny ϕ is recovered directly from a component of the matrix-representation of the higher-dimensional (d_A, d_A)-isogeny specifically crafted for the attack (where d_A is the degree of the torsion-points of which we have the images). This means that the computation of only one 2-dimensional isogeny is required, instead of one for every digit of the integer sk_A representing the secret isogeny ϕ_B as was done in [CD23]. Furthermore, for what concerns provability, the runtime of the attack is proved to be polynomial, in the case of known endomorphism ring of the starting curve, and subexponential otherwise, without having to rely on any heuristics as it was done in [CD23] (these will be briefly discussed towards the end of the chapter).

Let us describe the main idea behind the attack proposed in [Mai+23] that allows to solve Problem 3 in polynomial time if $\mathrm{End}(E_0)$ is known and subexponential time otherwise. Let all notation be as in Problem 3. Assume $2^a = d_A > d_B = 3^b$. Furthermore, assume we have complete access to some auxiliary isogeny $\gamma : C \to E_0$ of degree $c = d_A - d_B$, whose computation is context-dependent. The authors in [Mai+23, Algorithm 1] provide an algorithm to recover a generator of $\ker \phi_B$ (i.e. solve Problem 3) with complexity dominated by the cost of one evaluation of a (d_A, d_A)-isogeny with known kernel and two evaluations of $\hat{\gamma}$.

Therefore, the main idea is the following. Let $g_B : C \to F$ be the isogeny of kernel $\hat{\gamma}(\ker \phi_B)$, and $g_c : F \to E_B$ be the isogeny of kernel $g_B(\ker \gamma)$ so that the diagram in Figure 3.6 commutes.
Now consider the 2-dimensional isogeny

$$\begin{aligned} \Phi \colon E \times E_B &\longrightarrow E_0 \times F \\ (P, Q) &\longmapsto (\gamma(P) - \hat{\phi}_B(Q), g_B(P) + \hat{g}_c(Q)) \end{aligned}$$

[15] $\forall\ t \in [0, 1], \forall\ \gamma \in \mathbb{R}_{>0}$ we define $L_x(t, \gamma) := e^{(\gamma+o(1))(\log x)^t (\log\log x)^{1-t}}$ as $x \to +\infty$.

[16] This paper was later improved and updated to its most recent version [Mai+23].

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] & E_B \\
C \arrow[u, "\gamma"] \arrow[r, "g_B"'] & F \arrow[u, "g_c"']
\end{tikzcd}$$

Figure 3.6: Diagram representing the setup for the attack presented in [Mai+23, Figure 1].

and observe that $-\hat{\phi}_B$ is equal to the composition

$$E_B \xhookrightarrow{0\times\mathrm{id}_{E_B}} C \times E_B \xrightarrow{\Phi} E_0 \times F \overset{\mathrm{pr}_1}{\twoheadrightarrow} E_0,$$

where the first map is the inclusion map with image $\{0\} \times E_B$, the middle map is Φ, and the last is the natural projection on the first component, as it is defined in (1.3). Since the authors show that each map in the latter composition is efficiently computable, then it is possible to evaluate $\hat{\phi}_B$ on any input. Therefore, it is possible to directly recover $\ker(\phi_B)$, hence solve Problem 3 efficiently. Proving that each step of the composition is indeed efficiently computable amounts to verifying that the kernel of the 2-dimensional isogeny Φ can be computed efficiently, and that this kernel permits an efficient evaluation of Φ. Indeed the computation of the first inclusion is trivially simple and the last projection step seems simple to compute, even though it hides a subtlety. Finally, note that the decomposition $E_0 \times F$ is only available if Φ behaves well with respect to the implicit product polarizations of the domain and codomain.

For further details on the theory of polarizations, let us refer the reader to [Mai+23, §2.1], where it is introduced the following notation that will be particularly useful for us. Given four elliptic curves E_1, E_2, E_1', E_2' and four isogenies $\phi_{ij} : E_i \to E_j'$ for $i, j \in \{1, 2\}$, the matrix

$$M = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix}$$

represents the 2-dimensional isogeny

$$\begin{aligned} \Phi \colon E_1 \times E_2 &\longrightarrow E_1' \times E_2' \\ (P_1, P_2) &\longmapsto (\phi_{11}(P_1) + \phi_{12}(P_2), \phi_{21}(P_1) + \phi_{22}(P_2)). \end{aligned}$$

We will refer to M as the matrix form of Φ.

The attack presented in [Mai+23, Algorithm 1] is a consequence of the following theorem, that is based on Kani's criterion [Kan97, Theorem 2.6], and allows to determine whether a 2-dimensional isogeny, having as domain a product of elliptic curves, also has as codomain a product of elliptic curves.

Theorem 5 (Consequence of Kani's criterion). *Let f, d_A, d_B be pairwise coprime integers such that $d_A = f + d_B$, and let E_0, E_B, C, F be elliptic curves connected by the following commutative diagram of isogenies:*

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] & E_B \\
C \arrow[u, "\gamma"] \arrow[ur, "\rho"] \arrow[r, "g_B"'] & F \arrow[u, "g_c"']
\end{tikzcd}$$

where $\deg(\gamma) = \deg(g_c) = c$ *and* $\deg(\phi_B) = \deg(g_B) = d_B$. *The isogeny Φ represented by the matrix*

$$\begin{pmatrix} \gamma & -\hat{\phi}_B \\ g_B & \hat{g}_c \end{pmatrix}$$

is a (d_A, d_A)-isogeny with respect to the natural product polarization on $C \times E_B$ and $E_0 \times F$, and has kernel $\ker \Phi = \{([d_B]P, \rho(P)) \mid P \in E[d_A]\}$.

Theorem 5 states that the (d_A, d_A)-isogeny $\Phi : C \times E_B \to E_0 \times F$ can be computed efficiently (if d_A is smooth and preferably a power of two so that Richelot's formulae can be used) and implies the correctness of the algorithm described for recovering the secret key [Mai+23, Algorithm 1]. The authors exploit the fact that the information of the secret isogeny is carried in the matrix form of the higher-dimensional isogeny $\Phi : C \times E_B \to E_0 \times F$, and we content ourselves with this high-level description since presenting more details is beyond the scope of the survey.

Theorem 5 can be applied to attack SIDH by learning information about one of the two secret isogenies. In [CD23], the authors were able to obtain an oracle to determine whether a guess of a step along the secret isogeny path was correct. On a high-level, at every step of the attack they analyzed whether the (d_A, d_A)-isogeny splits into the product of supersingular curves or not. In [Mai+23], the authors noticed that the entire secret isogeny can be recovered through Kani's criterion since, up to isomorphism, the dual of the secret isogeny can be recovered from the element $-\hat{\phi}_B$ of the matrix representation of the (d_A, d_A)-isogeny. In [Rob23a], the latter strategy is generalized to higher dimension case to obtain a provable polynomial-time attack in the case of unknown endomorphism ring.

3.2 Key recovery without knowledge of the endomorphism ring

The most remarkable follow-up work is due to Robert [Rob23a], who had the idea of embedding the secret isogeny ϕ into an even higher-dimensional isogeny, working with abelian eightfolds of the type $E_0^4 \times E_B^4$ rather than surfaces. This allowed him to remove the need for any type of assumptions on the endomorphism ring and closed the door for any type of secure higher-dimensional variants of SIDH. Indeed, by going to dimension 8, it was possible to break all possible parameter instantiations for SIDH in polynomial time. Even though providing more details is beyond the scope of the survey, the following theorem extends Theorem 5 to even higher dimensions and is take from [Rob23a, Theorem 1.1].

Theorem 6 (Robert's attack). *Let d_A and d_B be smooth coprime integers such that $d_A > d_B$. Let $\mathbb{F}_q$ be the smallest field such that ϕ_B, and the points of $E_0[d_A]$ and $E_0[d_B]$ are defined*[17]. *Suppose that we are provided with the following:*

- *A secret d_B-isogeny $\phi_B : E_0 \to E_B$;*
- *The images of (a basis of) the d_A-torsion points of E_0;*
- *A basis of $E_B[d_B]$;*
- *The factorizations of d_A and d_B;*
- *A decomposition of $c := d_A - d_B$ as a sum of four squares.*

Then there is an explicit d_A-endomorphism $\Phi : E_0^4 \times E_B^4$ in dimension $g = 8$ such that evaluating Φ at (P, P, P, P, Q, Q, Q, Q) for any $P \in E_0(\mathbb{F}_q)$, $Q \in E_B(\mathbb{F}_q)$ allows to recover $\phi_B(P)$ and $\hat{\phi}_B(Q)$.

Moreover, the kernel of Φ can be explicitly described by 8 explicit rational generators which can be computed in time $O(\log d_A)$. This reduces recovering ϕ_B to evaluating the isogeny Φ in dimension 8 given generators of its kernel. Using optimized computations of smooth isogenies, as it is described in [DJP14, §4.2.2], such an isogeny can be evaluated in time $\tilde{O}(\ell_A^8 \log d_A)$ where ℓ_A is the largest prime divisor of d_A.

To conclude, a basis for the kernel of ϕ_B can be found in at most 2 evaluations of Φ on the basis of $E_B[d_B]$, for a total cost of $\tilde{O}(\ell_A^8 \log d_A)$.

[17] No further assumption on E_0 and E_A is made: it is not required that those curves are supersingular, even though in the case of SIDH they are indeed supersingular and $\mathbb{F}_q = \mathbb{F}_{p^2}$, for some prime p.

On a very high-level, as it is explained in [FMP23], Robert's attack [Rob23a] considers the genus-8 abelian variety $E_0^4 \times E_B^4$ and an endomorphism over it, described by the matrix

$$\Phi = \begin{pmatrix} \alpha_0 & \hat{\Phi}_B \\ -\Phi_B & \hat{\alpha}' \end{pmatrix}$$

where Φ_B is the natural extension of ϕ_B to the higher dimensional curve E_0^4, $\hat{\Phi}_B$ is its dual, and α_0 and α' are the endomorphisms over E_0^4 and E_B^4 respectively, with action given by the (same) matrix

$$M = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ -c_1 & c_0 & -c_4 & c_3 \\ -c_3 & c_4 & -c_1 & c_2 \\ -c_4 & -c_3 & c_2 & c_1 \end{pmatrix}$$

with c_0, c_1, c_2, c_3 such that $c := c_0^2 + c_1^2 + c_2^2 + c_3^2 = d_A - d_B$. We then have

$$\Phi \circ \hat{\Phi} = d_A \cdot \mathbf{I}_8,$$

where $\hat{\Phi} = \begin{pmatrix} \hat{\alpha}_0 & -\hat{\Phi}_B \\ \Phi_B & \alpha' \end{pmatrix}$ is the dual of Φ and $\mathbf{I}_8$ is the 8×8 identity matrix. This means that Φ is an endomorphism of degree d_A. As in previous torsion point attacks, one can evaluate Φ on the d_A-torsion using torsion point information provided by the SIDH protocol. Therefore, it is possible to compute Φ and finally deduce ϕ_B.

Therefore, Theorem 6 is the main ingredient for Robert's strategy that allowed to break SIDH for any arbitrary starting curve E_0 of unknown endomorphism ring. Moreover, in [Rob23a, Remark 1.2] the author also mentions the following result, that we present as a corollary of the previous theorem.

Corollary 1 (Robert's dual isogeny trick)**.** *Let d_A and d_B be sufficiently smooth coprime integers. Let $\{P_A, Q_A\}$ be the basis of $E_0[d_A]$. Assume $\{\phi_B(P_A), \phi_B(Q_A)\}$ is a known basis of $E[d_A]$. If $d_A^2 \geq d_B$ (i.e. $d_A \geq \sqrt{d_B}$), then it is possible to recover efficiently the d_B-isogeny $\phi_B : E_0 \to E_B$ by considering the dual isogeny Φ in Theorem 6.*

Remark 15. *Let us present three interesting details of the attack:*

- *The assumed knowledge of the decomposition of $c := d_A - d_B$ as a sum of four squares is not restrictive at all. Indeed, such decomposition can be pre-computed since it only depends on d_A and d_B. It can be done in random polynomial time $O(\log^2 c)$ binary operations [RS86]*
- *The attack is "quasi-linear", since it has a runtime of $\tilde{O}(\log d_A)$ arithmetic operations over $\mathbb{F}_q$, when $\ell_A \in \{O(1), O(\log\log d_A)\}$.*
- *The following* "embedding lemma" *is used. For any d-isogeny $\phi : E \to E'$ between abelian varieties of dimension g, and any $d' > d$, it is possible to efficiently embed ϕ into a d'-isogeny Φ in dimension $8g$ (or $4g$ or $2g$ in certain cases). This provides increased flexibility at the cost of going up in dimensions, and was used in [Rob23b] to show that an isogeny always admits an efficient representation.*

Robert's attack [Rob23a] generalizes the attacks presented in [CD23; Mai+23] to dimension g, where $g \in \{2, 4, 8\}$. Indeed, it is shown in [Rob23a] that the dimension-2 attacks, and the dimension-4 and dimension-8 attacks all fit together within the same framework, being all special instances of a more general attack. To conclude, let us present a brief summary of the situation of the attacks to SIDH when attempting to recover the secret isogeny $\phi_B : E_0 \to E_B$:

Known endomorphism ring:

- For E_0 being one of the starting curves submitted to NIST's competition (3.1), the attack proposed in [CD23] runs in heuristic polynomial time being $\tilde{O}(\log^{1.5} d_A \ell_A^2)$ arithmetic operations. Note that the runtime is "heuristic" for two reasons:

1. If a wrong path is guessed with κ_i, the codomain of Φ will not be a product of two supersingular elliptic curves, but rather a Jacobian of a superspecial curve. The estimated probability that this happens is $\approx 1/p$, but it still relies on the heuristic that the codomain of Φ for a wrong guess is uniform among all superspecial surfaces. Such heuristic was removed by works presented in [MM22; OP22; Wes22].
2. In order for the attack to work, it is required that $c = d_A - d_B = 2^a - 3^b$ is of the form $c = u^2 + 4v^2$. For a uniform integer smaller than x, the probability of it being decomposed in this way is $\approx 1/\sqrt{\log x}$. The heuristic here lies in the fact that tweaking the parameters to increase the likelihood of the attack succeeding does not increase their size too much. Moreover, the factorization of c is required in order to determine the pair (u, v). Such heuristic was removed in [Wes22].

- Using what is described in [Wes22; Mai+23], the dimension-2 attack can also apply to any elliptic curve with known endomorphism ring in *proven* polynomial time. Indeed, after a polynomial time precomputation to construct the c-isogeny γ and its action on a basis of $E_0[d_A]$, the attack follows the same process as the one described for Theorem 6, except for the fact that Φ is computed in dimension 2 and its evaluation costs $\tilde{O}(\log d_A \ell_A^2)$ arithmetic operations.
- Overall, if $\ell_A \in \{O(1), O(\log\log d_A)\}$, the dimension-2 attack has quasi-linear complexity of $\tilde{O}(\log d_A)$ arithmetic operations by Theorem 6.

Unknown endomorphism ring:

- When E_0 is a "random" curve, with unknown endomorphism ring, the dimension 2 attack presented in [Mai+23] (resp. [CD23]) has (resp. heuristic) subexponential runtime $L_p(1/2, 0)$.
- The dimension 4 attack presented in [Rob23a, §4] is in heuristic polynomial time because parameter tweaks are required. Indeed under the heuristics required for tweaking the parameters, a precomputation similar to the one involving the endomorphism $2\mathbf{i}$ in [CD23] costs $O(\log^3 d_A)$ binary operations to find the required decomposition of $d_A^2 = (b_1^2 + 2b_2)^2 d_B + (a_1^2 + a_2^2)$ [Rob23a, Heuristic 4.4]. After this step, the attack runs in $\tilde{O}(\log d_A \ell_A^4)$ arithmetic operations.
- The dimension 8 attack presented in [Rob23a, §2] is the best option to attack SIDH as it runs in proven polynomial time in $\tilde{O}(\log d_A \ell_A^8)$ arithmetic operations by Theorem 6. The precomputation step requires to compute the decomposition of $c = d_A - d_B$ as a sum of four squares and can be done in randomized $O(\log^2 d_A)$ binary operations.
- Overall, if $\ell_A \in \{O(1), O(\log\log d_A)\}$, the dimension-8 and dimension-4 attacks have quasi-linear complexity of $\tilde{O}(\log d_A)$ arithmetic operations by Theorem 6.

Therefore, we see that Robert's attack [Rob23a] allowed to break any instance of the SIDH protocol, even when the endomorphism ring of the starting curve is not known. Indeed, it is possible to recover the secret isogeny $\phi_B : E_0 \to E_B$ of degree d_B, given information about the d_A-torsion points, whenever $d_A \geq \sqrt{d_B}$. From now on, let us refer to the attacks described so far as the "SIDH attacks" [CD23; Mai+23; Rob23a].

Part II

Countermeasures & New Constructions

Chapter 4

M-SIDH & MD-SIDH: masking information

Masked torsion points SIDH (M-SIDH) and Masked-Degree SIDH (MD-SIDH) [FMP23] were among the first variants of the regular SIDH protocol that tried to limit the effectiveness of the series of devastating attacks published in August 2022 [CD23; Mai+23; Rob23a], and discussed in Chapter 3, with the goal of repairing the SIDH protocol.

Given that the focal point of the attacks was to exploit the knowledge of the torsion point information and the degree of the secret isogeny, publicly available in the SIDH protocol, the two constructions that will be analyzed in the present chapter partially hide this type of information. Indeed, the SIDH attacks were able to recover Bob's secret isogeny due to the knowledge of the following:

1. The degree d_B of Alice's secret isogeny $\phi_B : E_0 \to E_B$;
2. The images $\phi_B(P_A), \phi_B(Q_A)$ of a torsion basis such that $E_0[d_A] = \langle P_A, Q_A \rangle$, where d_A is an integer coprime to d_B such that $d_A > d_B$.

Therefore, the countermeasures that will be analyzed within the present chapter are able to mask each of these factors that are necessary in order to mount the attacks.

The main idea of the M-SIDH variant is to mask the torsion point information with a random integer, scaling the images of the torsion points, thus revealing less information than the regular SIDH protocol. The degrees of the secret isogenies remain fixed and publicly known as in SIDH.

The central concept underlying the MD-SIDH variant is to mask the degree of the secret isogeny. In other words, the degrees of the secret isogenies are no longer fixed, but rather uniformly sampled random divisors of d_A and d_B (the degrees of the secret isogenies in SIDH) respectively. However, the images of the torsion points must also be scaled by a random integer. In this way, we prevent the degree from being recovered by computing a Weil pairing and some discrete logarithm in a group of smooth order. Therefore, it is possible to interpret the MD-SIDH variant as a generalization of M-SIDH.

It is noteworthy that a randomly generated starting curve must be used in order to avoid attacks that leverage the knowledge of the endomorphism ring. The proposed method can be found in Section 4.4.1. Moreover, the M-SIDH variant seems to be the most promising of the two, due to it requiring smaller keys for achieving identical security levels, as discussed in Section 4.4.2. The public parameters are expected to be a factor of $O(\log \lambda)$ larger than in SIDH, having sizes of 4434, 7037 and 9750 bytes in order to achieve AES-128, AES-192 and AES-256 security levels (respectively NIST security levels 1, 3 and 5).

Lastly, before delving into the details of the two countermeasures, we state and prove the following lemma, that will be useful throughout the chapter.

Lemma 1 (Recover degree)**.** *Let $\phi : E \to E'$ be an isogeny of unknown degree d and let δ be a smooth integer coprime to d such that $E[\delta] \subset E(\mathbb{F}_{p^2})$. Set $E[\delta] = \langle P, Q \rangle$. Then, given P, Q, $\phi(P)$, and $\phi(Q)$, there exists a polynomial time algorithm to recover $d \mod B$.*

Proof. The idea is to first compute the Weil pairing values $e_\delta(P,Q)$ and $e_\delta(\phi(P),\phi(Q))$ and then solve a discrete logarithm instance between both quantities to recover $d \mod B$. Indeed, we have that

$$e_\delta(\phi(P),\phi(Q)) = e_\delta(P,\hat{\phi}\circ\phi(Q)) = e_\delta(P,[\deg\phi]Q) = e_\delta(P,E)^{\deg\phi}$$

by leveraging the property of Weil pairings for which $e_\delta(\phi(P),Q) = e_\delta(P,\hat{\phi}(Q))$ and also the property of supersingular isogenies for which $\hat{\phi}\circ\phi = [\deg\phi]$.

Moreover, since $E[\delta] \subset E(\mathbb{F}_{p^2})$, the pairing computation are efficient and run in polynomial time. Therefore, since δ is a smooth integer, Pohlig-Hellman's polynomial-time (running in $O(T\cdot\log\delta)$ for smooth δ, where T is the cost of a single group operation) algorithm can be used to solve the discrete logarithm instance. □

4.1 The M-SIDH variant

The goal of the *Masked torsion points* variant, presented in [FMP23, §3.1] is to make the images $\phi(P),\phi(Q)$ of P,Q under the secret isogeny ϕ not available to the adversary but, of course, still make the key exchange succeed. This means that, given a point $R \in E_0[B]$, it must be possible to compute a generator of the group $\phi(\langle R\rangle)$, as a commonly done within the SIDH framework.

In order for the scheme to function correctly, the images $\phi_A(P_B),\phi_A(Q_B)$ (resp. $\phi_B(P_A),\phi_B(Q_A)$) of P_B,Q_B (resp. P_A,Q_A) are scaled by a random integer $\alpha \in \mathbb{Z}/d_B\mathbb{Z}^\times$ (resp. $\beta \in \mathbb{Z}/d_A\mathbb{Z}^\times$), so Alice (resp. Bob) reveals $[\alpha]\phi_A(P_B),[\alpha]\phi_A(Q_B)$ (resp. $[\beta]\phi_B(P_A)$, $[\beta]\phi_B(Q_A)$) instead of $\phi_A(P_B),\phi_A(Q_B)$ (resp. $\phi_B(P_A)$, $\phi_B(Q_A)$). In order for the scheme to be secure against the SIDH attacks, the attacker must not be able to recover the scalar α (resp. β).

Focusing on Alice's execution of the protocol, let us now discuss further the value $\alpha \in \mathbb{Z}/d_B\mathbb{Z}^\times$ to be selected in order to effectively mask the images of the torsion points. Since the degree of the secret isogeny ϕ_A is fixed, one can recover $\alpha^2\cdot\deg\phi_A$ by applying Lemma 1. From this value, it is possible to derive $\alpha^2 \mod d_B$, since $\deg\phi_A$ is a known quantity. At this point it is possible to notice that α should be sampled directly from $\mu_2(d_B)$, where $\mu_2(N) := \{x \in \mathbb{Z}/N\mathbb{Z} \mid x^2 \equiv 1 \mod N\}$. This is due to the fact that even if Alice were to not pick $\alpha \in_\$ \mu_2(d_B)$, it would still be possible to compute a square root α_0 of α^2 and recover $[\alpha\alpha_0^{-1}\phi_A(P_B)]$ and $[\alpha\alpha_0^{-1}\phi_A(Q_B)]$, where we would have $(\alpha\alpha_0^{-1})^2 \equiv 1 \mod B$.

Therefore, the isogeny degrees d_A and d_B are chosen such that they have $t \geq 2\lambda$ distinct prime divisors and, thus, an exponential number of square roots of 1 modulo d_B (resp. d_A). In this way, the scalar cannot be recovered despite the fact that its square modulo d_A or d_B is known, preventing the efficient recovery of the secret scalar α (resp. β).

We hereby describe the M-SIDH variant presented in [FMP23, §3.1]:

Public parameters: Let λ be the security parameter and let $t = t(\lambda) \in \mathbb{N}$ be an integer depending on λ. Let $p = d_Ad_Bf - 1$ be a prime such that $d_A = \prod_{i=1}^t \ell_i$ and $d_B = \prod_{i=1}^t q_i$ are coprime integers, ℓ_i, q_i are distinct small primes, $d_A \approx d_B \approx \sqrt{p}$ and f is a small cofactor. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Set $E_0[d_A] = \langle P_A,Q_A\rangle$ and $E_0[d_B] = \langle P_B,Q_B\rangle$. The public parameters are $E_0,p,d_A,d_B,P_A,Q_A,P_B,Q_B$.

Public key (Alice): Alice samples uniformly at random two integers $\alpha \in_\$ \mu_2(d_B)$ and $a \in_\$ \mathbb{Z}/d_A\mathbb{Z}$. She computes the cyclic isogeny $\phi_A : E_0 \to E_A = E_0/\langle P_A + [a]Q_A\rangle$. Her public key is the tuple $\mathsf{pk}_A = (E_A, [\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$ and her secret key is $\mathsf{sk}_A = a$. The integer α is deleted.

Public key (Bob): Analogously, Bob samples uniformly at random two integers $\beta \in_\$ \mu_2(d_A)$ and $b \in_\$ \mathbb{Z}/d_B\mathbb{Z}$. He computes the cyclic isogeny $\phi_B : E_0 \to E_B = E_0/\langle P_B + [b]Q_B\rangle$. His public key is the tuple $\mathsf{pk}_B = (E_B, [\beta]\phi_B(P_A), [\beta]\phi_B(Q_A))$ and his secret key is $\mathsf{sk}_B = b$. The integer β is deleted.

Shared key (Alice): Upon receiving Bob's public key (E_B, R_a, S_a), Alice checks that $e_{d_A}(R_a,S_a) = e_{d_A}(P_A,Q_A)^{d_B}$, if not she aborts. She computes the isogeny $\phi'_A : E_B \to E_{BA} = E_B/\langle R_a + [a]S_a\rangle$. The shared key is the value $j(E_{BA})$.

Shared key (Bob): Analogously, upon receiving Alice's public key (E_A, R_b, S_b), Bob checks that $e_{d_B}(R_b, S_b) = e_{d_B}(P_B, Q_B)^{d_A}$, if not he aborts. He computes the isogeny $\phi'_B : E_A \to E_{AB} = E_A/\langle R_b + [b]S_b\rangle$. The shared key is the value $j(E_{AB}) = j(E_{BA})$.

Remark 16. *Let us clarify why Alice should check that $e_{d_A}(R_a, S_a) = e_{d_A}(P_A, Q_A)^{d_B}$. In case of correct execution of the protocol, we would have $R_a = [\beta]\phi_B(P_A)$, $S_a = [\beta]\phi_B(Q_A)$ and therefore*

$$\begin{aligned} e_{d_A}(R_a, S_a) &= e_{d_A}([\beta]\phi_B(P_A), [\beta]\phi_B(Q_A)) = e_{d_A}(P_A, \widehat{[\beta]\phi_B} \circ [\beta]\phi_B(Q_A)) \\ &= e_{d_A}(P_A, \widehat{[\beta]\phi_B} \circ [\beta]\phi_B(Q_A)) = e_{d_A}(P_A, [\beta^2]\hat{\phi}_B \circ \phi_B(Q_A) \\ &= e_{d_A}(P_A, [\beta^2][\deg \phi_B]Q_A) = e_{d_A}(P_A, Q_A)^{d_B}, \end{aligned}$$

since $e_{d_A}(\phi(P), Q) = e_{d_A}(P, \hat{\phi}(Q))$, $\hat{\phi} \circ \phi = [\deg \phi]$ and $\beta^2 = 1 \mod d_A$.

4.1.1 Security analysis of the M-SIDH variant

The problem assumed to be hard in order for the M-SIDH variant to be secure is the following.

Problem 4 (M-SIDH Problem). *Let $d_A = \ell_1 \dots \ell_t$ and let $d_B = q_1 \dots q_t$ be two smooth coprime integers, let f be a small cofactor such that $p = d_A d_B f - 1$ is a prime with $d_A \approx d_B$. Let $E_0/\mathbb{F}_{p^2}$ be a supersingular elliptic curve such that $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (d_A d_B f)^2$, set $E_0[d_B] = \langle P, Q\rangle$. Let $\phi : E_0 \to E$ be a uniformly random d_A-isogeny and let α be a uniformly random element of $\mu_2(d_B)$. Given $E_0, P, Q, E_A, P' = [\alpha]\phi(P), Q' = [\alpha]\phi(Q)$, compute the d_A-isogeny ϕ.*

It is worth mentioning that the main difference between Problem 3 and Problem 4, apart from wanting to recover Bob's isogeny in the former and Alice's in the latter, is that in the latter the torsion point images are provided scaled by a square root of unity. Therefore, when trying to apply the attack described in [Rob23a], the higher-dimensional endomorphism built specifically for the attack cannot be avaluated exactly and its kernel cannot be computed explicitly. Since Robert's attack [Rob23a] generalizes the attacks described in [CD23] and in [Mai+23], it is safe to say that the same holds for these attacks as well.

In what follows, with reference to [FMP23, §4], we analyze the security of this new variant to the SIDH protocol. Let us first discuss a general attack, that simply consists of guessing exact torsion point information until we have enough to run the SIDH attacks. The authors state that such an attack will have exponential complexity in the number of prime divisors of d_A and d_B and that it works for any starting curve, even for curves with unknown endomorphism ring. Secondly, let us present a polynomial time attack when the initial curve has j-invariant $j = 1728$. This attack is generalized to starting curves that have at least one (known) small degree endomorphism. The authors conclude that, from this preliminary analysis, an extension of the attacks to the case where the endomorphism ring of the starting curve is unknown seems to be hard. The target isogeny $\phi_A : E_0 \to E_A$ will be a d_A-isogeny, with images of the torsion points of order d_B revealed up to a scalar α.

Guessing enough exact torsion point information

We have $d_A \approx d_B$, where d_A is the degree of Alice's secret isogeny. As stated in [FMP23, §4.1], in order to apply the SIDH attacks, we only need the exact images of the $\sqrt{d_A} \approx \sqrt{d_B}$-torsion points, by leveraging Corollary 1. Indeed, it suffices that the square of the torsion-degree of the points of which we know the exact images is greater than the degree of the target secret isogeny.

Let $n \geq 1$ be the largest index such that $\sqrt{d_A} \leq \ell_n \dots \ell_t$, where the ℓ_i's are distinct small prime factors of d_B. Let $N = \ell_n \dots \ell_t$. Alice's secret isogeny ϕ_A can be recovered from its action on the N-torsion points. Indeed, since $N \mid d_B$, we have that any N-torsion point is also a d_B-torsion point. So, from the action of $[\alpha] \circ \phi_A$ on the d_B-torsion points, one can deduce the action of $[\alpha] \circ \phi_A$ on the N-torsion points. This is because a generic N-torsion point P_1 would be such that

$$P_1 = \left[\frac{d_B}{N}\right] P' = \left[\frac{d_B}{N}\right] [\alpha]\phi_A(P_B) = [\alpha]\phi_A\left(\left[\frac{d_B}{N}\right] P_B\right)$$

and $\left[\frac{d_B}{N}\right] P_B$ is an N-torsion point, since P_B is a d_B-torsion point (just like $P' = [\alpha]\phi_A(P_B)$).

Therefore, the only thing preventing an application of the SIDH attacks to recover Bob's secret isogeny is the unknown square root of unity α. Since N has $t-n+1$ prime factors, then there are at most 2^{t-n+1} square roots of unity modulo N. A possible strategy would be to try all these square roots of unity until one gets one for which the SIDH attack is successful. Therefore, the overall complexity of this attack is $\tilde{O}(2^{t-n+1})$[18] using a classical computer. Since, $N \approx \sqrt{d_B}$ and N is the product of the largest prime factors of d_B, then we have that $n > t/2$ and thus, for $t \sim 128$, $t-n+1 \leq t/2$.

Moreover, when using a quantum computer, the complexity of the attack can be improved due to Grover's algorithm [Gro96], allowing to run the same process in time $\tilde{O}(2^{(t-n+1)/2})$, where $t-n+1 \leq t/2$.

Remark 17. *From the run-times obtained, we have that the M-SIDH variant can be broken in $\tilde{O}(2^{t/2})$ steps with a classical computer and in $\tilde{O}(2^{t/4})$ steps with a quantum computer. Therefore, in order to achieve AES-λ security (i.e. λ bits of classical security and $\lambda/2$ bits of quantum security), the value t should be chosen such that $t \geq 2\lambda$.*

Polynomial time attack when E_0 has $j = 1728$

The authors in [FMP23, §4.2] notice that older attacks that leveraged the knowledge of torsion points, such as [Bas+21; FP22], only required the images of the torsion points up to a multiplicative constant, assuming that $d_A \ll d_B$. In what follows we see how a variant of these older attacks can be applied to M-SIDH.

Let $\iota : (x, y) \mapsto (-x, iy)$ be a non-trivial automorphism over E_0 and let $\psi := \phi_A \circ \iota \circ \hat{\phi}_A$ be the "lollipop endomorphism" first presented in [Pet17] and breifly introduced in 3. Additionally, given the framework described in M-SIDH, we have

$$[\alpha]\phi_A \circ \iota \circ \widehat{[\alpha]\phi_A} = [\alpha^2] \circ \phi_A \circ \iota \circ \hat{\phi}_A = [\alpha^2] \circ \psi.$$

Since $\alpha^2 \equiv 1 \mod d_B$, we have that $\psi \equiv [\alpha^2] \circ \psi$ over the d_B-torsion points. Therefore, from the action of $[\alpha]\phi_A$ on the d_B-torsion points, it is possible to obtain the images of the d_B-torsion points through $[\alpha^2] \circ \psi$, being these actually the exact images of d_B-torsion points through the action of ψ, since $\alpha^2 \equiv 1 \mod d_B$. Moreover, we have that $\deg \psi = \deg(\phi_A)^2 \cdot \deg \iota = d_A^2$, since $\deg \iota = 1$, and therefore we can apply Corollary 1 to recover ψ. After recovering the isogeny ψ, it is possible to recover ϕ efficiently, as presented in [Pet17].

Generalization attempts

In the attempt of generalizing the two previous attacks, we might wonder what can be done in the case of dealing with two more general cases, such as:

1. Having curves with j-invariant $j \neq 1728$;
2. Having curves with unknown endomorphism ring.

Eventually, from [FMP23, §4.3, §4.4] it seems like there is no strategy that can help recover the secret isogeny ϕ_A in a more efficient way than a simple guessing strategy.

4.1.2 Generalized lollipop attack for the M-SIDH variant

The main idea underlying the attack presented in [CV23] generalizes the "lollipop attack" [Pet17] by constructing a new isogeny ψ from E_A to a new curve E' that ignores the unknown α. From now on, for ease of notation, let us denote with E the codomain of Alice's secret d-isogeny ϕ.

[18] The notation $\tilde{O}$ is a variant of the big-O notation that "ignores" logarithmic factors [Cor+22]. That is:

$$f(n) \in \tilde{O}(g(n)) \iff \exists\, k : f(n) \in O(g(n) \cdot \log^k(g(n)))$$

Intuition of the attack

Let us first present a simpler version of the attack, described in [CV23, §1] and illustrated in Figure 4.1. Considering E_0 to be a $\mathbb{F}_p$-rational curve, let $E^{(p)}$ denote the Frobenius conjugate of E (raising all coefficients of E to the p-th power) and $\pi : E \to E^{(p)}$ the connecting Frobenius isogeny. The curve $E^{(p)}$ will play the role of the curve E' unobservant of the unknown value α. Moreover, the isogeny $\phi^{(p)} : E_0 \to E^{(p)}$ is the Frobenius conjugate of $\phi : E_0 \to E$ and is such that

$$\phi^{(p)} \circ \pi_0 = \pi \circ \phi,$$

where π_0 is the Frobenius endomorphism over E_0. Let us consider the isogeny $\psi = \phi^{(p)} \circ \hat{\phi}$ having degree d^2.

$$\begin{tikzcd} & E_0 \arrow[dl, "\phi^{(p)}"'] \arrow[dr, "\phi"] \arrow[loop below, "\pi_0"] & \\ E^{(p)} & & E \arrow[ll, "\pi"] \end{tikzcd}$$

Figure 4.1: Diagram illustrating a simple example of the attack

Let $T = [\alpha]\phi(P)$ and $S = [\alpha]\phi(Q)$ be the images revealed by the M-SIDH variant, then it can be shown that the following holds:

$$\psi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot M_{\pi_0}^{-1} \cdot \pi \begin{pmatrix} S \\ T \end{pmatrix}, \tag{4.1}$$

where M_{π_0} is the matrix representing the map π_0, that is:

$$\pi_0 \begin{pmatrix} P \\ Q \end{pmatrix} = M_{\pi_0} \begin{pmatrix} P \\ Q \end{pmatrix}.$$

Let us see why (4.1) holds:

$$\begin{aligned} \psi \begin{pmatrix} S \\ T \end{pmatrix} &= \phi^{(p)} \circ \hat{\phi} \begin{pmatrix} S \\ T \end{pmatrix} = \phi^{(p)} \circ \pi \circ \pi_0^{-1} \circ \hat{\phi} \begin{pmatrix} S \\ T \end{pmatrix} = \pi \circ \phi \circ \pi_0^{-1} \circ \hat{\phi} \begin{pmatrix} S \\ T \end{pmatrix} \\ &= A \cdot \pi \circ \phi \circ \pi_0^{-1} \circ \hat{\phi} \circ \phi \begin{pmatrix} P \\ Q \end{pmatrix} = d \cdot A \cdot \pi \circ \phi \circ \pi_0^{-1} \begin{pmatrix} P \\ Q \end{pmatrix} = d \cdot A \cdot \pi \circ \phi \circ M_{\pi_0}^{-1} \begin{pmatrix} P \\ Q \end{pmatrix} \\ &= d \cdot A \cdot M_{\pi_0}^{-1} \cdot \pi \circ \phi \begin{pmatrix} P \\ Q \end{pmatrix} = d \cdot M_{\pi_0}^{-1} \cdot \pi \begin{pmatrix} S \\ T \end{pmatrix}. \end{aligned}$$

Indeed, we have that $\phi^{(p)} \circ \pi = \pi \circ \phi$, $\begin{pmatrix} S \\ T \end{pmatrix} = A \cdot \phi \begin{pmatrix} P \\ Q \end{pmatrix}$ and $\hat{\phi} \circ \phi = [d]$.

Now, since in equation (4.1) we have only known quantities, it is possible to compute the exact images of S, T under the isogeny ψ and hence apply the polynomial time SIDH attacks to recover ψ. This is due to the fact that in M-SIDH we have $d_B > d$[19] and therefore $d_B^2 > \deg(\psi) = d^2$. Regardless of whether ψ is cyclic or not, we will see in Remark 18 that is possible to derive almost all information about ϕ.

Generalization

The previously described attack can be generalized to the cases for which E_0 is not necessarily $\mathbb{F}_p$-rational by using a different map than Frobenius to connect E to E'.

Let us describe the overall framework presented in [CV23, §3.1]. Recall that we want to retrieve Alice's secret isogeny $\phi : E_0 \to E$ of known degree d, given the bases $\{P, Q\} \subseteq E_0[d_B]$ and $\{S, T\} \subseteq$

[19] Generally speaking, we always have that one between d_A, that is d in this scenario, or d_B is bigger than the other. Indeed, we require for $d_A \approx d_B$, and not exactly equal.

$E[d_B]$ such that $S = [\alpha]\phi(P)$ and $T = [\alpha]\phi(Q)$. Let us express these last relations through the following notation:

$$\begin{pmatrix} S \\ T \end{pmatrix} = A \cdot \phi \begin{pmatrix} P \\ Q \end{pmatrix}$$

where A is a matrix sampled uniformly at random from the public set $X \subseteq \mathrm{GL}_2(\mathbb{Z}/d_B\mathbb{Z})$. In this case, X is the set of all scalar matrices and A has elements equal to $\alpha \in \mu_2(d_B)$. Moreover, we take into account also the two following additional auxiliary isogenies:

1. $\sigma_0 : E_0 \to E_0'$ such that $\deg \sigma_0 = s$. We assume that its push-forward σ under ϕ is known, where

$$\sigma := \phi_* \sigma_0 : E \to E';$$

   In other words, we have that $\ker(\sigma) = \phi(\ker(\sigma_0))$ is known.

2. $\omega : E_0 \to E_0'$, such that $\deg \omega = w$.

We must assume that d_B, d, s, w are all pairwise coprime and that $p \nmid dw$. The overall configuration is depicted in Figure 4.2. We can see that, indeed, the original "lollipop attack" presented in [Pet17] is generalized to arbitrary sets X and to an arbitrary choice of σ_0, that needs not to be the Frobenius isogeny, providing the advantage of possibly reducing the degree of the isogeny ϕ and therefore also improving on the bound required for applying the SIDH attacks.

$$\begin{tikzcd}
E_0' \arrow[d, "\phi' = \sigma_{0*}\phi"'] & E_0 \arrow[l, bend right, "\sigma_0"'] \arrow[l, bend left, "\omega"] \arrow[d, "\phi"] \\
E' & E \arrow[l, "\sigma = \phi_* \sigma_0"]
\end{tikzcd}$$

Figure 4.2: Diagram illustrating the generalized attack

The attack will provide a way to evaluate the wd^2-isogeny

$$\psi := \phi' \circ \omega \circ \hat{\phi} : E \to E'$$

for any given point. In this case $\phi' = \sigma_{0*}\phi$ is such that $\phi' \circ \sigma_0 = \sigma \circ \phi$ by construction. Just like in the simplified version previously described, some non-trivial information about ϕ can be recovered regardless of whether ψ is cyclic or not.

Remark 18. *Let us see why it is always possible to recover almost all information about ϕ, regardless of ψ being cyclic or not. If ψ is cyclic we can simply recover* $\ker(\hat{\phi})$ *as* $\ker(\psi)[d]$*, since $\hat{\phi}$ has degree d. If ψ is not cyclic, we have to assume that $\hat{\sigma} \circ \omega$ is cyclic, otherwise the attack will not be successful as we would be clueless about ϕ. As stated in [CV23, §3.2], we can recover a component of $\hat{\phi}$ of degree d/d' that is outgoing from E. This leads to having a remaining gap of degree d' to be filled in order to recover ϕ entirely, as depicted in Figure 4.3. Thus, we now want to find* $\ker(\phi)[d']$ *in order to retrieve* $\ker(\phi)$ *and, thus, the isogeny ϕ. Note that the case where ψ is cyclic translates to having $d' = 1$. At this point we can proceed by guessing the remaining gap of degree d' since the number of options is roughly of the order of $O(2^{r'})$, where r' denotes the number of distinct prime factors of d' and is expected to be very small by the authors.*

A crucial element of the attack is the following lemma [CV23, Lemma 3]:

Lemma 2 (Evaluate ψ)**.** *With the notation introduced above, assume that M is a matrix such that*

$$(\hat{\sigma}_0 \circ \omega) \begin{pmatrix} P \\ Q \end{pmatrix} = M \cdot \begin{pmatrix} P \\ Q \end{pmatrix}.$$

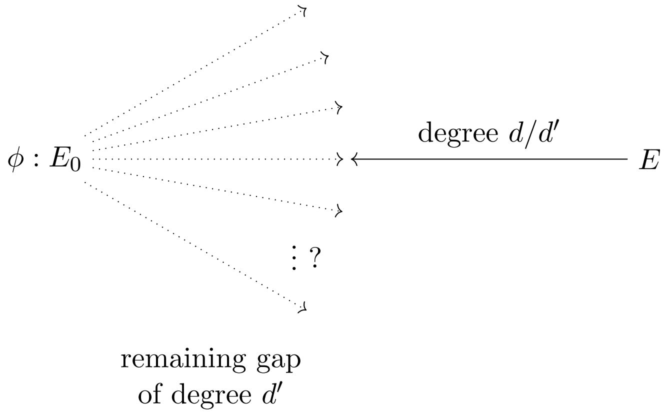

Figure 4.3: Intuition of extracting ϕ from ψ.

Moreover, if M commutes with any other matrix in the set X[20]*, then we have*

$$s \cdot \psi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot M \cdot \sigma \begin{pmatrix} S \\ T \end{pmatrix}.$$

Proof. Since $\begin{pmatrix} S \\ T \end{pmatrix} = A \cdot \phi \begin{pmatrix} P \\ Q \end{pmatrix}$ and it is known that $\hat{\phi} \circ \phi = [d]$, we have that

$$\hat{\phi} \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot A \cdot \begin{pmatrix} P \\ Q \end{pmatrix}.$$

Moreover, since $\phi' \circ \sigma_0 = \sigma \circ \phi$, we have that $[s]\phi' = \sigma \circ \phi \circ \sigma_0$ and therefore

$$s \cdot (\phi' \circ \omega \circ \hat{\phi}) \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot A \cdot (\sigma \circ \phi \circ \hat{\sigma}_0 \circ \omega) \begin{pmatrix} P \\ Q \end{pmatrix} = d \cdot A \cdot M \cdot (\sigma \circ \phi) \begin{pmatrix} P \\ Q \end{pmatrix}$$

Hence we have that

$$s \cdot \phi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot A \cdot M \cdot A^{-1} \cdot \sigma \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot M \cdot \sigma \begin{pmatrix} S \\ T \end{pmatrix}$$

since $\begin{pmatrix} S \\ T \end{pmatrix} = A \cdot \phi \begin{pmatrix} P \\ Q \end{pmatrix}$ and M commutes with the matrix $A \in X$. □

Lemma 2 is particularly important as it provides the knowledge of $\psi(S)$ and $\psi(T)$ whenever it applies, since we assume s and d_B to be pairwise coprime. Moreover, under the assumptions that d_B is smooth and that $d_B^2 > \deg \psi = wd^2$, we can apply the SIDH attacks and compute the kernel of ψ in a polynomial number of operation.

Even if assuming the knowledge of the push-forward of σ_0 by the unknown isogeny ϕ may seem quite restrictive, there are two natural candidates for the map σ_0, being:

1. The identity map $\mathrm{id} : E_0 \to E_0$, having push-forward the identity map over E;

2. The Frobenius isogeny $\pi_0 : E_0 \to E_0^{(p)}$, whose push-forward is the Frobenius isogeny $\pi : E \to E^{(p)}$.

Moreover, it is possible to obtain other examples by composing one of the two options above with an isogeny of small degree. In this scenario, the push-forward of σ_0 by the unknown isogeny ϕ can be guessed with a reasonable success probability, as seen in Remark 18.

[20] This is always true for the M-SIDH variant since scalar matrices commute with any other matrix.

Application to the M-SIDH variant

Let us see how the attack described can be applied to the M-SIDH variant, analyzing different possibilities for σ_0 (and then selecting ω accordingly). Recall that we have $S = [\alpha]\phi(P)$ and $T = [\alpha]\phi(Q)$ for a torsion basis $\{P, Q\}$ of $E[d_B]$, where $\alpha \in \mu_2(d_B)$ and $d = \deg(\phi)$. We will discuss two main cases: $\sigma_0 = \mathrm{id}$ and $\sigma_0 = \pi_0$, as presented in [CV23, §4].

Case $\boldsymbol{\sigma_0 = \mathrm{id}}$: It is possible to obtain an oracle for evaluating the isogeny $\psi = \phi \circ \omega \circ \hat{\phi}$ if $d_B > d\sqrt{w}$, with ω being an endomorphism over E_0 of degree w. Indeed, in this setting it is possible to apply Lemma 2, obtaining

$$\psi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot M \begin{pmatrix} S \\ T \end{pmatrix},$$

where M is the matrix representing the endomorphism ω. Thus, if ω happens to be an endomorphism of sufficiently small degree w, then it is likely that $d_B > d\sqrt{w}$[21]. Therefore, unless ω is a scalar endomorphism such that $\omega \equiv [\alpha] \mod d_B$ in $\mathrm{End}(E_0)$[22], it is possible to recover non-trivial information regarding ϕ. Indeed, as one expects ψ to be a cyclic isogeny, retrieving $\ker(\phi)$ is particularly explicit, as seen in Remark 18. In conclusion, as long as E_0 comes equipped with a small-degree non-scalar endomorphism ω, the M-SIDH variant should be considered broken, just like it was already discussed in Section 4.1.1 when performing a polynomial time attack in the case for which $j(E_0) = 1728$.

Remark 19. *In the case of overstretched parameters such that $d_B/d > p^{1/3}$, if $\mathrm{End}(E_0)$ is known then the attack can be run with a scalar endomorphism ω over E_0 of degree $p^{2/3}$. Such an endomorphism exists due to [LB20, Prop. B.5] and can be computed through the LLL reduction algorithm [LLL82]. Note that a similar statement was made also by the authors in [FMP23, §4.3], who thought that ω could be an isogeny of degree $\approx p^{1/2}$. However, this turns out to be an overoptimistic estimate according to [CV23, Footnote 5].*

Case $\boldsymbol{\sigma_0 = \pi_0}$: Assuming that the curve E_0 is $\mathbb{F}_p$-rational, we would have the configuration depicted in Figure 4.4, where $\omega = \mathrm{id}$ and $\psi = \phi^{(p)} \circ \hat{\phi}$. Given that $p \nmid d_B$ and since $p^{-1}\hat{M} = M_{\pi_0}^{-1}$, we have that

$$\psi \begin{pmatrix} S \\ T \end{pmatrix} = (p^{-1}d \mod d_B) \cdot \hat{M} \cdot \pi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot M_{\pi_0}^{-1} \cdot \pi \begin{pmatrix} S \\ T \end{pmatrix},$$

where $\pi : E \to E^{(p)}$ is the Frobenius isogeny and $\hat{M}$ is the matrix representing $\hat{\pi}_0$[23]. Therefore, one can apply Lemma 2 and be able to evaluate ψ, since in M-SIDH $d_B > d$ is always satisfied (either for Alice's secret isogeny or analogously for Bob's secret isogeny). Also in this case, it is expected that ψ is cyclic and therefore $\ker(\phi)$ can be explicitly recovered, as seen in Remark 18. In conclusion, the M-SIDH variant should be considered insecure if E_0 is defined over $\mathbb{F}_p$.

Remark 20. *This thought process can be generalized to the case where E_0 is not $\mathbb{F}_p$-rational, but such that it is possible to find an isogeny $\omega : E_0 \to E_0^{(p)}$ of small degree w. In this scenario, we consider the d^2w-isogeny $\psi = \phi^{(p)} \circ \omega \circ \hat{\phi}$ and the attack is successful if $d_B > d\sqrt{w}$. Therefore, if w is small enough, i.e. E_0 is not too far from its Frobenius conjugate $E_0^{(p)}$, then one should consider M-SIDH insecure.*

Remark 21. *It is noteworhty that all the attacks presented above can be also applied by targeting $\hat{\phi}$ instead of ϕ. Indeed, everything can be adapted analogously by simply swapping the roles of domain and codomain curves.*

[21] Alternatively, it would be possible to attack Bob's isogeny instead of Alice's since either $d_B > d\sqrt{w}$ or $d_A > d\sqrt{w}$. This is based on the assumption that one between d_A and d_B must be greater than the other, together with the assumed bound of the degrees for the attack.

[22] In this case we do not have enough information for retrieving the kernel of ψ.

[23] Indeed, given the definitions of M_{π_0} and $\hat{M}$, we have that

$$M_{\pi_0} \cdot \hat{M} = p \implies p^{-1} \cdot \hat{M} = M_{\pi_0}^{-1}.$$

$$\begin{tikzcd}
E_0^{(p)} \arrow[d, "\phi^{(p)}"'] & E_0 \arrow[l, bend right, "\pi_0"'] \arrow[l, bend left, "\mathrm{id}"] \arrow[d, "\phi"] \\
E^{(p)} & E \arrow[l, "\pi"]
\end{tikzcd}$$

Figure 4.4: Diagram illustrating the case $\sigma_0 = \pi_0$

4.1.3 Backdoors for M-SIDH

It has been shown in [CV23, §4.3], that an adversary could easily create a trapdoor for the M-SIDH variant by generating specific system parameters E_0, P_B, Q_B. This would be the case if such set of parameters can reach an instance of parameters that is susceptible to some attack, through an isogeny ε of low degree. The authors of [FMP23, §7.1] thought that any supersingular curve E_0 that had a sufficiently large endomorphism ring was enough to avoid this, but this assumption was proven wrong. Indeed, any random $\mathbb{F}_p$-rational supersingular curve could be attacked in some way previously described, even if it does not have any small endomorphism.

Let us see how it is possible to detect the existence of a backdoor for the given parameters. A backdoor might exist if there is an isogeny $\theta : E_0 \to E_0^{(p)}$ of small degree t. Thus, we want to verify that E_0 and $E_0^{(p)}$ are far apart within the $\mathbb{F}_{p^2}$-isogeny graph. If the isogeny θ exists, there will also be an endomorphism $\hat{\pi}_0 \circ \theta$ over E_0 of degree $t \cdot p$, where π_0 is the Frobenius isogeny of degree p. This scenario is depicted in Figure 4.5. Therefore, if we verify that the endomorphism $\hat{\pi}_0 \circ \theta$ does not exist, then we can say that E_0 and $E_0^{(p)}$ are not connected by an isogeny θ of small degree. This would mean that the parameters do not have a backdoor.

$$\begin{tikzcd}
E_0^{(p)} \arrow[d, "\phi^{(p)}"'] & E_0 \arrow[l, bend right, "\pi_0"'] \arrow[l, bend left, "\theta"] \arrow[d, "\phi"] \\
E' & E
\end{tikzcd}$$

Figure 4.5: Diagram illustrating the backdoor setting

As stated in [CV23, §4.3], there seems to not be any efficient test for this purpose and the only possibility appears to be to verify whether $\Phi_k(j(E_0), j(E_0)^p) = 0$ for all $k \in \{1, \dots, U\}$, evaluating the the k-th modular polynomial from Definition 17 for all possible path lengths[24]. Note that the bound U is dependent on the ratio between the degree d_A (resp. d_B) of the secret isogeny that one might want to recover and the order of the torsion points P_B, Q_B (resp. P_A, Q_A), that is d_B (resp. d_A). It is required that

$$U \geq \max \left\{ \frac{d_A^2}{d_B^2}, \frac{d_B^2}{d_A^2} \right\}$$

is the minimum number of tests to perform using the trivial approach of evaluating the k-th modular polynomial from Definition 17. Indeed, when wanting to recover $\phi = \phi_A$, the attacker will use the isogeny $\psi = \phi^{(p)} \circ \theta \circ \hat{\phi}$, that has degree $d_A^2 \cdot t$. Therefore, in order to apply the SIDH attacks, we require $d_B^2 > d_A^2 \cdot t$, that is $(d_B/d_A)^2 > t$. Applying the analogous thought process for attacking the secret isogeny ϕ_B, one obtains the bound on U. Such test can be made as efficient as possible by selecting d_A as close to d_B as possible, as proposed in [FMP23].

To answer the question of how to obtain a non-backdoored starting curve, the authors in [FMP23] propose that such curve should be generated through a multi-party computation protocol as described

[24] Since $j(E_0^{(p)}) = j(E_0)^p$ [Sil09, Example III.4.6], we have that $\Phi_k(j(E_0), j(E_0)^p) = 0$ if and only if E_0 and $E_0^{(p)}$ are connected by a k-isogeny.

in [Bas+23] by performing an isogeny walk over the full $\mathbb{F}_{p^2}$-isogeny graph. Additionally, even if this method returns a non-backdoored supersingular curve with high probability, the previously mentioned test can be performed to increase the conidence in the set of parameters.

4.2 The MD-SIDH variant

The goal of the *Masked-degree* variant, described in [FMP23, §3.2], is to mask the degree of the secret isogeny, as well as the torsion points. Let $p = d_A d_B f - 1$ be a prime number, with d_A and d_B being smooth coprime integers and f being a small cofactor. The main idea is to make Alice use cyclic isogenies of degree d'_A dividing d_A and make Bob use cyclic isogenies of degree d'_B dividing d_B.

Having a large amount of divisors for d_A and d_B will be crucial for the security of the scheme. For this reason, we will have to change the form of the values d_A and d_B, that in SIDH are of the form $d_A = \ell_A^{e_A}$ and $d_B = \ell_B^{e_B}$, since they have respectively only $e_A + 1$ and $e_B + 1$ divisors. In order to have more divisors, we will have $d_A = \ell_1^{a_1} \cdots \ell_t^{a_t}$ and $d_B = q_1^{b_1} \cdots q_t^{b_t}$, where t, the a_i's and the b_i's depend on the security parameter λ.

We hereby describe the MD-SIDH variant:

Public parameters: Let λ be the security parameter and let $t = t(\lambda) \in \mathbb{N}$ be an integer depending on λ. Let $p = d_A d_B f - 1$ be a prime such that $d_A = \prod_{i=1}^{t} \ell_i^{a_i}$ and $d_B = \prod_{i=1}^{t} q_i^{b_i}$ are coprime smooth integers, ℓ_i, q_i are distinct small primes, $d_A \approx d_B \approx \sqrt{p}$ and f is a small cofactor. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Set $E_0[d_A] = \langle P_A, Q_A \rangle$ and $E_0[d_B] = \langle P_B, Q_B \rangle$. The public parameters are $E_0, p, d_A, d_B, P_A, Q_A, P_B, Q_B$.

Public key (Alice): Alice samples a divisor d'_A of d_A uniformly at random and a random point $R_A \in_{\$} E_0[d'_A]$. Her secret key is $\mathsf{sk}_A = R_A$. She computes the isogeny $\phi_A : E_0 \to E_A := E_0/\langle R_A \rangle$ together with $\phi_A(P_B)$ and $\phi_A(Q_B)$. She samples a uniformly random integer $\alpha \in \mathbb{Z}/d_B\mathbb{Z}^\times$ and her public key is $\mathsf{pk}_A = (E_A, [\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$. The integer α is deleted.

Public key (Bob): Analogously, Bob samples a divisor d'_B of d_B uniformly at random and a random point $R_B \in_{\$} E_0[d'_B]$. His secret key is $\mathsf{sk}_B = R_B$. He computes the isogeny $\phi_B : E_0 \to E_B := E_0/\langle R_B \rangle$ together with $\phi_B(P_A)$ and $\phi_B(Q_A)$. He samples a uniformly random integer $\beta \in \mathbb{Z}/d_A\mathbb{Z}^\times$ and his public key is $\mathsf{pk}_B = (E_B, [\beta]\phi_B(P_A), [\beta]\phi_B(Q_A))$. The integer β is deleted.

Shared key (Alice): From $[\beta]\phi_B(P_A)$ and $[\beta]\phi_B(Q_A)$, Alice recovers $\langle R'_A \rangle = \langle \phi_B(R_A) \rangle$. She computes $E_{AB} := E_B/\langle R'_A \rangle$. The shared key is the value $j(E_{AB})$.

Shared key (Bob): Analogously, from $[\alpha]\phi_A(P_B)$ and $[\alpha]\phi_A(Q_B)$, Bob recovers $\langle R'_B \rangle = \langle \phi_A(R_B) \rangle$. He computes $E_{BA} := E_A/\langle R'_B \rangle$. The shared key is the value $j(E_{BA}) = j(E_{AB})$.

Remark 22. *When Alice generates her public key, she must be careful. Indeed, any attacker could apply Lemma 1 to recover $d'_A = \deg \phi_A$. To avoid this from happening, Alice must also sample an integer $\alpha \in_{\$} \mathbb{Z}/d_B\mathbb{Z}^\times$ uniformly at random. Note that, in this case it is not needed to sample α uniformly at random from $\mu_2(d_B)$ as it was done for the M-SIDH variant. Indeed, it is not possible to recover $\alpha^2 \mod d_B$ from $\alpha^2 \cdot \deg \phi_A$ as before, since $\deg \phi_A$ is masked and thus not known.*

4.2.1 Security analysis of the MD-SIDH variant

The problem assumed to be hard in order for the MD-SIDH variant to be secure is the following.

Problem 5 (MD-SIDH Problem)**.** *Let $d_A = \ell_1^{a_1} \cdots \ell_t^{a_t}$ and let $d_B = q_1^{b_1} \cdots q_t^{b_t}$ be two smooth coprime integers, let f be a small cofactor such that $p = d_A d_B f - 1$ is a prime, with $d_A \approx d_B$. Let $E_0/\mathbb{F}_{p^2}$ be a supersingular elliptic curve such that $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (d_A d_B f)^2$, set $E_0[d_B] = \langle P, Q \rangle$. Let $d'_A = \ell_1^{a'_1} \cdots \ell_t^{a'_t}$ be a uniformly random divisor of d_A and let α be a uniformly random element of $\mathbb{Z}/d_B\mathbb{Z}^\times$. Let $\phi : E_0 \to E_A$ be a uniformly random isogeny of degree d'_A.*
Given $E_0, P, Q, E_A, P' = [\alpha]\phi(P), Q' = [\alpha]\phi(Q)$, compute ϕ.

The MD-SIDH variant can be seen as a generalization of the M-SIDH variant, with non-fixed degree and with the scalars α and β not necessarily square roots of unity. Moreover, not having fixed isogeny degrees can help to achieve higher security as it would make harder to the adversary to execute the SIDH attacks as the knowledge of the degree of the secret isogeny is pivotal.

Let us first see that it is possible to recover the square-free part of the degree of the secret isogeny in an efficient way and then how it is possible to reduce any instance of MD-SIDH to an instance of M-SIDH under the assumption that the square-free part of the secret isogeny is known. This will mean that all the attacks that apply for M-SIDH can be extended to MD-SIDH.

Recovering the degree up to squares

As described in [FMP23, §5.1], let us see that the square-free part of the secret isogeny can be recovered in probabilistic polynomial time. The following lemma shows how it is possible to deduce a small set of candidates for the square-free part of d'_B.

Lemma 3. *Let $\mathsf{sf}(d'_B)$ be the square-free part of d'_B. Then, given E_0, P, Q, E_B and Q', there exists a probabilistic polynomial time algorithm that reduces the search space for $\mathsf{sf}(d'_B)$ to a set of order 2^γ where γ is expected to be small for most public parameter sets.*[25]

The main idea behind proving Lemma 3 is to build a map that takes as input a vector of exponents of some integer $\overline{d_B}$ and allows to determine whether $\overline{d_B} = \mathsf{sf}(d'_B)$ through polynomial-time computational steps.

Reduction to the M-SIDH variant

Let us present how it is possible to reduce any instance of MD-SIDH to an M-SIDH instance, as we assume that the square free part of d'_B is known and is $\mathsf{sf}(d'_B)$. Indeed from Lemma 3 we know that this is always possible to do.

Let $d_{B,0}$ be the largest divisor of d_B equal to d'_B up to squares (i.e. $d_{B,0} \mid d'_B$). Let β_0 be the divisor of d_B such that $d_{B,0} = \beta_0^2 \cdot d'_B$. Since d_B is smooth and we know $\mathsf{sf}(d'_B)$, we can compute the value of $d_{B,0}$. Now we can reduce the MD-SIDH problem to an M-SIDH problem. Let $\phi_0 = [\beta_0] \circ \phi$, that has $\deg \phi_0 = \beta_0^2 \cdot \deg \phi = \beta_0^2 d'_B \leq d_B$. Additionally, we have the following:

$$\begin{cases} P' = [\beta]\phi(P) = [(\beta\beta_0^{-1}) \cdot \beta_0]\phi(P) = [\beta\beta_0^{-1}]\phi_0(P) \\ Q' = [\beta]\phi(Q) = [(\beta\beta_0^{-1}) \cdot \beta_0]\phi(Q) = [\beta\beta_0^{-1}]\phi_0(Q) \end{cases}$$

We are able to recover $\beta^2 d'_B \mod d_A$ applying Lemma 1. With this knowledge, we can compute

$$\beta_1^2 = \beta^2 d'_B \cdot (\beta^2 d'_B)^{-1} \mod d_A = (\beta_0 \cdot \beta{-}1) \mod d_A.$$

We now randomly sample a square root β'_1 of $\beta_1^2 \mod d_A$. Such square root will be of the form $\beta'_1 = \mu\beta_1$, where μ is a square root of unity modulo d_A. We have that

$$\begin{cases} [\mu]\phi_0(P) = [\mu \cdot \beta_0]\phi(P) = [\mu \cdot \beta_0 \cdot \beta^{-1} \cdot \beta]\phi(P) = [\mu \cdot \beta_1]P' = [\beta'_1]P' \\ [\mu]\phi_0(Q) = [\mu \cdot \beta_0]\phi(Q) = [\mu \cdot \beta_0 \cdot \beta^{-1} \cdot \beta]\phi(Q) = [\mu \cdot \beta_1]Q' = [\beta'_1]Q' \end{cases}$$

Since $E_0, P, Q, E_B, [\mu]\phi_0(P) = [\beta'_1]P', [\mu]\phi_0(Q) = [\beta'_1]Q', \deg \phi_0 = \beta_0^2 d'_B$ are known quantities, one can solve for ϕ_0.

This is indeed an M-SIDH instance, except for the fact that the secret isogeny is not cyclic. However, as stated in [FMP23, §5.3], this detail is not a problem since the SIDH attacks do not have a restriction on the type of isogeny. Additionally, the authors in [FMP23, §5.4] state that the attacks presented for the M-SIDH variant still apply to the MD-SIDH variant, even if the secret isogeny is not cyclic.

[25] In [FMP23, §5.2] it is stated that it is computationally hard in practice to build d_A and d_B such that $\gamma \gg 2$. If such integers were to be computed, they would lead to an impractical scheme.

4.3 Effectiveness of these variants

Let us first justify why in [FMP23, §3.3] it is believed that the SIDH attacks do not extend to the variants presented throughout this chapter. First recall that the main idea of the attacks is to embed the target secret isogeny ϕ of degree d_B into a higher genus isogeny Ψ of known degree $d_A = d_B + a$, where d_A is the order of the torsion points of which we reveal the image through ϕ and $a = d_A - d_B$.

In the M-SIDH variant, Bob's secret isogeny has degree $\beta^2 d_B$, where β is such that $\beta^2 \equiv 1 \mod d_A$. Embedding Bob's secret isogeny in this scenario would lead to an auxiliary isogeny Ψ of degree $\beta^2 d_B + a$ that an attacker would want to use to recover Bob's secret isogeny. However, they would actually be clueless of what this degree could be, since the value β is unknown to the attacker. Therefore, the SIDH attacks cannot be applied. Note that, even though highly unlikely, if $\beta = \pm 1$ then $\beta^2 = 1$ and Ψ has degree $d_B + a$, allowing for a successful attack.

Similarly, in the MD-SIDH variant, Bob's secret isogeny has degree $\beta^2 d'_B$, where β is a random integer and d'_B is a random divisor of d_B. Embedding Bob's secret isogeny in this scenario would lead to an auxiliary isogeny Ψ of degree $\beta^2 d'_B + a$ that an attacker would want to use to recover Bob's secret isogeny. Just as in the M-SIDH case, they would not know what this degree could be and thus could not apply the SIDH attacks.

4.4 Parameters selection and efficiency

4.4.1 Selecting the starting curve E_0

From what we discussed in Section 4.1.1, the starting curve E_0 should not be an elliptic curve with a small-degree endomorphism for either scheme. Thus, we have a need for a setup algorithm that generates E_0 as a curve with no such small-degree endomorphisms. In order to achieve this, the authors in [FMP23, §7.1] present 3 possibilities, where they let the endomorphism ring of the curve be either

1. publicly known;
2. known by only one party (either Alice or Bob);
3. not known by anyone.

The first possibility has the main advantage of allowing anyone to verify that E_0 does not come equipped with a small endomorphism, by simply computing the norm of the shortest element in $\mathrm{End}(E_0)$, but might not be the most secure option. Indeed, using Bröker's algorithm [Brö09] there are two possibilities for generating the starting curve: one can run the algorithm to generate E_0 or, alternatively, one can compute a supersingular curve through the algorithm and then perform a random walk to reach E_0. Both these possibilities present security concerns: the first option is not valid since Bröker's algorithm generates curves that have small endomorphisms and the second option allows for the party performing the random walk on the graph to backdoor the scheme.

For the second possibility, we would require for one between Alice or Bob to generate the curve and not reveal its endomorphism ring. The problem with this scenario is that the party generating E_0 could backdoor the parameters and choose a weak curve, for which the attacks are particularly efficient.

The third possibility takes into account MPC techniques to generate a curve with unknown endomorphism ring. This approach stems from the fact that generating a supersingular elliptic curve with unknown endomorphism ring is a hard problem [MMP22, §3.1] and therefore, one might want to execute a multi-party computation that can simulate a trusted third party that performs a long random walk from a known curve, deleting the walk they used to reach the truly random supersingular elliptic curve E_0 [Bas+23].

Overall, the authors suggested that having E_0 being a curve with no small-degree endomorphism should be sufficient for avoiding attacks, but [CV23] showed that this is not the case, as discussed throughout Section 4.1.2 and Section 4.1.3.

4.4.2 Parameters selection

Within the present section, we will discuss the process for generating the public parameters for the variants discussed throughout this chapter, studying their increased size with respect to the ones required for the SIDH protocol.

M-SIDH

To generate the public parameters of M-SIDH in order to achieve AES-λ security (i.e., classical λ bits security with classical computer and $\lambda/2$ bits security with respect to a quantum computer), we will do as follows. Given λ, sample the $2t$ smallest primes for $t \geq 2 \cdot \lambda$ and partition them into two sets of the same size. Use the first set to compute d_A and use the second set to compute d_B, such that $d_A \approx d_B$. Now perform checks on $t - n + 1$, as discussed in Section 4.1.1, where n is the largest index such that $\sqrt{d_A} \leq \ell_n \dots \ell_t$ for ℓ_i distinct small prime factors of d_B. If $t - n + 1 < \lambda$, restart with a larger t. If $t - n + 1 \geq \lambda$, find a cofactor f such that $p = d_A d_B f - 1$ is prime. Overall, Table 4.1 taken from [FMP23, §7.2] contains the sizes of the prime p and the key sizes for AES-128 (NIST level 1), AES-192 (NIST level 3) and AES-256 (NIST level 5). The recommended primes are

$$\begin{aligned} p_{128} &= 2^2 \cdot \ell_1 \cdots \ell_{571} \cdot 10 - 1, \\ p_{192} &= 2^2 \cdot \ell_1 \cdots \ell_{851} \cdot 207 - 1, \\ p_{256} &= 2^2 \cdot \ell_1 \cdots \ell_{1131} \cdot 13 - 1, \end{aligned}$$

where ℓ_i's are odd primes and Alice will use $d_A = 2^2 \cdot \ell_2 \cdot \ell_4 \cdots \ell_{2t}$ and Bob will use $d_B = \ell_1 \cdot \ell_3 \cdots \ell_{2t-1}$.

AES	**NIST**	**prime p**	**secret key**	**public key**	**compressed pk**
128	level 1	5911 bits	≈ 369 bytes	4434 bytes	≈ 2585 bytes
192	level 3	9382 bits	≈ 586 bytes	7037 bytes	≈ 4103 bytes
256	level 5	13000 bits	≈ 812 bytes	9750 bytes	≈ 5687 bytes

Table 4.1: Suggested parameters for 128, 192 and 256 bits of security for the M-SIDH variant.

MD-SIDH

For what concerns the generation of MD-SIDH public parameters, we notice that this variant can be broken by the same attacks that break M-SIDH. Thus, we require for $t - n + 1 \geq \lambda$ in order to achieve AES-λ security, were n is the largest integer such that we can find a subset $S \subset \{1, \dots, t\}$ satisfying $\sqrt{d_B} \leq \prod_{i \in S} \ell_i^{a_i}$ and $n = t + 1 - \#S$. Additionally, since a Weil pairing computation can reduce the degree of the secret isogeny by a factor 2^t, we require also that there are $2^{\lambda+t}$ plausible degrees for each secret isogeny. In order to achieve this given the security parameter λ, sample the first $2 \cdot t$ primes for $t \geq \lambda$ and set the first λ exponents $a_i = b_i = 3$ for all $i \in \{1, \dots, \lambda\}$ and the remaining exponents to 1. Partition the exponents into two sets of the same size. Use the first set to compute d_A and use the second set to compute d_B, such that $d_A \approx d_B$. Now perform checks on $t - n + 1$, as discussed in Section 4.1.1. If $t - n + 1 < \lambda$, restart with a larger t. If $t - n + 1 \geq \lambda$, find a cofactor f such that $p = d_A d_B f - 1$ is prime. Overall, Table 4.2 taken from [FMP23, §7.3] contains the sizes of the prime p and the key sizes for AES-128 (NIST level 1), AES-192 (NIST level 3) and AES-256 (NIST level 5). The recommended primes are

$$\begin{aligned} p_{128} &= 2^3 \cdot \ell_1^3 \cdots \ell_{255}^3 \cdot \ell_{256} \cdots \ell_{839} \cdot 537 - 1, \\ p_{192} &= 2^3 \cdot \ell_1^3 \cdots \ell_{383}^3 \cdot \ell_{384} \cdots \ell_{1273} \cdot 131 - 1, \\ p_{256} &= 2^3 \cdot \ell_1^3 \cdots \ell_{511}^3 \cdot \ell_{512} \cdots \ell_{1811} \cdot 1485 - 1, \end{aligned}$$

where ℓ_i's are odd primes and Alice will use $d_A = 2^3 \cdot \ell_2^3 \cdot \ell_{\lambda-2}^3 \cdot \ell_\lambda \cdots \ell_{2t}$ and Bob will use $d_B = \ell_1^3 \cdot \ell_3^3 \cdots \ell_{\lambda-1}^3 \cdot \ell_{\lambda+1} \cdots \ell_{2t-1}$.

AES	**NIST**	**prime p**	**secret key**	**public key**	**compressed pk**
128	level 1	13810 bits	$\approx$ 863 bytes	10358 bytes	$\approx$ 6040 bytes
192	level 3	22291 bits	$\approx$ 1393 bytes	16719 bytes	$\approx$ 9751 bytes
256	level 5	31226 bits	$\approx$ 1951 bytes	23420 bytes	$\approx$ 13660 bytes

Table 4.2: Suggested parameters for 128, 192 and 256 bits of security for the MD-SIDH variant.

4.4.3 Efficiency analysis of M-SIDH

It is noteworthy that M-SIDH variant is more secure than the MD-SIDH variant at comparable parameter sizes. This motivates why we only analyze the efficiency of the former within this concluding section of the chapter.

Compressed public key sizes for M-SIDH are bigger than previously developed SIKE [Jao+17] protocols approximately by a factor of 6.8, 7.3 or 7.8 for achieving the same security guarantees of respectively 128, 192, 256 bits of AES-security.

For what concerns the computations of the M-SIDH variant, it is stated in [FMP23, §7.4] that they are similar to those of SIDH. In the new variant presented throughout the chapter, we will have to perform an additional negligible scalar multiplication that is needed to mask the images of the torsion points and also perform individual isogeny steps of degrees $O(\lambda \log \lambda)$ instead of 2 and 3. The authors state that, when neglecting $\log \log \lambda$, the computation cost of the overall scheme is $O(\lambda^{3/2} \log^{3/2} \lambda)$ field operations using square root Vélu's formulae, whereas SIDH computational cost is $O(\lambda^2 \log^2 \lambda)$. Overall, a increase in the run-time of $O(\sqrt{\lambda} \log^{3/2} \lambda)$ is expected, even though some implementation optimizations used for SIDH should be applicable also to the M-SIDH variant.

Chapter 5

FESTA – Fast Encryption from Supersingular Torsion Attacks

Within the present chapter, strongly referencing [BMP23], let us present a Public-key Encryption (PKE) scheme called FESTA, that stands for Fast Encryption from Supersingular Torsion Attacks. It is an isogeny-based PKE protocol that leverages the SIDH attacks to construct a trapdoor function, with the main goal of being both practical and efficient. Moreover, FESTA results IND-CCA secure in the QROM through the use of the standard OAEP transform [Ebr22] and also IND-CCA secure in the standard model through the use of a generic transform presented in [HKW20].

The main idea of the FESTA trapdoor function is to use the SIDH attacks in a constructive way to invert the function. Once we will have seen how this is possible, we will see how to use such trapdoor function to construct an IND-CCA secure PKE.

5.1 Notation and preliminaries

Let us introduce some useful notation and also recall some cryptographic notions that will be used throughout the entire chapter when talking about trapdoor functions, public-key encryption schemes and indistinguishability security for PKEs.

Let λ be the security parameter. We say that a function is negligible if, for all positive integers $c \in \mathbb{Z}_{>0}$, there exists an integer N such that $|f(x)| < x^{-c}$ for all $x > N$. In this case, we will use the notation $\mathsf{negl}(\cdot)$ to identify a negligible function. For a positive integer $t \in \mathbb{Z}_{>0}$, its square-free part will be denoted by $\mathsf{sf}(t)$. Moreover, to indicate that an element x is sampled uniformly at random between the elements of the set $\mathcal{X}$, we will use the notation $x \leftarrow_{\$} X$.

Moreover, TorGen will represent a deterministic algorithm that returns a basis $\{P, Q\}$ of the n-torsion subgroup of E, i.e. $E[n]$, when given in input a supersingular curve E and an integer n. Additionally, with a slight abuse of notation, given four isogenies $\phi_{i,j} : E_i \to E_j$ and two points $P_i \in E_i$ for $i = 1, 2$ and $j = 3, 4$, we can evaluate the isogeny

$$\begin{pmatrix} \phi_{1,3} & \phi_{2,3} \\ \phi_{1,4} & \phi_{2,4} \end{pmatrix} : E_1 \times E_2 \to E_3 \times E_4$$

at $\begin{pmatrix} P_1 \\ P_2 \end{pmatrix}$ by computing

$$\begin{pmatrix} \phi_{1,3} & \phi_{2,3} \\ \phi_{1,4} & \phi_{2,4} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} \phi_{1,3}(P_1) + \phi_{2,3}(P_2) \\ \phi_{1,4}(P_1) + \phi_{2,4}(P_2) \end{pmatrix}.$$

Furthermore, scaling the points P_1, P_2 by a matrix A can be interpreted as above, with the elements of the matrix being scalar endomorphism:

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} [\alpha](P_1) + [\beta](P_2) \\ [\gamma](P_1) + [\delta](P_2) \end{pmatrix}.$$

Turning our focus to providing the notions relevant to properly discuss the FESTA public-key encryption protocol, we present the following relevant concepts. They are adapted from [BS20], as mentioned also by the authors in [BMP23]. As for any public-key encryption scheme, the central element of FESTA is the concept of *injective trapdoor function.* On a high-level, it can be understood as a one-way function that additionally can be inverted efficiently due to additional knowledge by the party that intends to invert the function. More formally, we introduce the following definition taken from [BMP23, §2.1].

Definition 19 (Family of trapdoor functions)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. A* family of trapdoor functions *is a triple of algorithms* $(\mathsf{KeyGen}, f, f^{-1})$ *such that:*

- $\mathsf{KeyGen}(\lambda) \to_{\$} (\mathsf{sk}, \mathsf{pk})$*:* $\mathsf{KeyGen}(\lambda)$ *is a probabilistic key generation algorithm that outputs a secret key* sk *and a public key* pk *for a given security parameter λ;*

- $f(\mathsf{pk}, x) \to y$*: f is a deterministic algorithm that outputs $y \in \mathcal{Y}$ taking as input a public key* pk *and $x \in \mathcal{X}$;*

- $f^{-1}(\mathsf{sk}, y) \to x$*: f^{-1} is a deterministic algorithm that outputs $x \in \mathcal{X}$ taking as input the secret key* sk *and $y \in \mathcal{Y}$.*

Every trapdoor function f should satisfy two additional properties:

- *Correctness: $f^{-1}(\mathsf{sk}, f(\mathsf{pk}, x)) = x$, for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KeyGen}$ *and for all $x \in \mathcal{X}$;*

- *One-wayness: Given the public key* pk *and aa valid output $y \in \mathcal{Y}$ computed using* $(\mathsf{pk}, \mathsf{sk})$*, any probabilistic polynomial-time adversary cannot compute $x \in \mathcal{X}$ such that $f(\mathsf{pk}, x) = y$ with probability greater than* $\mathsf{negl}(\lambda)$*.*

Within the FESTA framework, the notion of *partial-domain trapdoor function* is used, that is a stronger version of trapdoor functions where it is required that even recovering a smaller portion of the input is hard. Let us make this definition explicit, as stated in [Ebr22, §2.2].

Definition 20 (Quantum partial-domain one-way function)**.** *Let $\mathcal{X}_0, \mathcal{X}_1$ and $\mathcal{Y}$ be three finite sets. A function $f : \mathcal{X}_0 \times \mathcal{X}_1 \to \mathcal{Y}$ is a* quantum partial-domain one-way function *if, for any polynomial-time quantum adversary $\mathcal{A}$, the following holds:*

$$\mathbb{P}\Big(s' = s \;\Big|\; s \leftarrow_{\$} \mathcal{X}_0, t \leftarrow_{\$} \mathcal{X}_1, s' \leftarrow \mathcal{A}(f(s,t))\Big) < \mathsf{negl}(\lambda).$$

Since this chapter revolves around the construction of a Public-key Encryption (PKE) scheme from a quantum partial-domain one-way function, let us provide a formal definition taken from [BMP23, §2.1].

Definition 21 (Public-key Encryption)**.** *A* public-key encryption scheme *is a triple of efficient algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

- $\mathsf{KeyGen}(\lambda) \to_{\$} (\mathsf{sk}, \mathsf{pk})$*:* KeyGen *is a probabilistic key generation algorithm that outputs a secret key* sk *and a public key* pk *for a given security parameter λ;*

- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$*:* Enc *is a probabilistic algorithm that, given a public key* pk *and a message m, returns a ciphertext* ct*;*

- $\mathsf{Dec}(\mathsf{sk}, ct) \to m$*:* Dec *is a deterministic algorithm that returns a message m having a ciphertest* ct *and a secret key* sk *as input.*

Additionally, a public-key encryption scheme must satisfy the correctness *property. A public-key encryption scheme is correct if* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$ *for all possible messages m and for all possible* $(\mathsf{sk}, \mathsf{pk}) \leftarrow_{\$} \mathsf{KeyGen}$*.*

To conclude this section, we recall that to evaluate the security of a public-key encryption scheme the following two notions of indistinguishability are used: security against a chosen plaintext attack (CPA), and security against a chosen ciphertext attack (CCA). It will be shown that the FESTA PKE (constructed in 5.4) is IND-CCA secure. On a high level, this means that, given two random plaintexts, any probabilistic polynomial-time adversary is not capable of distinguishing which message of the two has been encrypted, even by asking to decrypt as many ciphertexts (different from the challenge ciphertext) as they want at any point during the attack game.

5.2 The FESTA trapdoor function

Before delving into the details, let us present an overview of FESTA, a family of trapdoor functions resistant to a quantum adversary [BMP23, §3]. See Figure 5.1. The trapdoor key is the couple formed by a random matrix A and an isogeny $\phi_A : E_0 \to E_A$, where the curve E_A is part of the public parameters together with R_A and S_A, that are the images of the points of a torsion basis $\{P_b, Q_b\}$ multiplied by the random matrix A. The matrix A helps in neutralizing the SIDH attacks to recover the secret isogeny ϕ_A, since the torsion points are scaled due to a similar reasoning to what was presented in Chapter 4. Indeed, we will see that one-wayness of the FESTA trapdoor function is ensured by an analogous argument.

The one-way function takes as input the two isogenies $\phi_1 : E_0 \to E_1$ and $\phi_2 : E_A \to E_2$ and a random matrix B. To evaluate the FESTA one-way function, one will compute the two isogenies ϕ_1 and ϕ_2 that start from two different curves linked by the secret isogeny ϕ_A. The outputs of the function are the image curves E_1 and E_2, together with the scaled images of the torsion basis on E_0 and E_A under the action of ϕ_1 and ϕ_2, respectively, multiplied by the matrix B.

In order to successfully invert the trapdoor function, we will see that it is necessary for the two matrices A and B to commute with each other, just like two diagonal matrices would do. Commutativity of A and B is required because scaling the points on E_2 by the matrix A^{-1} will produce the *exact* images of the torsion points on E_1 under the isogeny $\psi := \phi_2 \circ \phi_A \circ \hat{\phi}_1$, in fact undoing the scaling. Therefore, we will see how an application of the SIDH attacks will allow to recover the input ϕ_1, ϕ_2 and B to the party holding the trapdoor, while being infeasible for any party that does not have knowledge of the secret matrix A.

$$\begin{tikzcd}
\begin{pmatrix}P_b\\Q_b\end{pmatrix} & & \begin{pmatrix}R_A\\S_A\end{pmatrix} = A\cdot\begin{pmatrix}\phi_A(P_b)\\\phi_A(Q_b)\end{pmatrix} \\
E_0 \arrow[rr, "\phi_A"] \arrow[d, "\phi_1"'] & & E_A \arrow[d, "\phi_2"] \\
E_1 \arrow[rr, dashed, bend left, "\psi"] & & E_2 \\
\begin{pmatrix}R_1\\S_1\end{pmatrix} = B\cdot\begin{pmatrix}\phi_1(P_b)\\\phi_1(Q_b)\end{pmatrix} & & \begin{pmatrix}R_2\\S_2\end{pmatrix} = B\cdot\begin{pmatrix}\phi_2(R_A)\\\phi_2(S_A)\end{pmatrix}
\end{tikzcd}$$

Figure 5.1: Diagram illustrating the FESTA trapdoor function, adapted from [BMP23, §1].

Hence, a key element for inverting the FESTA trapdoor function is the application of the SIDH attacks, that leverage the information leaked by the images of the torsion points. Within the FESTA framework we will write TorAtk to denote a generic attack that exploits the information provided by the torsion points. Will see the details in Section 5.5. Regardless, when given the points $P' = \psi(P)$ and $Q' = \psi(Q)$, where P and Q are such that $\langle P, Q\rangle = E[2^b]$ for some $b \in \mathbb{Z}_{>0}$ and $\psi : E \to E'$ is an unknown isogeny of known degree d, $\mathsf{TorAtk}(E, P, Q, E', P', Q', d)$ returns a description of the secret isogeny $\psi : E \to E'$.

Therefore, to invert the FESTA trapdoor function, we will need to apply the SIDH attacks, that take into account higher-dimensional isogenies. Since a well known result of Richelot [Smi+05, Chapter 8] provides an efficient way to compute $(2, 2)$-isogenies between Jacobians of genus-2 hyperelliptic curves, the FESTA framework is restricted to d_i-isogenies such that $d_1 + d_2 = 2^b$, for some $b \in \mathbb{Z}_{>0}$. Thus, the protocol is implemented only through the use of $(2, 2)$-isogenies in dimension two, motivated by the lack of efficient ways to compute more generic (ℓ, ℓ)-isogenies.

Let us formalize a bit more the details of the FESTA trapdoor function. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, let $\{P_b, Q_b\}$ be such that $E_0[2^b] = \langle P_b, Q_b\rangle$ and let the degrees of the isogenies be known. These constitute the public parameters shared across different functions of the same trapdoor family. The public key of each trapdoor function is (E_A, R_A, S_A), where E_A is the image curve of the isogeny $\phi_A : E_0 \to E_A$ and R_A, S_A are the images of the points of the torsion basis $\{P_b, Q_b\}$ multiplied by the random matrix $A \in \mathcal{M}_b$. Let us postpone the accurate description of the set of all possible matrices $\mathcal{M}_b$ to after the accurate trapdoor inversion procedure is described in Algorithm 2. The set of all possible public keys is described by following set:

$$\mathcal{A}^{\mathsf{pk}} := \left\{ (E_A, R_A, S_A) \;\middle|\; \begin{array}{l} \phi_A : E_0 \to E_A,\ \deg(\phi_A) = d_A, \\ A \in \mathcal{M}_b,\ \begin{pmatrix} R_A \\ S_A \end{pmatrix} = A \cdot \begin{pmatrix} \phi_A(P_b) \\ \phi_A(Q_b) \end{pmatrix} \end{array} \right\}.$$

For every public key $(E_A, R_A, S_A) \in \mathcal{A}^{\mathsf{pk}}$, the notation $f_{(E_A,R_A,S_A)}$ highlights the dependence of the trapdoor function f from the public key. To evaluate the one-way function $f_{(E_A,R_A,S_A)}$, one will compute the isogeny $\phi_1 : E_0 \to E_1$ of degree d_1 and the isogeny $\phi_2 : E_A \to E_2$ of degree d_2. Note that d_1, d_2, d_A are taken smooth and coprime. The outputs of the function are the image curves E_1 and E_2, together with the scaled torsion images obtained by computing the images of the torsion basis on E_0 and E_A under the action of ϕ_1 and ϕ_2, respectively, and multiplying both of them by the matrix $B \in \mathcal{M}_b$. Therefore, the one-way function will yield an array $(E_1, R_1, S_1, E_2, R_2, S_2)$. Let us summarize this procedure in Algorithm 1 taken from [BMP23, §3].

```
Algorithm 1 $f_{(E_A,R_A,S_A)}(\langle K_1\rangle, \langle K_2\rangle, B)$
Input: Two cyclic subgroups $\langle K_1\rangle \subset E_0[d_1]$ and $\langle K_2\rangle \subset E_0[d_2]$ of order $d_1$ and $d_2$ respectively, and
    the matrix $B \in \mathcal{M}_b$.
Output: The array $(E_1, R_1, S_1, E_2, R_2, S_2)$.
  Compute the $d_1$-isogeny $\phi_1 : E_0 \to E_1$ having kernel $\langle K_1\rangle$.
  Compute the $d_2$-isogeny $\phi_2 : E_A \to E_2$ having kernel $\langle K_2\rangle$.
  Acting with scalar multiplication, compute
      $\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = B \cdot \begin{pmatrix} \phi_1(P_b) \\ \phi_2(Q_b) \end{pmatrix}, \qquad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = B \cdot \begin{pmatrix} \phi_2(R_A) \\ \phi_2(S_A) \end{pmatrix}.$
  return $(E_1, R_1, S_1, E_2, R_2, S_2)$
```

To invert the function $f_{(E_A,R_A,S_A)}$, we want to have the knowledge of $(\psi(R_1), \psi(S_1))^T$, with $\psi = \phi_2 \circ \phi_A \circ \hat{\phi}_1$ so that then we can apply the SIDH attacks to the $d_1 d_A d_2$-isogeny ψ and recover the kernels of ϕ_1 and ϕ_2 as well as the secret matrix B. However, in order to do this, we need to scale the points R_2, S_2 on E_2 to undo the scaling by A and therefore require A and B to be diagonal (so that they commute). In more detail, we can recover the isogeny ψ as $\mathsf{TorAtk}(E_1, R_1, S_1, E_2, R_2', S_2', d_1 d_A d_2) = \psi$, where R_2' and S_2' are computed as described in Algorithm 2 taken from [BMP23, §3]. The trapdoor function can be inverted using any type of attack to SIDH that leverages the information provided by the torsion points, given a starting curve of unknown endomorphism ring. More details will be provided in Section 5.5.1.

Algorithm 2 relies on the fact that

$$d_1 \cdot \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = B \cdot A \cdot B^{-1} \cdot \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix} = A \cdot \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix}. \tag{5.1}$$

Let us see why (5.1) is true:

$$\begin{aligned} \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix} &= \psi \begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \phi_2 \circ \phi_A \circ \hat{\phi}_1 \begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = B \cdot \phi_2 \circ \phi_A \circ \hat{\phi}_1 \circ \phi_1 \begin{pmatrix} P_b \\ Q_b \end{pmatrix} = B \cdot [d_1] \cdot \phi_2 \circ \phi_A \begin{pmatrix} P_b \\ Q_b \end{pmatrix} \\ &= B \cdot A^{-1} \cdot [d_1] \cdot \phi_2 \begin{pmatrix} R_A \\ S_A \end{pmatrix} = d_1 \cdot B \cdot A^{-1} \cdot B^{-1} \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}. \end{aligned}$$

Where we have used the following facts:

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = B \cdot \begin{pmatrix} \phi_1(P_b) \\ \phi_1(Q_b) \end{pmatrix},\ \hat{\phi}_1 \circ \phi_1 = [\deg \phi_1] = [d_1],\ \phi_A \begin{pmatrix} P_b \\ Q_b \end{pmatrix} = A^{-1} \begin{pmatrix} R_A \\ S_A \end{pmatrix},\ \phi_2 \begin{pmatrix} R_A \\ S_A \end{pmatrix} = B^{-1} \begin{pmatrix} R_2 \\ S_2 \end{pmatrix},$$

Algorithm 2 $f^{-1}_{(E_A,R_A,S_A)}(E_1,R_1,S_1,E_2,R_2,S_2)$

```
Input: An array $(E_1,R_1,S_1,E_2,R_2,S_2)$, the trapdoor $(A \in \mathcal{M}_b,\ \phi_A : E_0 \to E_A)$.
Output: $(\langle K_1\rangle, \langle K_2\rangle, B)$ such that $f_{(E_A,R_A,S_A)}(\langle K_1\rangle, \langle K_2\rangle, B) = (E_1,R_1,S_1,E_2,R_2,S_2)$.
1: Recover $(R'_2, S'_2)$ by inverting $A$ and acting with scalar multiplication:
       $$\begin{pmatrix}\psi(R_1)\\ \psi(S_1)\end{pmatrix} = \begin{pmatrix}R'_2\\ S'_2\end{pmatrix} = d_1 \cdot A^{-1} \cdot \begin{pmatrix}R_2\\ S_2\end{pmatrix}.$$
2: Compute $\psi = \phi_2 \circ \phi_A \circ \hat{\phi}_1 : E_0 \to E_2$ via $\mathsf{TorAtk}(E_1,R_1,S_1,E_2,R'_2,S'_2,d_1 d_A d_2)$.
3: Recover the kernel $\langle K_1\rangle$ of the $d_1$-isogeny $\phi_1 : E_0 \to E_1$ from $\psi$ using $\phi_A$.
4: Recover the kernel $\langle K_2\rangle$ of the $d_2$-isogeny $\phi_2 : E_A \to E_2$ from $\psi$ using $\phi_A$.
5: Compute $B \in \mathcal{M}_b$ such that
       $$\begin{pmatrix}R_1\\ S_1\end{pmatrix} = B \cdot \begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix}.$$
6: return $(\langle K_1\rangle, \langle K_2\rangle, B)$
```

as well as the fact that the matrices A and B commute with each other.

For what concerns the set from which A and B are sampled, the most reasonable choice is to let $\mathcal{M}_b$ be the commutative subset of invertible diagonal matrices over $\mathbb{Z}/2^b\mathbb{Z}$ modulo $\langle -\mathbf{I}_2\rangle$, where $\mathbf{I}_2$ is the 2×2 identity matrix. This is tied to the fact that torsion-based attacks only recover isogenies up to automorphisms, and, in the FESTA framework, the automorphism groups of the curve E_1 and E_2 are exactly $\langle -\mathsf{id}\rangle$[26].

The FESTA trapdoor function is correct, since TorAtk recovers the unique isogeny up to automorphism whenever the starting curve has j-invariant such that $j \notin \{0, 1728\}$. Indeed, the isogeny ψ is uniquely determined by its action on the 2^b torsion [MP19, §4] and therefore, the isogeny of degree $d_1 d_A d_2$ that maps $(R_1, S_1)^T$ to $(R'_2, S'_2)^T$ is unique. Thus, the kernels are uniquely defined, since all the automorphisms are trivially in the set $\mathrm{Aut}(E) = \langle -\mathrm{id}\rangle$, and the images of torsion points are defined up to an automorphism (or up to inversion, depending on how one prefers to view it). Moreover, since the matrix B is invertible, the torsion point scaling is also an injection. In conclusion, it is very likely that the function f is injective and that the inversion algorithm yields the correct output.

Remark 23. *Just like in SIDH, we can represent each isogeny with an element in $\mathbb{Z}/d\mathbb{Z}$, since its kernel can be expressed as $\langle P + [x]Q\rangle$ with $x \in \mathbb{Z}/d\mathbb{Z}$, where $\langle P, Q\rangle = E[d]$ for some $d \in \mathbb{Z}$. Since this representation is injective if the automorphisms on the curve E are only $\pm\mathsf{id}$, we choose the starting curve E_0 to have $j(E_0) \notin \{0, 1728\}$. Therefore, the domain of $f_{(E_A,R_A,S_A)}$ can be expressed as $\mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b$.*

5.3 Security analysis of the FESTA trapdoor

As it is done in [BMP23, §4], let us present an analysis of the security of the FESTA trapdoor function by generalizing the classic isogeny problems presented in [DJP14, §5] to the scaled torsion points scenario.

Problem 6 (Decisional isogeny with scaled-torsion (DIST) problem)**.** *Let E_0 be a supersingular elliptic curve, and P_0, Q_0 be two points generating $E_0[n]$, for some smooth order n. Let d be a smooth degree, coprime with n. Consider the two following distributions:*

- *$\mathcal{D}_0 = \{E_1, P_1, Q_1\}$ be a distribution, where E_1 is the codomain of a d-isogeny $\phi : E_0 \to E_1$, and the points P_1, Q_1 are computed as*

$$\begin{pmatrix}P_1\\ Q_1\end{pmatrix} = A \cdot \begin{pmatrix}\phi(P_0)\\ \phi(Q_0)\end{pmatrix},$$

[26] Unless $j(E_1) \vee j(E_2) \in \{0, 1728\}$, which happens with negligible probability.

where we sample the matrix $A \leftarrow_{\$} \mathcal{M}_n$.

- *$\mathcal{D}_1 = \{E_1, P_1, Q_1\}$, where E_1 is a random supersingular elliptic curve with the same order as E_0 and $\{P_1, Q_1\}$ is a random basis of $E_1[n]$.*

Given an elliptic curve E_1 and two points P_1, Q_1, sampled with probability $1/2$ from either of the two distributions $\mathcal{D}_0$ or $\mathcal{D}_1$, distinguish from which distribution the values were sampled.

Problem 7 (Computational isogeny with scaled-torsion (CIST) problem)**.** *Let $\phi : E_0 \to E_1$ be an isogeny of smooth degree d between supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ and let n be a smooth integer coprime with d. Let $\{P_0, Q_0\}$ be a basis of $E_0[n]$. Let $\{P_1, Q_1\}$ be a basis of $E_1[n]$ obtained as*

$$\begin{pmatrix} P_1 \\ Q_1 \end{pmatrix} = A \cdot \begin{pmatrix} \phi(P_0) \\ \phi(Q_0) \end{pmatrix},$$

where we sample the matrix $A \leftarrow_{\$} \mathcal{M}_n$.

Given the curve E_0 with the basis $\{P_0, Q_0\}$ of $E_0[n]$ and the curve E_1 with a basis $\{P_1, Q_1\}$ of $E_1[n]$, compute the isogeny ϕ.

Since Problem 6 is the decisional variant of Problem 7, it is clear that the former is at least as hard as the latter. In other words, solving Problem 7 implies that one is also able to solve Problem 6. However, also the converse is true due to the search-to-decision reduction[27] for classical isogeny problems [GV18]. Assuming that the CIST problem is hard guarantees that recovering the trapdoor information from the public parameters is hard. However, we now must introduce the following problem since the FESTA one-way function presents the issue of returning two pairs of curves with their relative torsion points scaled by the same matrix. Thus the correlated scaling might simplify the inversion of the one-way function.

Problem 8 (Computational isogeny with double scaled-torsion (CIST^2) problem)**.** *Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, and let E_0' be a random supersingular elliptic curve defined over the same field. Consider two isogenies $\phi : E_0 \to E_1$ and $\phi' : E_0' \to E_1'$ of smooth degree d and d', respectively. Let n be a smooth integer coprime with d and d', and let A be a matrix samples as $A \leftarrow_{\$} \mathcal{M}_n$.*

Given the curves E_0, E_1, E_0', E_1', two bases $\{P, Q\}$ of $E_0[n]$ and $\{P', Q'\}$ of $E_0'[n]$, and the points

$$\begin{pmatrix} R \\ S \end{pmatrix} = A \cdot \begin{pmatrix} \phi(P) \\ \phi(Q) \end{pmatrix}, \quad \begin{pmatrix} R' \\ S' \end{pmatrix} = A \cdot \begin{pmatrix} \phi'(P') \\ \phi'(Q') \end{pmatrix},$$

compute the isogenies ϕ and ϕ'.

Note that the hardness of the CIST^2 problem implies the hardness of the CIST problem, since the former provides additional information (two sets of torsion images scaled by the same matrix instead of only one). Indeed, if an adversary can break the CIST problem, we have the following reduction. Let the attacker have access to two CIST samples, scaled by the same matrix A. So, if the attacker can recover the isogeny in one of the two CIST samples, then they can also obtain the exact torsion images in the other sample by undoing the scaling by A and simply multiplying the sample by the matrix A^{-1}. Thus, the second isogeny can be recovered in polynomial time. The converse implication seems to not be true in general. Moreover, the authors suggest that the correlated scaling matrix A does not seem to reveal valuable information to be exploited by an attacker to break a CIST instance. This statement stems from the fact that the two samples have different starting curves and apply isogenies of different degrees.

Relying on the hardness of Problem 8, it is possible to show the one-wayness of the FESTA trapdoor function, as is done in [BMP23, §4].

[27] A "search-to-decision" reduction involves breaking a computation problem, given access to an *oracle* that can break the decisional version of the same problem. Note that many times an oracle can be expressly crafted, as it is done for the attack to SIDH presented in Section 3.1.3 [CD23] though the use of Kani's criterion.

Let us now discuss possible strategies that could be used by an attacker to solve the mentioned problems. In response to that, let us present arguments to justify the presumed hardness of the corresponding computational assumptions, ensuring the security of FESTA. Firstly, let us see why FESTA is not affected by the SIDH attacks. Indeed, even though FESTA reveals images of torsion points of sufficiently large order, these are scaled by a random diagonal matrix. Let us present why, even though an attacker can reduce the number of scalars protecting the security of FESTA, the security of the scheme is not affected. We state that an attacker can recover the determinant of the diagonal matrix

$$A = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$$

through a Weil pairing computation. Indeed, the following holds true:

$$\begin{aligned} e([\alpha]\phi(P), [\beta]\phi(Q)) &= e(\phi(P), \phi(Q))^{\alpha\beta} = e(P, \hat{\phi} \circ \phi(Q))^{\alpha\beta} \\ &= e(P, [\deg \phi]Q)^{\alpha\beta} = e(P, Q)^{\alpha\beta \deg \phi} \end{aligned}$$

by efficiently solving an easy instance of the discrete logarithm problem, this allows to remove one variable multiplying $Q' = [\beta]\phi(Q)$ by $(\alpha\beta)^{-1} \mod 2^b$, that produces the couple of points

$$P' = [\alpha]\phi(P), \; Q'' = [1/\alpha]\phi(Q).$$

We now have that P' and Q'' are the images of the torsion points scaled by the diagonal matrix

$$A' = \begin{pmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{pmatrix}$$

of determinant one and dependent uniquely on α. As the authors in [BMP23, §4] suggest, this modification does not change the security guarantees since α is sampled uniformly at random from a set of exponential cardinality, maintaining the inversion of the trapdoor hard. This discussion leads to the realization that the matrix A can be chosen such that its determinant is one without affecting the security of the protocol, similarly to what was noticed in Section 4.1 with the M-SIDH variant where we restricted the selection of the scalar α to the set $\mu_2(d_B)$. Even though the SIDH attacks do not apply to the FESTA trapdoor function, the question of whether they could be extended to account for this case is natural. It is noteworthy that FESTA reveals significantly less torsion information than SIDH or even the variants discussed in Chapter 4: M-SIDH and MD-SIDH. This is due to the fact that in SIDH, M-SIDH and MD-SIDH the parties must reveal the values $[\alpha]\phi(P)$ and $[\alpha]\phi(Q)$ (where $\alpha = 1$ in SIDH), while in FESTA they reveal $[\alpha]\phi(P)$ and $[\alpha^{-1}]\phi(Q)$, where $\alpha \neq \alpha^{-1}$ with overwhelming probability. However, as will be presented in Section 5.3.1, some instances of FESTA can be attacked just like we saw for M-SIDH in Section 4.1.2.

Secondly, another attack strategy would be for the adversary to guess (brute forcing) the value α used for scaling the images of the torsion points, by trying to recover only part of α. The strategy for this attack would be to scale the points $P' = [\alpha]\phi(P)$ and $Q' = [\alpha^{-1}]\phi(Q)$ by 2^{b-j}, in order to work with points of order 2^j, obtaining:

$$2^{b-j}P' = [\alpha \mod 2^j]\phi([2^{b-j}]P), \; 2^{b-j}Q' = [\alpha^{-1} \mod 2^j]\phi([2^{b-j}]Q).$$

This strategy focuses on guessing $\alpha \mod 2^j$ if the attacker is able to recover enough information on the images of the torsion points of order 2^j. However, this strategy fails since FESTA requires isogenies of degree $2^{2\lambda}$ and therefore we would need $j = \lambda$, obtaining a computational cost of 2^λ. To conclude, the two strategies just presented make us believe that the best known attack to solve Problem 6 and Problem 7 is by simply applying a variant of the meet-in-the-middle attack discussed in Section 2.2 that ignores the additional information provided by the torsion points.

Even in the case of an adversary with quantum-computing capabilities, the best strategy to apply seems to be the meet-in-the-middle attack discussed in Section 2.2, without exploiting the scaled torsion images. Therefore, the authors rely on the quantum security analysis presented for SIDH [DJP14, §5.1], for which sufficiently long isogenies are hard to recover even with a quantum computer.

Remark 24. *As pointed out by the authors, within the literature several isogeny-based protocols have been shown to be vulnerable when one assumes the knowledge of the endomorphism ring [BKW20; Bas+21], or when the starting curve is maliciously chosen [Que+21], or when the known endomorphism ring contains small endomorphisms as previously discussed for M-SIDH and MD-SIDH in Chapter 4. To solve these issues, a trusted setup appears to be a necessary countermeasure to adopt [Bas+23]. However, given the fact that the endomorphism ring of E_0 is not a crucial information to keep secret, these type of attacks do not apply for solving Problem 7 and any attack that acts by exploiting the unknown endomorphism ring could be avoided altogether by including its description in the public key.*

As we will see in Section 5.3.1, it is possible to recover an isogeny from its scaled action and efficiently solve Problem 7 under the assumption that the attacker knows an endomorphism on E_0 that performs scalar multiplication on the torsion basis $\{P, Q\}$. However, as the authors in [BMP23] highlight, the FESTA PKE scheme is subject to such attack with negligible probability in the security parameter since the basis are generated randomly.

5.3.1 Generalized lollipop attack for FESTA

Within the present section we will adopt the approach previously introduced in Section 4.1.2, strongly referencing [CV23]. Indeed, we will see how the same framework previously described will be applied to attack certain overstretched instances of FESTA. For the reader's convenience and for the sake of clarity, there will be some repetitions from Section 4.1.2. To attack FESTA, there are two options: either try to recover the private key ϕ_A (or the matrix A) or try to invert the one-way function by recovering the input ϕ_1, ϕ_2, B. The authors notice that both cases are instances of the same problem, where the only differences concern the degree of the secret isogenies. More specifically, in the first case we want to recover a secret isogeny of secret degree d_A given 2^b-torsion information, while in the second case we want to recover a secret isogeny of secret degree d_1 or d_2 given 2^b-torsion information. Indeed, in the second case, once the secret matrix B is recovered, it is possible to obtain the second isogeny as described when comparing the hardness of CIST and CIST^2.

Intuition of the attack

Consider E_0 to be a $\mathbb{F}_p$-rational curve, let $E^{(p)}$ denote the Frobenius conjugate of E (raising all coefficients of E to the p-th power) and $\pi : E \to E^{(p)}$ the connecting Frobenius isogeny. The curve $E^{(p)}$ will play the role of the curve E' unobservant of the unknown value α. Moreover, the isogeny $\phi^{(p)} : E_0 \to E^{(p)}$ is the Frobenius conjugate of $\phi : E_0 \to E$ and is such that

$$\phi^{(p)} \circ \pi_0 = \pi \circ \phi,$$

where π_0 is the Frobenius endomorphism over E_0. Let us consider the isogeny $\psi = \phi^{(p)} \circ \hat{\phi}$ having degree d^2.

$$\begin{tikzcd}
& E_0 \arrow[ld, "\phi^{(p)}"'] \arrow[rd, "\phi"] \arrow[loop below, "\pi_0"] & \\
E^{(p)} & & E \arrow[ll, "\pi"]
\end{tikzcd}$$

Figure 5.2: Diagram illustrating a simple example of the attack

Let $T = [\alpha]\phi(P)$ and $S = [\alpha^{-1}]\phi(Q)$ be the images revealed by FESTA, then it can be shown that the following holds:

$$\psi \begin{pmatrix} S \\ T \end{pmatrix} = d \cdot A \cdot M_{\pi_0}^{-1} \cdot A^{-1} \cdot \pi \begin{pmatrix} S \\ T \end{pmatrix}, \tag{5.2}$$

where $A \in X \subseteq GL_2(\mathbb{Z}/2^b\mathbb{Z})$ is the typical FESTA scaling diagonal matrix having α and $\alpha^{-1} \mod 2^b$ as entries and M_{π_0} is the matrix representing the map π_0, that is:

$$\pi_0 \begin{pmatrix} P \\ Q \end{pmatrix} = M_{\pi_0} \begin{pmatrix} P \\ Q \end{pmatrix}.$$

The presence of the matrix A is the main difference compared to what was previously described and presents itself with the problem that in general $A \cdot M_{\pi_0}^{-1} \cdot A^{-1} \neq M_{\pi_0}^{-1}$, unless $M_{\pi_0}^{-1}$ itself is a diagonal matrix. This means that in order for the attack to be effective, we need P and Q to be eigenvectors of π_0.

Generalization

The intuitive attack can be generalized in an analogous way as it was done for M-SIDH in Section 4.1.2, with the main difference that now the scaling matrices are sampled from the set X of all diagonal matrices and therefore X is its own centralizer[28]. However, for the sake of clarity, let us repeat the setting.

Recall that we want to retrieve a secret isogeny $\phi : E_0 \to E$ of degree d (that in the FESTA setting is ϕ_A of degree d_A), given the bases $\{P, Q\} \subseteq E_0[2^b]$ and $\{S, T\} \subseteq E[2^b]$ such that $S = [\alpha]\phi(P)$ and $T = [\alpha^{-1}]\phi(Q)$. Let us express these last relations through the following notation:

$$\begin{pmatrix} S \\ T \end{pmatrix} = A \cdot \phi \begin{pmatrix} P \\ Q \end{pmatrix}$$

where A is a matrix sampled uniformly at random from the public set $X \subseteq \mathrm{GL}_2(\mathbb{Z}/2^b\mathbb{Z})$. In this case, X is the set of all diagonal matrices and A has elements equal to α and $\alpha^{-1} \mod 2^b$. Moreover, we take into account also the two following additional auxiliary isogenies:

1. $\sigma_0 : E_0 \to E_0'$ such that $\deg \sigma_0 = s$. We assume that its push-forward σ under ϕ is known, where

$$\sigma := \phi_* \sigma_0 : E \to E'.$$

   In other words, we have that $\ker(\sigma) = \phi(\ker(\sigma_0))$ is known.

2. $\omega : E_0 \to E_0'$, such that $\deg \omega = w$.

We must assume that $2^b, d, s, w$ are all pairwise coprime and that $p \nmid dw$. The overall configuration is depicted in Figure 5.3. As already noticed previously, the original "lollipop attack" presented in [Pet17] is generalized to arbitrary sets X and to an arbitrary choice of σ_0, that needs not to be the identity map over E_0, providing the advantage of possibly reducing the degree of the isogeny ϕ and therefore also improving on the bound required for applying the SIDH attacks.

$$\begin{tikzcd}
 & E_0' \arrow[dl, "\phi' = \sigma_{0*}\phi"'] & E_0 \arrow[l, bend right, "\sigma_0"'] \arrow[l, bend left, "\omega"] \arrow[dr, "\phi"] & \\
E' & & & E \arrow[lll, "\sigma = \phi_*\sigma_0"]
\end{tikzcd}$$

Figure 5.3: Diagram illustrating the generalized attack

The attack will provide a way to evaluate the wd^2-isogeny

$$\psi := \phi' \circ \omega \circ \hat{\phi} : E \to E'$$

[28] The centralizer of a subset S in a group G is the set of elements of G that commute with every element of S.

for any given point. In this case $\phi' = \sigma_{0*}\phi$ is such that $\phi' \circ \sigma_0 = \sigma \circ \phi$ by construction. Just like in the simplified version previously described, some non-trivial information about ϕ can be recovered regardless of whether ψ is cyclic or not, as seen in Remark 18. Whenever Lemma 2 applies, it is possible to compute $\psi(S)$ and $\psi(T)$.

Even if assuming the knowledge of the push-forward of σ_0 by the unknown isogeny ϕ may seem quite restrictive, there are two natural candidates for the map σ_0, being:

1. The identity map $\mathrm{id} : E_0 \to E_0$, having push-forward the identity map over E;
2. The Frobenius isogeny $\pi_0 : E_0 \to E_0^{(p)}$, whose push-forward is the Frobenius isogeny $\pi : E \to E^{(p)}$.

Moreover, it is possible to obtain other examples by composing one of the two options above with an isogeny of small degree. In this scenario, the push-forward of σ_0 by the unknown isogeny ϕ can be guessed with a reasonable success probability, as seen in Remark 18.

Application to FESTA

In order to apply the attack, we will need the centralizer of X to be sufficiently large. This condition is equivalent to requiring that P, Q are eigenvectors of $\hat{\sigma}_0 \circ \omega$ acting on $E_0[2^b]$. However, due to the following remark, at least one of the basis points needs to be an eigenvector of the endomorphism $\hat{\sigma}_0 \circ \omega$, where σ_0 is either the identity or the Frobenius endomorphism and ω is an endomorphism of small degree.

Remark 25. *As stated in [CV23, Remark 5], the condition on the centralizer can be relaxed at the expense of a stronger condition concerning 2^b. Namely, it is possible to evaluate $\phi(S)$ if P is an eigenvector of $\hat{\sigma}_0 \circ \omega$, even if Q is not an eigenvector. Indeed, the desired evaluation oracle can be still obtained if $2^b > d^2 w$. Analogously the remark holds true if Q is an eigenvector, but P not necessarily is.*

From now on, let B denote the order of the torsion points, i.e., $B = 2^b$ in the case of FESTA[29]. We will focus on recovering $\psi = \phi' \circ \omega \circ \hat{\phi}$. If we know the images of a full basis, we will require for $B > d\sqrt{w}$ to have a successful attack. If we know only the image of a single point, we will require for $B > d^2 w$, due to Remark 25.

Within the present section, we see how many different endomorphisms ω and how many different eigenspaces exist for the supersingular elliptic curve $E_0 : y^2 = x^3 + 6x^2 + x$ over $\mathbb{F}_p$ used in FESTA's implementation. Firstly, note that E_0 is 2-isogenous to the curve $E_1 : y^2 = x^3 + x$ via an isogeny θ with $\ker\theta = \langle(0,0)\rangle$. Since E_1 has well known endomorphism ring, we can obtain the following $\mathbb{Z}$-module basis of $\mathrm{End}(E_0)$:

$$\left\{\mathrm{id}, \frac{\pi_0 - [1]}{2}, \mathbf{i} - \mathbf{i}\pi_0, \frac{\mathbf{i} + \mathbf{i}\pi_0}{4}\right\},$$

where $\mathbf{i} = [\sqrt{-1}]$ and π_0 is the Frobenius endomorphism. Since $\mathbf{i}$ is an endomorphism over E_1 and not over E_0, we will consider the subring of $\mathrm{End}(E_0)$ generated by

$$\{\mathrm{id}, \pi_0, 2\mathbf{i}, 2\mathbf{i}\pi_0\},$$

where $2\mathbf{i} = \hat{\theta} \circ \mathbf{i} \circ \theta$ is an endomorphism over E_0. Given this setup, we are limited to choosing ω of the form $\omega = a + b2\mathbf{i}$ for some $a, b \in \mathbb{Z}$ of degree $a^2 + 4b^2$. Indeed, our endomorphism ω must avoid having π_0 as a component since $\deg(\pi_0) = p$ is very large and $w = \deg(\omega)$ is required to be as small as possible. Let us discuss what is required to have weak eigenspaces in the two main cases of $\sigma_0 = \mathrm{id}$ and $\sigma_0 = \pi_0$, as presented in [CV23, §5].

[29] Note that this is an abuse of notation, as B previously represented the scaling matrix. However, this should not cause confusion in the following discussion, as the matrix B will not be mentioned until the end of the section. When the notation will change, it will always be made clear (and comprehensible from the context).

Case $\boldsymbol{\sigma_0} = \textbf{id}$: In this case, we consider the endomorphism over E_0 that is $\hat{\sigma}_0 \circ \omega = \omega$. We have that different choices of ω do not result in distinct eigenspaces. Indeed, it is clear that if P is an eigenvector of ω with eigenvalue μ and $\gcd(b, B) = 1$, then P is also an eigenvector of $2\mathbf{i}$ with eigenvalue $(\mu - a)/b \mod B$. This case is not particularly satisfying since only the eigenspaces of $2\mathbf{i}$ are known to be weak.

Case $\boldsymbol{\sigma_0 = \pi_0}$: In this case, we consider the endomorphism over E_0 that is $\hat{\sigma}_0 \circ \omega = \hat{\pi}_0 \circ \omega = \hat{\pi}_0 \circ (a + 2b\mathbf{i})$ and want to analyze its eigenspaces. Due to the fact that $\pi_0^2 = [-p]$ over E_0, we have $\hat{\pi}_0 = -\pi_0$, it is suficient to analyze the eigenspaces of $\pi_0 \circ (a + 2b\mathbf{i})$. Let us present the attack for B odd and discuss later the case of $B = 2^b$ as it is in FESTA.

Case $\boldsymbol{B = \ell^n}$ with $\boldsymbol{\ell}$ odd prime: Let $\{U, V\}$ be a basis of eigenvectors of π_0 over $E_0[B]$ such that $\pi_0(U) = U$ and $\pi_0(V) = -V$. Let $P \in E_0[B]$ be an eigenvector of $\pi_0 \circ (a + 2b\mathbf{i})$ of order exactly B. Using the basis $\{U, V\}$, we have $P = cU + dV$ with $c, d \in \mathbb{Z}/B\mathbb{Z}$ and at least one of c, d is invertible in $\mathbb{Z}/B\mathbb{Z}$. Without loss of generality we can assume that this is c. Thus, after scaling by $c^{-1} \mod B$, we have $P = U + eV$ with $e \in \mathbb{Z}/B\mathbb{Z}$. Note that we obtain different eigenvectors when considering different values for $e \in \mathbb{Z}/B\mathbb{Z}$. Let the eigenvalue corresponding to P be μ. Then, using the fact that we must have $V = 2\mathbf{i}(U)$ and $2\mathbf{i}(V) = -4U$ in order to get $\pi_0(U) = U$ and $\pi_0(V) = -V$[30], we have

$$\pi_0 \circ (a + 2b\mathbf{i})(P) \equiv (a - 4be)U + (-ae - b)V \equiv \mu(U + eV) \mod B.$$

This is equivalent to[31]

$$4be^2 - 2ae - b \equiv 0 \mod B. \tag{5.3}$$

For every choice of e and b, we obtain a different quadratic equation for the variable e and discriminant $\Delta = 4a^2 + 16b^2$. Equation (5.3) will have two different solutions for e when $\frac{\Delta}{\ell} = 1$ and they will be given by:

$$e_\pm = \frac{a \pm \sqrt{a^2 + 4b^2}}{4b} = \frac{(a/2b) \pm \sqrt{(a/2b)^2 + 1}}{2} \mod B.$$

Note that the two different solutions for e will produce two different eigenspaces, that come in pairs $\{e_-, e_+\}$, as one fully determines the other and vice versa. We now distinguish whether we apply the attack in the case we have the images of a full basis or whether we have the image of only a single point of the basis.

1. Assume we have the images of a full basis. We have $w = \deg(\omega) = a^2 + 4b^2 < (B/d)^2$. The number of total pairs of weak eigenspaces can be computed as the number of different values of e that solve Equation (5.3) where a, b vary inside the ellipse $x^2 + 4y^2 = (B/d)^2$. The authors in [CV23, §5.2] state that in this case the proportion of weak eigenspaces for FESTA in this scenario is $O\left(\min\left\{\frac{1}{d^2}, \frac{1}{B}\right\}\right)$.
2. Assume we have the image of only a single point. An analogous analysis can be done. One would obtain that the proportion of weak eigenspaces for FESTA in this scenario is expected to be $O\left(\frac{1}{d^2}\right)$.

Case $\boldsymbol{B = 2^b}$ with $\boldsymbol{b > 3}$: With the same exact high-level intuition, let us discuss the differences between this case and the previous one. Firstly, we can only select U of order $B/2$, since $E[2]$ is already rational over $\mathbb{F}_p$ in FESTA. Moreover, we have that if $V = 2\mathbf{i}(U)$, then V only has

[30] We want the following to be true: $\pi_0(U) = U$ and $\pi_0(V) = -V$. In particular, suppose that $\pi_0(U) = U$ without loss of generality. Now notice that

$$\pi_0 \circ 2\mathbf{i}(U) = -2\mathbf{i} \circ \pi_0(U) = -2\mathbf{i}(U),$$

where we have used that π_0 and $2\mathbf{i}$ "anti-commute". From this we see that $V = 2\mathbf{i}(U)$ satisfies the desired property for V, i.e. $\pi_0(V) = -V$.

[31] The last passage tells us that we require

$$\begin{cases} a - 4be & \equiv \mu \mod B \\ -ae - b & \equiv e\mu \mod B \end{cases} \implies -ae - b \equiv e(a - 4be) \mod B \implies 4be^2 - 2ae - b \equiv 0 \mod B.$$

order $B' = B/8$. Therefore, we consider the basis $\{U' = 4U, V\}$ for $E[B']$, where we have that $2\mathbf{i}(V) = -U'$. Let $P = U' + eV \in E[B']$ with eigenvalue μ. This gives us

$$\pi_0 \circ (a + 2b\mathbf{i})(P) \equiv (a - be)U' + (-ae - 4b)V \equiv \mu(U' + eV) \mod B'.$$

This is equivalent to[32]

$$be^2 - 2ae - 4b \equiv 0 \mod B'. \tag{5.4}$$

If $b \not\equiv 0 \mod 2$, it is easy to verify that Equation (5.4) has no solutions. If $b \equiv 0 \mod 2$, then we let $b' = b/2$ and obtain the equivalent equation:

$$b'e^2 - ae - 4b' \equiv 0 \mod B'/2. \tag{5.5}$$

If $b' \equiv a \mod 2$, it is easy to verify that Equation (5.5) has 2 solutions modulo $B'/2$ and no solutions otherwise. The rest of the analysis stays the same as the case of odd B and therefore we have that the proportion of weak eigenspaces for FESTA is $O\left(\min\left\{\frac{1}{d^2}, \frac{1}{B}\right\}\right)$ if we have the images of the full basis and $O\left(\frac{1}{d^2}\right)$ in the case where we only have a single image point.

Overstretched FESTA

Let us present an instantiation of FESTA parameters that always allow to construct an endomorphism ω such that P and Q become eigenvectors. Such instantiation will involve a choice of parameters that are unbalanced and not natural for FESTA, leading to a theoretically interesting study that is far from practical.

Let us consider a general elliptic curve E_0 over $\mathbb{F}_{p^2}$ with unknown endomorphism ring and a general basis $\{P, Q\}$ of $E_0[B]$. Using lattice reduction [LLL82] we can find a $\mathbb{Z}$-basis $\{\mathrm{id}, \omega_1, \omega_2, \omega_3\} \subset \mathrm{End}(E_0)$ where $\deg(\omega_i) \approx p^{2/3}$ for all i, as described in [LB20, Prop. B.5]. Let us express the matrix representing ω_i as M_i. If we can find scalars $\lambda_i \in \mathbb{Z}$ such that

$$\lambda_1 M_1 + \lambda_2 M_2 + \lambda_3 M_3$$

is diagonal (and non-scalar), then P and Q would be eigenvectors of π_0 (since they would be eigenvectors of any endomorphism in $\mathrm{End}(E_0)$). The authors in [CV23, §5.4] state that if $B \geq pd^3$, then we are able to find an endomorphism ω of degree w of which P and Q are eigenvectors and such that $B > d\sqrt{w}$, as required to apply the attack. However, the fact that $B \geq pd^3$ implies that the B-torsion cannot be $\mathbb{F}_{p^2}$-rational, unlike in FESTA. Hence, this attacks only applies to an overstretched case and does not directly apply to concrete instantiations of FESTA.

5.3.2 Backdoors for FESTA

Introducing a backdoor in FESTA can be done similarly to what was done in the case of M-SIDH in Section 4.1.3. Namely, an attacker will generate system parameters E_0, P_B, Q_B that are obtained as the images of one of the weak instances previously discussed through an isogeny ε of low degree e. Let E_w, P_w, Q_w be a weak instance of FESTA such that $E_0 = \varepsilon(E_w), P_B = \varepsilon(P_w), Q_B = \varepsilon(Q_w)$. If $B^2 > e^2d^2w$, the attack will attempt to recover the isogeny $\varepsilon \circ \phi$ using the SIDH attacks. In the case where the weak basis is optimal, i.e. P, Q are eigenvectors of Frobenius, we have $w = 1$. Thus, the backdoor can tolerate isogenies ε up to degree B/d, that is very large in FESTA. If the endomorphism ring of E_0 is known, one can test if the basis is weak just as discussed in the M-SIDH case in Section 4.1.3. However, if the endomorphism ring of E_0 is unknown, then verifying whether FESTA is backdoored becomes almost impossible since the degree of ε can be too large.

The authors in [CV23, §5.3] propose two solutions for this problem. Firstly, since the proportion of weak bases is of order $O(1/d^2)$, that is very small, the basis $\{P_B, Q_B\}$ can be re-randomized as it

[32] The last passage tells us that we require

$$\begin{cases} a - be & \equiv \mu \mod B' \\ -ae - 4b & \equiv e\mu \mod B' \end{cases} \implies -ae - 4b \equiv e(a - be) \mod B' \implies be^2 - 2ae - 4b \equiv 0 \mod B'.$$

will result in a no longer weak instance with overwhelming probability. Secondly, the basis $\{P_B, Q_B\}$ can be deterministically generated using a hash function as described in [Zan+18], re-randomized and included as part of the public key.

Remark 26. *Similarly to what was mentioned in Remark 21, the same verifications have to be performed for the co-domain curve E due to the symmetry of FESTA. Indeed, one could attack the dual isogeny, where domain and codomain are swapped.*

5.4 The FESTA Public-key Encryption scheme

Let us see how to derive a Public-key Encryption (PKE) scheme from the FESTA trapdoor function. We will analyze the security in the Quantum Random Oracle Model (QROM) and in the standard model obtaining an IND-CCA secure PKE scheme in both scenarios.

5.4.1 IND-CCA encryption in the QROM

It has been shown in [Ebr22] that it is possible to obtain a IND-CCA PKE that is secure in the Quantum Random Oracle Model (QROM) using the OAEP transform, given an injective partial-domain trapdoor function. In order to apply the OAEP transform to the FESTA trapdoor function, the authors in [BMP23, §5.1] prove that the FESTA trapdoor function is indeed a partial-domain trapdoor function in the following theorem.

Theorem 7. *The function $f_{(E_A,R_A,S_A)}$ that has domain $\mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b$ defined in Algorith 1 is a quantum partial one-way function, assuming that Problem 7 and Problem 8 are hard.*

Proof. Let us prove a stronger statement: recovering any of the three inputs is as hard as inverting $f_{(E_A,R_A,S_A)}$ over the entire domain. This will imply the thesis. Given the isogeny ϕ_1, the matrix B^{33} can be obtained by computing the change-of-basis matrix between $\phi_1(P_b), \phi_1(Q_b)$ and R_1, S_1. The remaining input to recover, the isogeny ϕ_2, can be computed as the output of $\mathsf{TorAtk}(E_A, R_A, S_A, E_2, R_2^\star, S_2^\star, d_2)$, where $R_2^\star, S_2^\star$ are obtained as

$$\begin{pmatrix} R_2^\star \\ S_2^\star \end{pmatrix} = B^{-1} \cdot \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}.$$

Since this proves that recovering any of the three inputs is equivalent to recovering the whole input, we have the thesis. □

Therefore, this shows that the OAEP transform can be applied to obtain the following PKE scheme. The PKE parameters are the prime p, the curve E_0, the values d_1, d_2, d_A, b and a description of the set $\mathcal{M}_b$. We rely on two random oracles $G : \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b \to \mathbb{Z}/d_1\mathbb{Z}$ and $H : \mathbb{Z}/d_1\mathbb{Z} \to \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b$. The KeyGen algorithm is similar to that in the trapdoor function and produces the public key (E_A, R_A, S_A).

To encrypt, first evaluate the oracle G at a random sample (r, R) and use this, together with the message m to determine the kernel of the isogeny ϕ_1. Deterministically derive the isogeny ϕ_2 and the matrix B from the randomness (r, R) and the kernel of ϕ_1 through the oracle H. Algorithm 3 shows how this provides the ciphertext though the encryption algorithm. Note that, as previously discussed in Remark 23, the kernel of an isogeny can be represented with a single integer modulo the degree of the isogeny. This is what is done in the algorithm, where s represents $\ker(\phi_1)$ and t represents $\ker(\phi_2)$.

To decrypt, we use the trapdoor information to recover the isogenies ϕ_1, ϕ_2 and the matrix B. This information is then used to extract the message m from the ciphertext, as described in Algorithm 4.

Remark 27. *As the authors suggest in [BMP23, Remark 12], this approach does not leverage Fujisaki-Okamoto transform [FO97], contrarily to what is commonly done to obtain IND-CCA security for most post-quantum encryption protocols. The Fujisaki-Okamoto transform re-evaluates the encryption procedure during decryption and this has enabled several side-channel attacks in the past [Uen+22].*

[33] Note that from now on, except for Section 5.5.2, B will represent the matrix $B \in \mathcal{M}_b$ again and not the order of the torsion points.

Algorithm 3 FESTA.$\mathsf{Enc}(\mathsf{pk}, m)$

```
Input: The public key $\mathsf{pk} = (E_A, R_A, S_A)$ and the message $m$ to be encrypted.
Output: The ciphertext $(E_1, R_1, S_1, E_2, R_2, S_2)$.
Sample $r \leftarrow_\$ \mathbb{Z}/d_2\mathbb{Z}$ and $R \leftarrow_\$ \mathcal{M}_b$.
Write $m' = m \mid\mid 0^k \mod d_1$ and compute $s = m' + G(r, R)$.
Write $(x, X) = H(s)$ and compute $t = x + r$, $T = XR$.
Compute $\mathsf{ct} = f_{(E_A,R_A,S_A)}(s, t, T)$.        ▷ Using Algorithm 1
return $\mathsf{ct} = (E_1, R_1, S_1, E_2, R_2, S_2)$
```

Algorithm 4 FESTA.$\mathsf{Dec}(\mathsf{pk}, m)$

```
Input: The secret key $\mathsf{sk} = (A, \phi_A)$ and the ciphertext $\mathsf{ct} = (E_1, R_1, S_1, E_2, R_2, S_2)$ to be decrypted.
Output: The decrypted message $m$ or $\perp$ on failure.
Compute $(s, t, T) = f^{-1}_{(E_A,R_A,S_A)}(\mathsf{sk}, \mathsf{ct})$.        ▷ Using Algorithm 2
Write $(x, X) = H(s)$ and compute $r = t - x$, $R = X^{-1}T$.
Compute $m' = s - G(r, R)$ and write $M \mid\mid m_k = m'$, where $|m_k| = k$.
if $m_k = 0^k$ then
    return $m$
else
    return $\perp$
```

Using the OAEP transform for FESTA avoids these problems entirely as decryption does not need to run the encryption algorithm, reducing latency and protecting more efficiently against side-channel analysis.

5.4.2 IND-CCA encryption in the standard model

At the time of publication, FESTA was the only secure trapdoor function based on non-group-action isogenies, while in the literature trapdoor functions based on group actions were more known [Ala+20]. This provides a large advantage to FESTA as it allows for the application of the techniques described in [HKW20] to obtain the first PKE scheme based on non-group-action isogenies that is IND-CCA secure in the standard model.

As stated in [BMP23, §5.2], the construction relies on the following two building blocks: a randomness-recoverable IND-CPA PKE and a tagged set commitment protocol.

1. Randomness-recoverable IND-CPA PKE (RR-PKE) is a special type of probabilistic encryption scheme that allows to extract from the ciphertext both the plaintext and the randomness used in the encryption algorithm, with the help of the private key [LW10; DA13]. A RR-PKE can be built from an *almost-all-keys injective trapdoor function*, as described in [HKW20, §2]. Roughly, this requires that for almost all pairs of secret-public keys and for all inputs, the trapdoor inversion function returns precisely the same input that the trapdoor function was evaluated at. As discussed in Remark 23, in the case where we have a curve with j-invariant in $\{0, 1728\}$, the FESTA trapdoor function may not be injective since the curve has additional automorphisms. Thus, the required property does not hold for a large class of public keys for which a specific input produces as output a curve with j-invariant in $\{0, 1728\}$. However, as the authors in [BMP23, §5.2] notice, in FESTA we can check if an input can lead to issues by evaluating the trapdoor function and checking the j-invariant of the output. This is due to the fact that the correctness of the inversion depends only on public information. Therefore, the *almost-all-keys injective trapdoor function* property can be satisfied by redefining the function input to exclude the inputs that are identified as potentially problematic.

2. A tagged set commitment protocol is a commitment scheme for which the committer and the receiver have the common input of the security parameter λ, and a "tag" (i.e. an identifier) of $l(\lambda)$-bits [KLV17, Definition 3]. The construction of a tagged set commitment protocol requires

a secure one-time signature, that can be constructed from any one-way function [Lam79], and, of course, a trapdoor function.

Therefore, the FESTA trapdoor function can be used to construct all the elements needed to obtain a PKE scheme that is IND-CCA secure in the standard model.

5.5 Concrete instantiation and parameters selection

Within the present section we address some details that were not discussed in previous sections, but are crucial for instantiating the best parameters so that one can recover the d_i-isogenies as fast as possible applying Corollary 2. Note that Corollary 2 is a specification of Theorem 5 to our case.

Corollary 2 (Application of Theorem 5)**.** *Let E_0, E_1, E_2 and E_3 be four elliptic curves defined over $\mathbb{F}_{p^2}$ such that there exist two isogenies $\phi_{N_1} : E_0 \to E_1$ and $\phi_{N_2} : E_0 \to E_2$. Let $\deg \phi_{N_1} = N_1$ and $\deg \phi_{N_2} = N_2$ be coprime integers. The isogeny Φ represented by the matrix*

$$\begin{pmatrix} \hat{\phi}_{N_1} & -\hat{\phi}_{N_2} \\ g_{N_2} & \hat{g}_{N_1} \end{pmatrix},$$

is a $(N_1 + N_2, N_1 + N_2)$-isogeny, where g_{N_i} is a N_i-isogeny such that $\phi_{N_2} \circ \hat{\phi}_{N_1} = g_{N_1} \circ g_{N_2}$. We have the commutative diagram depicted in Figure 5.4.

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_{N_2}"] \arrow[d, "\phi_{N_1}"'] & E_2 \\
E_1 \arrow[r, "g_{N_2}"'] & E_3 \arrow[u, "g_{N_1}"']
\end{tikzcd}$$

Figure 5.4: Commutative diagram representing the situation described in Corollary 2.

5.5.1 Recovering an isogeny from torsion point images

Let us describe how the trapdoor function proposed in Section 5.2 can be inverted. Recall that $\phi_1 : E_0 \to E_1$, $\phi_A : E_0 \to E_A$ and $\phi_2 : E_A \to E_2$ are three isogenies between supersingular curves having odd coprime degrees. Let ϕ_A be the composition of two isogenies $\phi_{A,1} : E_0 \to \tilde{E}_A$ and $\phi_{A,2} : \tilde{E}_A \to E_A$ of coprime degrees $d_{A,1}$ and $d_{A,2}$, respectively. This means that we have the configuration depicted in Figure 5.5.

$$\begin{tikzcd}
 & E_0 \arrow[r, "\phi_{A,1}"] \arrow[rr, bend left, "\phi_A"] \arrow[dl, "\phi_1"'] & \tilde{E}_A \arrow[r, "\phi_{A,2}"] \arrow[dll, dashed, "\phi_{N_1}"] \arrow[drr, dashed, "\phi_{N_2}"'] & E_A \arrow[dr, "\phi_2"] & \\
E_1 \arrow[loop below, "{[m_1]}"] & & & & E_2 \arrow[loop below, "{[m_2]}"]
\end{tikzcd}$$

Figure 5.5: Configuration required to recover the isogeny, adapted from [BMP23, §6.1].

Suppose we have found values of $m_1, m_2, b \in \mathbb{Z}_{>0}$ such that

$$m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b, \tag{5.6}$$

for some odd m_i coprime to d_1, d_2 and d_A. This will be a key relation in the context of the present section and we will discuss how the integers m_1,m_2 and b can be found in Section 5.5.2. Equation (5.6) stems from Kani's criterion in higher dimensions and, if satisfied, allows us to find parameters more efficiently. Moreover, since Equation (5.6) implies that the order of the points is smooth, this will allow to recover the matrix B by solving a simple instances of the discrete logarithm problem. The relevance of this relation will be clearer as we proceed throughout the section.

Given a security parameter λ, let

$$\mathsf{params}_\lambda = (m_1, m_2, b, p, d_1, d_{A,1}, d_{A,2}, d_2, E_0)$$

be a parameter set, where E_0 is a supersingular curve whose j-invariant $\notin \{0, 1728\}$ and $E_0(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p+1)\mathbb{Z})^2$, the prime p is of the form $f 2^b d_1 \cdot \mathsf{sf}(d_{A,1} d_{A,2}) \cdot d_2 - 1$ for some small cofactor $f > 0$ and $m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b$.

Provided as input the system's parameter set params_λ, Algorithm 5 taken from [BMP23, Algorithm 5] provides a precise description of the evaluation of the trapdoor function. The kernels $\langle K_1 \rangle \subset E_0[d_1]$ and $\langle K_2 \rangle \subset E_0[d_2]$ are cyclic subgroups chosen such that they are generated by an element of the form $P + [x]Q$, where $\{P, Q\}$ is a basis the torsion subgroup. As discussed in Remark 23 the kernels can be represented by elements $s_1 \in [0, d_1 - 1]$ and $s_2 \in [0, d_2 - 1]$, given two bases $\{P_{d_1}, Q_{d_1}\}$ of $E_0[d_1]$ and $(P^A_{d_2}, Q^A_{d_2})$ of $E_A[d_2]$.

Algorithm 5 $f_{(E_A, R_A, S_A)}(s_1, s_2, B)$

Input: Two integers $s_1 \in [0, d_1 - 1]$ and $s_2 \in [0, d_2 - 1]$, and the matrix $B \in \mathcal{M}_b$.
Output: The array $(E_1, R_1, S_1, E_2, R_2, S_2)$.

```
1: Compute the bases $(P_{d_1}, Q_{d_1}) \leftarrow \mathsf{TorGen}(E_0, d_1)$ and $(P_{d_2}, Q_{d_2}) \leftarrow \mathsf{TorGen}(E_A, d_2)$.
2: Compute the $d_1$-isogeny $\phi_1 : E_0 \to E_1$ having kernel $\langle P_{d_1} + [s_1]Q_{d_1} \rangle$.
3: Compute the $d_2$-isogeny $\phi_2 : E_A \to E_2$ having kernel $\langle P_{d_2} + [s_2]Q_{d_2} \rangle$.
4: Acting with scalar multiplication, compute
       $\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = B \cdot \begin{pmatrix} \phi_1(P_b) \\ \phi_2(Q_b) \end{pmatrix}, \qquad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = B \cdot \begin{pmatrix} \phi_2(R_A) \\ \phi_2(S_A) \end{pmatrix}.$
5: return $(E_1, R_1, S_1, E_2, R_2, S_2)$
```

To invert the trapdoor function $f_{(E_A, R_A, S_A)}$ we need to specialize Corollary 2 to our case and consider $\phi_{N_1} = [m_1] \circ \phi_1 \circ \hat{\phi}_{A,1}$ and $\phi_{N_2} = [m_2] \circ \phi_2 \circ \phi_{A,2}$ so that $N_1 = \deg \phi_{N_1} = m_1^2 d_{A,1} d_1$, $N_2 = \deg \phi_{N_2} = m_2^2 d_{A,2} d_2$ and $N_1 + N_2 = 2^b$. Given the array $(E_1, R_1, S_1, E_2, R_2, S_2)$, we compute the points

$$\begin{pmatrix} R_2' \\ S_2' \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}.$$

and note that the isogeny Φ in Corollary 2 has matrix form

$$\begin{pmatrix} [m_1] \circ \phi_{A,1} \circ \hat{\phi}_1 & -[m_2] \circ \hat{\phi}_{A,2} \circ \hat{\phi}_2 \\ [m_2] \circ g_{d_2 d_{A,2}} & [m_1] \circ \hat{g}_{d_{A,1} d_1} \end{pmatrix},$$

where $\phi_2 \circ \phi_A \circ \hat{\phi}_1 = g_{d_{A,1} d_1} \circ g_{d_2 d_{A,2}}$ such that $\deg g_{d_{A,1} d_1} = d_{A,1} d_1$ and $\deg g_{d_2 d_{A,2}} = d_2 d_{A,2}$.

Now let us first see how to recover the values s_1 and s_2 representing the kernels of the isogenies ϕ_1 and ϕ_2 respectively and then describe how it is possible to retrieve also the matrix B by solving a discrete logarithm problem. The inversion procedure is summarized in Algorithm 7 taken from [BMP23, Algorithm 7].

Recover s_1 and s_2: Let $\{P^1_{d_1}, Q^1_{d_1}\} \leftarrow \mathsf{TorGen}(E_1, d_1)$ and $\{P^2_{d_2}, Q^2_{d_2}\} \leftarrow \mathsf{TorGen}(E_2, d_2)$. Then we have that

$$\begin{pmatrix} L \\ - \end{pmatrix} := \Phi \begin{pmatrix} P^1_{d_1} + R_1 \\ P^2_{d_2} \end{pmatrix} = \begin{pmatrix} [m_1]\phi_{A,1} \circ \hat{\phi}_1(P^1_{d_1} + R_1) - [m_2]\hat{\phi}_{A,2} \circ \hat{\phi}_2(P^2_{d_2}) \\ - \end{pmatrix},$$

where the symbol "–" represents elements that do not matter to us as they carry exclusively information concerning g_{N_1} and g_{N_2}. Unpacking L and leveraging the homomorphic properties of isogenies together with the fact that the subscript of the points represents their order, we obtain

$$\begin{aligned}
[2^b d_2]L &= [2^b d_2 m_1]\phi_{A,1} \circ \hat{\phi}_1(P^1_{d_1}),\\
-[2^b d_1]L &= [2^b d_1 m_2]\hat{\phi}_{A,2} \circ \hat{\phi}_2(P^2_{d_2}),\\
[d_1 d_2]L &= [d_1 d_2 m_1]\phi_{A,1} \circ \hat{\phi}_1(R_1).
\end{aligned}$$

Analogously, we evaluate $\Phi(Q^1_{d_1} + S_1, Q^2_{d_2})$ to obtain values for

$$[2^b d_2 m_1]\phi_{A,1} \circ \hat{\phi}_1(Q^1_{d_1}),\ [2^b d_1 m_2]\hat{\phi}_{A,2} \circ \hat{\phi}_2(Q^2_{d_2}) \text{ and } [d_1 d_2 m_1]\phi_{A,1} \circ \hat{\phi}_1(S_1).$$

From this information, using the knowledge of $\phi_{A,i}$, we can obtain the values of $\hat{\phi}_1(P^1_{d_1})$, $\hat{\phi}_1(Q^1_{d_1})$, $\hat{\phi}_2(P^2_{d_2})$ and $\hat{\phi}_2(Q^2_{d_2})$. We can now compute s^1_1 and s^2_2, with a slight abuse of notation, such that $\ker(\hat{\phi}_1) = \langle P^1_{d_1} + [s^1_1]Q^1_{d_1}\rangle$ and $\ker(\hat{\phi}_2) = \langle P^2_{d_2} + [s^2_2]Q^2_{d_2}\rangle$. This is possible because we know that, e.g., $P^1_{d_1} + [s^1_1]Q^1_{d_1} \in \ker(\hat{\phi}_1)$. Leveraging the homomorphic properties of the isogeny $\hat{\phi}_1$, we have that

$$\hat{\phi}_1(P^1_{d_1} + [s^1_1]Q^1_{d_1}) = \hat{\phi}_1(P^1_{d_1}) + [s^1_1]\hat{\phi}_1(Q^1_{d_1}) = 0$$

and therefore one can solve for $[s^1_1]$ and, analogously for $[s^2_2]$. At this point it is clearly possible to compute $\ker(\phi_1)$ and $\ker(\phi_2)$ from the knowledge of the kernels of the dual isogenies, thus successfully completing the inversion of the trapdoor function as we compute s_1 and s_2 such that $\ker(\phi_1) = \langle P_{d_1} + [s_1]Q_{d_1}\rangle$ and $\ker(\phi_2) = \langle P^A_{d_2} + [s_2]Q^A_{d_2}\rangle$. Note that this case is slightly more complicated than the more common case where d_1 is a prime power. Algorithm 6 taken from [BMP23, Algorithm 6] describes the procedure to follow.

Algorithm 6 $\mathsf{ComputeCanonicalKernel}(\hat{\phi}(P'), \hat{\phi}(Q'), d)$

```
Input: $\hat{\phi}(P')$ and $\hat{\phi}(Q')$, where $\phi : E \to E'$ is a $d$ isogeny and $\langle P', Q'\rangle = E'[d]$.
Output: $s \in [0, d-1]$ such that $\ker(\phi) = \langle P + [s]Q\rangle$, where $\{P, Q\} \leftarrow \mathsf{TorGen}(E, d)$[34].
Compute the canonical basis $\{P, Q\} \leftarrow \mathsf{TorGen}(E, d)$, and let $d = \prod_{i=1}^n \ell_i^{e_i}$.
Compute $a_1, b_1 \in [0, d-1]$ such that $\hat{\phi}(P') = [a_1]P + [b_1]Q$.
Compute $a_2, b_2 \in [0, d-1]$ such that $\hat{\phi}(Q') = [a_2]P + [b_1]Q$.
for $i = 1, \ldots, n$ do
    if $a_1 \equiv 0 \mod \ell_i$ then
        Impose $t_1 \equiv 0 \mod \ell_i^{e_i}$ and $t_2 = a_2^{-1} \mod \ell_i^{e_i}$.
    else
        Impose $t_1 \equiv a_1^{-1} \mod \ell_i^{e_i}$ and $t_2 = 0 \mod \ell_i^{e_i}$.
Lift $t_1$ and $t_2$ in $\mathbb{Z}/d\mathbb{Z}$ and define $s \leftarrow t_1 b_1 + t_2 b_2$.
return s
```

Recover B: Firstly note that

$$\begin{pmatrix}\hat{\phi}_1(R_1)\\ \hat{\phi}_1(S_1)\end{pmatrix} =: \hat{\phi}_1\begin{pmatrix}R_1\\ S_1\end{pmatrix} = \hat{\phi}_1\left(B \cdot \phi_1\begin{pmatrix}P_b\\ Q_b\end{pmatrix}\right) = B \cdot \hat{\phi}_1 \circ \phi_1\begin{pmatrix}P_b\\ Q_b\end{pmatrix} = d_1 \cdot B \cdot \begin{pmatrix}P_b\\ Q_b\end{pmatrix}.$$

This means that the matrix B can be recovered by solving a discrete logarithm problem. Since the order of out points is a power of two, the discrete logarithm can be solved efficiently.

5.5.2 Parameters selection

Since the parameters of FESTA must satisfy Equation (5.6) so that the inversion of the trapdoor function can be done efficiently, let us see how we can generate solutions to such equation. Moreover, from Section 5.3, we also have additional requirements that the solutions should satisfy, that is:

Algorithm 7 $f^{-1}_{(E_A,R_A,S_A)}(E_1,R_1,S_1,E_2,R_2,S_2)$

```
Input: The output of the trapdoor function $y := (E_1,R_1,S_1,E_2,R_2,S_2)$, and $\mathsf{sk} = (A,\phi_{A,1},\phi_{A,2})$.
Output: The input of the trapdoor function $x := (s_1,s_2,B)$ such that $f_{(E_A,R_A,S_A)}(x) = y$.
Compute $\begin{pmatrix} R_2' \\ S_2' \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}$.
Define $\Phi$ to be the isogeny from Theorem 5
if the codomain of $\Phi$ does not split then
    return $\perp$
Let $\{P^1_{d_1}, Q^1_{d_1}\} \leftarrow \mathsf{TorGen}(E_1,d_1)$ and $\{P^2_{d_2}, Q^2_{d_2}\} \leftarrow \mathsf{TorGen}(E_2,d_2)$
Evaluate
    $\begin{pmatrix} L_1 \\ - \end{pmatrix} = \Phi\begin{pmatrix} P^1_{d_1} + R_1 \\ P^2_{d_2} \end{pmatrix}$ and $\begin{pmatrix} L_2 \\ - \end{pmatrix} = \Phi\begin{pmatrix} Q^1_{d_1} + S_1 \\ Q^2_{d_2} \end{pmatrix}$.
Unpack $L_1$ to obtain $\phi_{A,1} \circ \hat{\phi}_1(P^1_{d_1})$, $\phi_{A,1} \circ \hat{\phi}_1(R_1)$ and $\hat{\phi}_{A,2} \circ \hat{\phi}_2(P^2_{d_2})$.
Unpack $L_2$ to obtain $\phi_{A,1} \circ \hat{\phi}_1(Q^1_{d_1})$, $\phi_{A,1} \circ \hat{\phi}_1(S_1)$ and $\hat{\phi}_{A,2} \circ \hat{\phi}_2(Q^2_{d_2})$.
Set $s_1 \leftarrow \mathsf{ComputeCanonicalKernel}(\hat{\phi}_1(P^1_{d_1}), \hat{\phi}_1(Q^1_{d_1}), d_1)$    ▷ Using Algorithm 6
Set $s_2 \leftarrow \mathsf{ComputeCanonicalKernel}(\hat{\phi}_2(P^2_{d_2}), \hat{\phi}_2(Q^2_{d_2}), d_2)$    ▷ Using Algorithm 6
Compute the matrix $B \in \mathcal{M}_b$ such that
    $\begin{pmatrix} \hat{\phi}_1(R_1) \\ \hat{\phi}_1(S_1) \end{pmatrix} = d_1 \cdot B \cdot \begin{pmatrix} P_b \\ Q_b \end{pmatrix}$.
if $B \notin \mathcal{M}_b$ then
    return $\perp$
else
    return $(s_1,s_2,B)$
```

1. We want $m_1, d_1, m_2, d_2, d_{A,1}, d_{A,2}$ to be odd, so that the isogenies have degree coprime with the torsion points order;

2. We require that the isogeny degrees d_1, $d_A = d_{A,1}d_{A,2}$ and d_2 are pairwise coprime and sufficiently large so that we prevent meet-in-the-middle and torsion-guessing attacks. Therefore, we require $\log(d_1), \log(d_A), \log(d_2) \geq 2\lambda$.

As mentioned in [BMP23, §6.2], the number of solutions and the protocol's efficiency strongly rely on the smoothness of the degrees of the isogenies ϕ_1, ϕ_A, ϕ_2. From now on, let $\overline{B}$ be the smoothness bound [35] and let $c \in \mathbb{Z}_{>0}$ be such that $T := 2^c - 1$ is $\overline{B}$-smooth. To find appropriate parameters for FESTA, we start by looking for solutions $(x,y) \in \mathbb{Z} \times \mathbb{Z}$ with $\gcd(x,y) = 1$, that we call "primitive", for the equation

$$x^2 + y^2 T = 2^b. \tag{5.7}$$

It is possible to find solutions to Equation (5.7) through Cornacchia's algorithm [Cor08], by ranging the value b within a bounded interval. Given a primitive solution (x,y) for some even positive integer b, we have

$$y^2 T = (2^{b/2} - x)(2^{b/2} + x).$$

If we consider T_1 to be the $\overline{B}$-smooth part of $2^{b/2} - x$ and T_2 to be the $\overline{B}$-smooth part of $(2^{b/2} + x)$, then there must exist positive integers m_1, m_2 such that $m_1^2 T_1 = 2^{b/2} - x$ and $m^2 T_2 = 2^{b/2} + x$. This leads to focusing on the relation

$$m_1^2 T_1 + m_2^2 T_2 = 2^{b/2+1},$$

that resembles Equation (5.6). For this reason, the authors let d_i be the smoothest factor of T_i such that $d_i \sim 2^{2\lambda}$ and $d_{A,i}$ be the smoothest part of T_i/d_i such that $d_{A,1}d_{A,2} > 2^{2\lambda}$ for $i = 1, 2$.

[35] Let us remind the reader that in this case $B \in \mathcal{M}_b$ represents the smoothness bound and not the scaling matrix anymore.

Moreover, it is suggested to multiply m_i by $\sqrt{T_i/(d_i d_{A,i})}$ to have $d_1, d_{A,1}, d_{A,2}, d_2, m_1, m_2$ pairwise coprime. Note that we require T_i and T_1T_2 to be sufficiently large to guarantee security, i.e. $T_i > 2^{2\lambda}$ and $T_1T_2 > 2^{6\lambda}$. In conclusion, this way we are able to find solution to Equation (5.6) that guarantee the size requirements specified by FESTA.

Remark 28. *Even though this procedure performs an exhaustive search over different values of b and c in bounded intervals, this is experimentally very efficient and allows to generate parameters sets for any desired security level, as stated by the authors in [BMP23].*

Moreover, keeping in mind that we work with primes of the form $p = 2^b d_1 \cdot \mathsf{sf}(d_{A,1}d_{A,2}) \cdot d_2 f - 1$, one can increase the smoothness bound $\overline{B} = 2^b$ to find a smaller prime p and obtain a more compact scheme. This might be relevant if bandwidth is a particularly valuable resource. Indeed, if one is allowed to have a larger value for b, then its possible to adjust the other factors to get as close as possible to the minimum required length for p to achieve λ bits of security. However, this comes at the cost of a loss in efficiency as the degrees of the isogenies to compute will be larger, and possibly less smooth. Therefore, there is an inherent trade-off between efficiency and bandwidth.

Note that if the smoothness bound $\overline{B} = 2^b$ is decreased, then b is reduced and therefore the length of the $(2^b, 2^b)$-isogeny to compute in order to decrypt decreases as well. Hence, a speed-up in the decryption algorithm is achieved. However, this will come at the cost of a less compact scheme as the prime p will be larger than required.

Concrete parameters

For FESTA-128, the authors in [BMP23, §7.3] define the following parameter set:

$$\begin{aligned}
b &:= 632,\\
d_1 &:= (3^3 \cdot 19 \cdot 29 \cdot 37 \cdot 83 \cdot 139 \cdot 167 \cdot 251 \cdot 419 \cdot 421 \cdot 701 \cdot 839 \cdot 1009 \cdot 1259 \cdot 3061 \cdot 3779)^2,\\
d_2 &:= 7 \cdot (52 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 41 \cdot 43 \cdot 71 \cdot 89 \cdot 127 \cdot 211 \cdot 281 \cdot 503 \cdot 631 \cdot 2309 \cdot 2521 \cdot 2647 \cdot 2729)^2,\\
d_{A,1} &:= (59 \cdot 6299 \cdot 6719 \cdot 9181)^2,\\
d_{A,2} &:= (3023 \cdot 3359 \cdot 4409 \cdot 5039 \cdot 19531 \cdot 22679 \cdot 41161)^2,\\
m_1 &:= 1492184945093476592520242083925044182103921,\\
m_2 &:= 25617331336429939300166693069,\\
f &:= 107.
\end{aligned}$$

Note that d_1 and d_2 are 2^{12}-smooth, while $d_A = d_{A,1}d_{A,2}$ is 2^{16}-smooth. The resulting prime $p = 2^b d_1 \cdot \mathsf{sf}(d_{A,1}d_{A,2}) \cdot d_2 f - 1$ is 1292-bit long. The public key and the ciphertext are respectively 561 bytes and 1122 bytes long, while the secret key is pretty compact as it consists of 485 bytes. Higher levels of security can be achieved through the approach described above.

Finally, benchmarks of the proof-of-concept implementation of FESTA were executed on an Apple M1 PRO CPU clocked at 3.2 GHz using a single performance core. They revealed that, averaging over 100 executions, KeyGen, Enc and Dec run in 4.47, 3.09 and 9.14 seconds respectively.

5.6 Implementation and further optimizations

The authors provide a proof-of-concept implementation of FESTA in SageMath [Sag24] and decided to explicitly implement the algorithms for computing the isogenies between elliptic curves and abelian varieties, avoiding the generic SageMath implementation. The implementation of the FESTA PKE scheme can be found at:

https://github.com/FESTA-PKE/FESTA-SageMath.

Within the present section we explain some details and techniques employed.

5.6.1 Implementation details

Montgomery curve x-only isogenies

In order to compute isogenies, the authors use the more efficient x-only formulae between Montgomery curves [CH17; Ren18]. More specifically, $\sqrt{\text{élu}}$ [Ber+20] is used to evaluate isogenies of large prime degree and formulae based on Edwards curves [MR18] for computing the codomain curve. Moreover, since working with x-only isogenies leads to significant performance improvements, one would want to implement FESTA adopting this approach. However, the y-coordinate of image points must be recovered eventually in order to compute the chain of $(2,2)$-isogenies between elliptic products expressed by Φ. The following method, presented in [BMP23, §7.1] shows how to obtain the valid y-coordinate from the x-only point evaluation.

Consider two elliptic curves $E : y^2 = x^3 + Ax^2 + x$ and $E' : y^2 = x^3 + A'x^2 + x$ in Montgomery form connected by a d-isogeny $\phi : E \to E'$. Suppose we want to evaluate ϕ on the basis $\{P, Q\} \subset E[n]$, where $n \neq d$ is the order of the n-torsion subgroup of the curve E. Through the use of the x-only isogeny formulae, we obtain $x(\phi(P))$ and $x(\phi(Q))$. To recover the y-coordinates y_P and y_Q, we want to extract any square root of $x(\phi(P))^3 + A'x(\phi(P))^2 + x(\phi(P))$ and $x(\phi(Q))^3 + A'x(\phi(Q))^2 + x(\phi(Q))$, respectively. Thus, we can recover $\phi(P) = \pm(x(\phi(P)), y_P)$ and $\phi(Q) = \pm(x(\phi(Q)), y_Q)$ up to an overall sign.

The apparent limitation of this approach is that we cannot recover the correct sign for one point, but we can recover the tuple

$$\pm(\phi(P), \phi(Q)) = \pm\phi \begin{pmatrix} P \\ Q \end{pmatrix}.$$

by computing the following Weil pairing and checking whether

$$e_n^E(P, Q)^d \overset{?}{=} e_n^{E'}\ (\ \phi(P), \phi(Q)\ ) := e_n^{E'}\ (\ (x(\phi(P)), y_P), (x(\phi(Q)), y_Q)\ ).$$

The authors notice that this provides a tool to identify whether the values of y_P and y_Q are correct. If the equality holds, then they are indeed correct. Otherwise, we invert the sign of one of the two so that we have $\phi(Q) = (x(\phi(Q), -y_Q)$. Therefore, we are able to evaluate either ϕ or $-\phi$ on the torsion basis $\{P, Q\}$.

The key fact here is that in the FESTA framework this does not represent a problem for the trapdoor, since we use the canonical representation of the scaling matrices sampled from $\mathcal{M}_b$. Therefore, we are able to evaluate two isogenies using x-only formulae, with the additional cost of computing two square-roots and two Weil pairings.

Optimizations of the $(2,2)$-isogeny chain

In order to decrypt, i.e. to invert the FESTA trapdoor function, one must evaluate a chain of $(2,2)$-isogenies between elliptic products, since one of the SIDH attacks must be used. This is the most expensive step of the decryption algorithm, and the most expensive procedure out of all FESTA algorithms in general. The authors in [BMP23, §7.2] notice that the majority of the cost is due to the computations of the isogenies via the Richelot correspondence between Jacobians of genus-2 hyperelliptic curves [Smi+05, Chapter 8]. The two optimizations that the authors present are briefly described in what follows.

The first optimization adopted stems from the optimal strategies introduced in [DJP14, §4] to reduce the cost of long isogeny chains. The higher dimensional case has been studied more recently in [CPR23].

Another optimization to improve the performance of FESTA focuses on using only base field operations and isomorphisms to minimize the cost. This procedure turns out to be very rewarding as the resulting formulae are approximately four times faster than the pre-implemented SageMath operations and two times faster than the optimized formulae used in the SageMath implementation of the attack to SIDH presented by Castryck and Decru in [CD23].

5.6.2 Further optimizations

Within the present section, let us mention potential further optimizations that will be subject of further studies in future work.

Using larger (ℓ, ℓ)-isogenies

There is no conceptual argument for which the FESTA algorithms should be restricted to considering torsion points of order a power of two, as in the inversion algorithm one computes (ℓ, ℓ)-isogenies for any ℓ dividing the order of the torsion points. Future developments in the computation of isogenies between principally polarized abelian surfaces could provide improvements to the cost of inverting the trapdoor function, and thus also improve on the run-time of the decryption algorithm. Indeed, the reason for considering torsion points of order a power of two is strictly tied to the fact that, at the moment, formulae to compute (ℓ, ℓ)-isogenies are efficient only for $\ell = 2$.

Moreover, the authors in [BMP23, §6.3] mention that the technique used for finding parameters in Section 5.5.2 can be generalized to any prime power, but it would be necessary to develop new tools to efficiently find parameters in the case where we consider products of prime powers.

Higher dimensional trapdoor inversion

The authors in [BMP23, §6.3] also explore the idea of relying on isogenies in dimension four. Indeed, they take inspiration from [Rob23a, §6.4], where Robert proposes a way to recover a d-isogeny from the images of the m-torsion points under the assumption that $m^2 > 4d$ where he relies on isogenies of dimension four. The main advantage of this method would be to obtain smaller parameters, leading to a prime $p \approx 2^{7\lambda}$ instead of the $p \approx 2^{10\lambda}$ as it is currently. This would lead to a huge improvement in the protocol's bandwidth usage, as well as key generation and encryption running times. However, it would also lead to a larger decryption time, as a computation of higher dimensional isogenies would be required.

Further trade-offs would involve using irrational isogenies ϕ_1 and ϕ_2 to achieve very small primes to improve compactness, as well as the key generation and encryption running times. However, the decryption algorithm would be very slow since higher dimensional isogenies would have to be compute multiple times.

Chapter 6

POKE – Point-based Key Exchange

Very recently, Andrea Basso in [Bas24] explored ways to allow two parties to manipulate the representations of higher-dimensional isogenies in order to obtain P(O)KE, the most compact post-quantum PKE and the most efficient isogeny-based Public-key Encryption (PKE) scheme to date.

Since its first appearance, isogeny-based cryptography focuses on isogenies between elliptic curves, since the constructions that focused on isogenies between abelian varieties (a generalization of elliptic curves to higher dimension) [FT19; KTW21; LTZ23] were less efficient than the one-dimensional counterparts. However, higher-dimensional isogenies began to be much more considered after the publication of the SIDH attacks in August 2022 [CD23; Mai+23; Rob23a], that showed how it was possible, given two curves, the degree of the connecting isogeny, and the images of a torsion basis, to retrieve the connecting isogeny. If the degree of the isogeny is sufficiently smooth, the kernel of the isogeny can be completely recovered, but if this is not the case the attacks still allow to evaluate the isogeny at any given point. Therefore the SIDH attacks brought to the realization that two pairs of curves and torsion bases, together with the degree, can provide a way to represent an isogeny, since the knowledge of an isogeny is actually tied to one's ability to evaluate it. This type of representation, that stems from the SIDH attacks, allows to efficiently describe non-smooth isogenies between any two curves and is exploited by the POKE construction. Moreover, further improvements to the SIDH attacks showed that it was possible to recover the key in only a couple of seconds [OP22] and that, through new techniques, the same SIDH attacks could be performed in milliseconds [Dar+23].

Therefore, a renewed interest in basing cryptographic protocols on isogenies between abelian surfaces was found and schemes such as FESTA [BMP23], that relies on higher-dimensional representations of isogenies, were constructed. Additionally, using higher dimensional isogenies to compute isogenies of non-smooth degree was employed in QFESTA [NO23], an improvement of FESTA. On the same note, protocols such as SQIsign [De +20] and SCALLOP [Feo+23] obtain significant improvements by relying on the higher-dimensional representation of isogenies as well. At the same time, schemes such as MD-SIDH and M-SIDH [FMP23], even if less compact and less efficient than SIDH, improve on the one-dimensional case by obtaining key exchange protocols that avoid the SIDH attacks. However, the author in [Bas24] notices that the two approaches have never been blended together. Namely, schemes in the literature either deploy one-dimensional isogenies, that are often rational, or higher-dimensional isogenies, that are often irrational, but never use them in a mixed manner.

The POKE framework allows two parties to get involved in a two-round protocol to establish a commutative diagram and obtain a shared secret. One party will compute irrational non-smooth isogenies and the other party will compute rational smooth isogenies. The author notices that all four curves in the commutative diagram need to be public in order to compute the isogenies. Therefore both parties will have to reveal the scaled images of a fixed point through their secret isogeny. This will allow to obtain a shared point on a public curve, hence the name Point-based Key Exchange. More details on the POKE framework will be provided throughout the chapter.

The PKE scheme constructed from the POKE framework uses an irrational isogeny of secret-degree as the long-term secret key. The sender will encrypt by computing two parallel SIDH isogenies and revealing a sufficient amount of information so that the receiver will be able to complete the commutative diagram. The resulting protocol is extremely compact and efficient: it works with a prime $p \approx 2^{3\lambda}$. Moreover, the author states that the unoptimized proof-of-concept implementation in SageMath runs in less than 300 milliseconds.

6.1 Notation and preliminaries

The POKE framework relies on the two parties manipulating different representations of isogenies in order to obtain a very compact and efficient PKE scheme.

Throughout the present chapter, $\mathbb{Z}_m$ will represent the ring $\mathbb{Z}/m\mathbb{Z}$ and λ will denote the security parameter. Recall the following notions introduced in Chapter 1:

- $\phi_{A*}\phi_B$ represents the *push-forward* of $\phi_B : E_0 \to E_B$ under $\phi_A : E_0 \to E_A$ and it is such that if we let $\phi_{A*}\phi_B =: \phi'_B$, then $\phi'_B : E_A \to E_{AB}$ is such that $\ker \phi'_B = \phi_A(\ker \phi_B)$;
- Two isogenies ϕ and ϕ' are called *parallel* with respect to an isogeny ψ if $\ker \phi' = \psi(\ker \phi)$;
- *Rational* isogenies are such that their kernel is defined over $\mathbb{F}_{p^2}$;
- *Irrational* isogenies are such that their kernel is defined over a large extension of $\mathbb{F}_{p^2}$.

As already discussed in the beginning of the present chapter, the main idea underlying [Bas24] consists of considering different types of representation for different parties in the protocol. In particular, the author considers two types of representations of isogenies to construct the PKE scheme:

1. SIDH isogenies [DJP14]: they are isogenies of prime-power degree ℓ^e (in the POKE framework, $\ell \in \{3, 5\}$) that have kernel defined over $\mathbb{F}_{p^2}$. If one wants to compute the push-forward of an SIDH isogeny under a secret isogeny, the images of a basis of the torsion subgroup $E[\ell^e]$ under the secret isogeny must be revealed. To avoid the SIDH attacks, these images should be scaled by the same random scalar in $\mathbb{Z}^*_{\ell^e}$. Note that these are rational (one-dimensional) isogenies that can be represented by a single curve and a generating kernel defined over $\mathbb{F}_{p^2}$.
2. FESTA isogenies [BMP23]: they are isogenies whose degree is of the form $q(2^a - q)$, for some integer q. This type of isogenies are represented by their domain and codomain curves, together with their degree and the images of a basis of the 2^a torsion. In Section 6.2 we will see how it is possible to compute their push-forward under a secret isogeny. This will be crucial for "closing" the commutative diagram and making the scheme work. Note that these are irrational (higher-dimensional) isogenies as their kernel is defined exclusively over a large extension field of $\mathbb{F}_{p^2}$.

6.2 The POKE construction

The POKE framework, described in [Bas24, §3], will allow two parties to get involved in a protocol that will allow to establish a commutative diagram and obtain a shared secret through the use of different isogeny representations. Since the parties will work with irrational isogenies, such as FESTA isogenies, let us present a method to compute the push-forward of isogenies represented by two curves, their degree and the images of a basis of the 2^a torsion. Compared to the previous SIDH constructions, this will be possible by requiring additional interaction between the two parties.

The scheme involves two parties: Alice, that will compute any type of isogeny (even e.g. FESTA isogenies), and Bob, that will be limited to computing isogenies represented by a single curve (e.g. SIDH isogenies). The description of the protocol is kept as generic as possible since it can be built with any isogeny that respects the rules imposed for representation.

So, Bob samples a random rational isogeny $\phi_B : E_0 \to E_B$ whose degree is coprime with that of ϕ_A. After Alice has published the images of a basis of her torsion subgroup under ϕ_A, Bob can compute the push-forward of ϕ_B under ϕ_A and obtain the isogeny $\phi'_B : E_A \to E_{AB}$.

Once Bob has revealed the curves E_B and E_{AB}, as well as the images of some points of E_0 and E_A under ϕ_B and ϕ'_B, Alice can compute the push-forward of ϕ_A under ϕ_B and obtain the isogeny

$\phi'_A : E_B \to E_{AB}$. Therefore, a commutative diagram has been created, but this is not sufficient to create a key exchange. Indeed, even though the curve E_{AB} was usually used as the shared secret in similar constructions, now it cannot serve that role anymore since it must be published in order for Alice to compute her part of the diagram.

Therefore, to get around this problem, the additional point $X_0 \in E_0$ of order x is introduced. Let x be a large prime, thus coprime with both $\deg \phi_A$ and $\deg \phi_B$. Let us now add some additional details to the protocol so that the two parties can agree on a shared secret in the end. Thus, we require that Alice also reveals the image $X_A = [\alpha]\phi_A(X_0)$, for some random scalar α. Analogously, Bob reveals the image $X_B = [\beta]\phi_B(X_0)$, for some random scalar β. In conclusion, both parties will agree on the point $X_{AB} = [\alpha]\phi'_A(X_B) = [\beta]\phi'_B(X_A)$ on E_{AB} as the shared secret of the exchange. The protocol is illustrated in Figure 6.1.

$$\begin{tikzcd}
\begin{array}{c} E_0, \\ X_0 \in E_0[x] \end{array} \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & \begin{array}{c} E_A, \\ X_A = [\alpha]\phi_A(X_0) \end{array} \arrow[d, "\phi'_B"] \\
\begin{array}{c} E_B, \\ X_B = [\beta]\phi_B(X_0) \end{array} \arrow[r, "\phi'_A"'] & \begin{array}{c} E_{AB}, \\ X_{AB} = [\alpha]\phi'_A(X_B) = [\beta]\phi'_B(X_A) \end{array}
\end{tikzcd}$$

Figure 6.1: Diagram illustrating the POKE construction, taken from [Bas24, §3].

Let us briefly explain why, according to the author in [Bas24, §3], the introduction of the point X_0 does not lead to an increase in the size of the prime p and does not deteriorate the security of the protocol. Firstly, the size of the underlying prime p is not increased due to the fact that the order of X_0 does not need to be smooth. This means that the point X_0 can be defined over the extension field $\mathbb{F}_{p^4}$ and have $x \mid p - 1$. The isogeny can be efficiently evaluated on X_0 through the use of an x-only representation, while keeping the underlying prime smaller. Secondly, from a security perspective, the parties are required to reveal the scaled action of their secret isogeny on the public point X_0, and this is argued to not affect the security of the protocols since the *scaled* action of a secret isogeny on some points is already revealed by the isogeny representations. Indeed, we already have to reveal the scaled action on some basis of a torsion of smooth degree. This clearly already threatens the security of the protocol more than revealing the scaled action on a single point of prime order. Hence, the introduction of an additional point can be simple viewed as a slight increase in the parameters [Bas24, Footnote 3].

Moreover, the value x is chosen to be a large prime such that $x \approx 2^{\lambda/2}$. It is argued that even revealing the non-scaled action on X_0 would not provide any advantage to an attacker, since the application of the SIDH attack presented in [Rob23a, Theorem 4.3] would lead to a runtime of $O(x^4) \approx O(2^{2\lambda})$. To conclude, even brute-force attacks are avoided since there are approximately $x^2 \approx 2^\lambda$ possibilities for $X_{AB} \in E[x] \subseteq E_{AB}$, due to the fact that $E[x] \cong \mathbb{Z}_x \times \mathbb{Z}_x$.

6.3 The P(O)KE Public-key Encryption scheme

Within the present section we describe how to obtain a PKE scheme from the POKE construction. One party will compute irrational isogenies of non-smooth degree, i.e. FESTA isogenies, and the other will compute SIDH isogenies. This will allow to obtain a highly compact and efficient isogeny-based PKE.

The overall setup will include a prime p of the form $p = 2^a 3^b f - 1$, a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ with j-invariant 1728, a basis $\{P_0, Q_0\}$ of $E_0[2^a]$, a basis $\{R_0, S_0\}$ of $E_0[3^b]$, and a point $X_0 \in E_0$ of order x, with x being coprime with 2 and 3.

One first issue to address is that the endomorphism ring of E_0 is known and this could lead to the creation of a backdoor in the protocol by maliciously choosing the basis $\{P_0, Q_0\}$, as discussed in

[CV23]. Therefore, the author mentions that the basis $\{P_0, Q_0\}$ should be selected using a *nothing-up-my-sleeve* approach, relying on standard algorithms to compute a deterministic basis from a given curve [PDJ21]. Due to the fact that the resulting basis is uniformly random, the probability of it being backdoorable is negligible [CV23, §5.2].

6.3.1 Description

We let Bob be the party that wants to encrypt a message m to send to Alice. In order to generate a public key, Alice will have to generate an isogeny of degree $q(2^a - q)$, since it has to factor into integers that sum up to 2^a. This will allow to represent Alice's FESTA isogeny easily. To do this, the author in [Bas24, §4] suggests to adapt the algorithm proposed in QFESTA [NO23, Algorithm 2], a variant of FESTA that leverages the quaternion algebra structure. Firstly, generate an endomorphism θ of degree $q(2^a - q)3^b$ using Algorithm 8 that is based in [Koh+14]. After that, we compute the 3^b-isogeny that factors through θ, and we compute its dual to retrieve the codomain of an isogeny ϕ of degree $q(2^a - q)$. Therefore, computing the images of points under ϕ is equivalent to computing them under the isogeny $[1/3^b]\psi \circ \theta$, under the assumption that their order is coprime with 3. Through this method, we can compute the images of points of order 2^a and obtain a two-dimensional representation of ϕ. The overall process is described in Algorithm 9.

```
Algorithm 8 GenerateEndomorphism($E_0, N$)
Input: An elliptic curve $E_0$ over $\mathbb{F}_{p^2}$, a degree $N$ such that $N > p$.
Output: An endomorphism of $E_0$ of degree $N$.
Set $u := \lceil \sqrt{N/p} \rceil$.
while true do
    Sample random integers $z, t$ in $\mathbb{Z}_u$.
    Set $C = N - p(z^2 + t^2)$.
    if $C$ is prime & $C = x^2 + y^2$ then    ▷ $x, y$ computed through Cornacchia's algorithm [Cor08]
        return $[x] + [y]\iota + [z]\pi + [t]\iota\pi$
```

```
Algorithm 9 GenerateHigherDimensionalIsogeny($q, p$)
Input: A degree $q$, a prime $p$ of the form $p = 2^a B - 1$, where $B$ is smooth.
Output: A representation of a higher-dimensional isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$.
$\theta \leftarrow$ GenerateEndomorphism($E_0, q(2^a - q)B$).    ▷ Using Algorithm 8
Generate a basis $\{P_0, Q_0\}$ of $E_0[2^a]$.
Generate a basis $\{R_0, S_0\}$ of $E_0[3^b]$.
Compute $P_0' = \theta(P_0)$, $Q_0' = \theta(Q_0)$.
Compute $R_0' = \theta(R_0)$, $S_0' = \theta(S_0)$.
Find $x, y \in \mathbb{Z}_B$ such that $[x]R_0' + [y]S_0' = \mathcal{O}$.
Compute $\psi : E_0 \to E_A$ such that $\ker \psi = \langle [x]R_0 + [y]S_0 \rangle$.
Compute $P_A = [1/B]\psi(P_0')$, $Q_A = [1/B]\psi(Q_0')$.
Compute $\Phi$ such that $\ker \Phi = \langle ([-q]P_0, P_A), ([-q]Q_0, Q_A) \rangle$
return $\Phi$
```

After the generation of the $q(2^a - q)$-isogeny, the protocol proceeds as previously described. To obtain a PKE from the previous construction, Bob sends the ciphertext and also $ct' = \mathrm{KDF}(X_{AB}) \oplus m$, where KDF represents a key derivation function. Alice will recompute X_{AB} following the POKE framework and extracts the message m as $m = \mathrm{KDF}(X_{AB}) \oplus ct'$.

We hereby describe P(O)KE:

Public parameters: Let p be a prime of the form $p = 2^a 3^b f - 1$, where f is a small cofactor needed for primality. Let E_0 be the supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with j-invariant 1728. Let $\{P_0, Q_0\}$ be a basis of $E_0[2^a]$ and $\{R_0, S_0\}$ be a basis of $E[3^b]$ deterministically chosen. Let X_0 be a deterministically-generated point on $E_0(\mathbb{F}_{p^4})$ of order x such that $x \mid p - 1$ and $\gcd(x, 2) = \gcd(x, 3) = 1$. Let KDF represent a key derivation function.

KeyGen: Alice samples a random prime $q \in \{2, \dots, 2^a\}$ such that $q(2^a - q)$ is coprime with 3. Alice computes a random isogeny $\phi_A : E_0 \to E_A$ of degree $q(2^a - q)$ using Algorithm 9. Alice samples random integers $(\alpha_2, \alpha_2', \alpha_3, \alpha_x) \in_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{3^b}^* \times \mathbb{Z}_x^*$. Alice's public key consists of the tuple $(E_A, P_A, Q_A, R_A, S_A, X_A)$, where $P_A = [\alpha_2]\phi_A(P_0)$, $Q_A = [\alpha_2']\phi_A(Q_0)$ are 2^a-images, $R_A = [\alpha_3]\phi_A(R_0)$, $S_A = [\alpha_3]\phi_A(S_0)$ are 3^b-images, and $X_A = [\alpha_x]\phi_A(X_0)$ is the scaled image of X_0. Alice's secret key will be the tuple $(\alpha_2, \alpha_2', \alpha_3, x, q)$.

Encrypt: To encrypt a message m, Bob samples a random integer $\beta \in_\$ \mathbb{Z}_{3^b}$ and computes the two parallel isogenies $\phi_B : E_0 \to E_B$ with kernel $\langle R_0 + [\beta]S_0 \rangle$ and $\phi_B' : E_A \to E_{AB}$ with kernel $\langle R_A + [\beta]S_A \rangle$. Bob samples random integers $(\beta_2, \beta_2', \beta_x) \in_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{2^a}^* \times \mathbb{Z}_x^*$. Bob computes the points $X_B = [\beta_x]\phi_B(X_0)$ and $X_{AB} = [\beta_x]\phi_B'(X_A)$. The ciphertext is the tuple $(E_B, P_B, Q_B, X_B, E_{AB}, P_{AB}, Q_{AB}, ct')$, where $P_B = [\beta_2]\phi_B(P_0)$, $Q_B = [\beta_x]\phi_B(Q_0)$, $P_{AB} = [\beta_2]\phi_B'(P_A)$, $Q_{AB} = [\beta_2']\phi_B'(Q_A)$, and $ct' = \mathrm{KDF}(X_{AB}) \oplus m$.

Decrypt: To decrypt the ciphertext, Alice scales the points P_{AB}, Q_{AB} by $1/\alpha_2$ and $1/\alpha_2'$ respectively to obtain the images of P_B and Q_B under ϕ_A'. Given this information, she obtians the two-dimensional representation of ϕ_A', from which she can obtain the point $X_{AB} = [\alpha_x]\phi_A'(X_B)$. The message is computed as $m = \mathrm{KDF}(X_{AB}) \oplus ct'$.

Remark 29. *The endomorphism ring of E_0 is supposed to be known, since this is essential to compute an isogeny of non-smooth degree when performing the* **KeyGen** *algorithm. The author in [Bas24, §4] states that this does not affect the security of the protocol, as long as more attention is given to the selection of the points P_0 and Q_0. Moreover, to prevent any type of attack that leverages the knowledge of the endomorphism ring of E_0, as already done in [Pet17; CV23], the author suggests that one could be prudent by replacing the curve E_0 with some $\overline{E}_0$ of unknown endomorphism ring. Let us see how.*

Alice can start from the curve with j-invariant 1728 and then generate the key as described in the **KeyGen** *algorithm to obtain the tuple $(E_A, P_A, Q_A, R_A, S_A, X_A)$. After this, Alice computes two parallel isogenies $\psi : E_0 \to \overline{E}_0$ and $\psi' : E_A \to \overline{E}_A$. Subsequently she computes the images $\psi(P_0), \psi(Q_0)$ and $\psi'(P_A), \psi'(Q_A)$, from which she can retrieve a two dimensional representation of the isogeny $\overline{\phi}_A : \overline{E}_0 \to \overline{E}_A$ that has degree $q(2^b - q)$. We have the configuration depicted in Figure 6.2.*

$$\begin{tikzcd}
E_0 \arrow[r, "\psi"] & \overline{E}_0 \arrow[d, "\overline{\phi}_A"] \\
E_A \arrow[r, "\psi'"] & \overline{E}_A
\end{tikzcd}$$

Figure 6.2: Configuration required for an alternative key generation procedure.

Alice then generates a 2^a-basis of $\overline{E}_0[2^a]$, a 3^b-basis of $\overline{E}_0[3^b]$ and a point of order x. Due to the fact that she has a higher-dimensional representation of $\overline{\phi}_A$, she can map the points onto the curve $\overline{E}_A$ and scale them as done in the original **KeyGen***. In the end, the public key consists of the curves $\overline{E}_0$, $\overline{E}_A$, and the points on $\overline{E}_A$. Therefore now the starting curve of the protocol has unknown endomorphism to anyone but Alice.*

Overall, the starting curve E_0, that was part of the public parameters, is now considered part of the public key as $\overline{E}_0$ and is connected to $\overline{E}_A$ through the secert isogeny $\overline{\phi}_A$. So, as a concluding remark, we notice that we had a trade-off to achieve more security at the expenses of a longer key generation procedure and a larger public key, that now is made up of two curves and five points instead of one curve and five points.

6.3.2 Security analysis

Hardness assumptions and Security proof

On a high-level, the security of the scheme requires that it is hard to recover either the secret isogeny ϕ_A from the public key, or recovering the isogenies ϕ_B and ϕ_B' from the ciphertext.

Formalizing this, the security of Alice's secret isogeny relies on the hardness of the following problem:

Problem 9 (Secret-Degree Isogeny (SDI) Problem)**.** *Let p be a prime of the form $p = 2^a 3^b f - 1$, with x a prime such that $x \mid p-1$. Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let $\{P_0, Q_0\}$ be a basis of $E[2^a]$ and let $\{R_0, S_0\}$ be a basis of $E[3^b]$. Let X_0 be a point of order x. Let q be a random prime in $\{2, \ldots 2^a\}$. Let $\phi : E \to E'$ be a random isogeny of degree $q(2^a - q)$, generated as in Algorithm 9. Let $(\alpha_2, \alpha_2', \alpha_3, \alpha_x) \leftarrow_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{3^b}^* \times \mathbb{Z}_x^*$.*

Given the curves E, E', the points P_0, Q_0, R_0, S_0, X_0 and the image points $[\alpha_2]\phi(P_0), [\alpha_2']\phi(Q_0)$, $[\alpha_3]\phi(R_0), [\alpha_3]\phi(S_0), [\alpha_x]\phi(X_0)$, recover ϕ.

The security of the isogenies ϕ_B and ϕ_B', computed during encryption, relies on the hardness of the following problem:

Problem 10 (Double Masked-Torsion Isogeny (DMTI) Problem)**.** *Let E_0, E_0' be two supersingular curves defined over $\mathbb{F}_{p^2}$, connected by an isogeny $\psi : E_0 \to E_0'$. Let $\{P_0, Q_0\}$ be a basis of $E_0[2^a]$ and let $\{R_0, S_0\}$ be a basis of $E_0'[2^a]$. Let X_0 be a point on E_0 of order x. Let $\phi : E_0 \to E_1$ be a random isogeny of degre 3^b, and let $\phi' : E_0' \to E_1'$ be its push-forward $\psi_*\phi$. Let $(\beta_2, \beta_2', \beta_x) \leftarrow_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{2^a}^* \times \mathbb{Z}_x^*$.*

Given E_0, E_0', the points $P_0, Q_0, X_0, P_0', Q_0'$ the curves E_1, E_1', the image points $[\beta_2]\phi(P_0), [\beta_2']\phi(Q_0)$, $[\beta_x]\phi(X_0), [\beta_2]\phi(P_0'), [\beta_2']\phi(Q_0')$, recover ϕ.

Assuming that Problem 9 and Problem 10 cannot be solved efficiently, the security against isogeny recovering attacks is guaranteed. The author in [Bas24, §4] states that in reality, the overall scheme relies on a stronger assumption, that involves the hardness of the following computational $\text{POKE}^{\text{FESTA,SIDH}}$ problem.

Problem 11 ($\text{C-POKE}^{\text{FESTA,SIDH}}$)**.** *Let p be a prime of the form $p = 2^a 3^b f - 1$, where f is a small cofactor needed for primality. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let $\{P_0, Q_0\}$ be a basis of $E_0[2^a]$. Let $\{R_0, S_0\}$ be a basis of $E_0[3^b]$. Let X_0 a point of order x on $E_0(\mathbb{F}_{p^4})$.*

Let $\phi_A : E_0 \to E_A$ be an isogeny of degree $q(2^a - q)$, for some unknown value q. Let $(\alpha_2, \alpha_2', \alpha_3, \alpha_3') \in_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{3^b}^* \times \mathbb{Z}_x^*$. Let $P_A = [\alpha_2]\phi_A(P_0), Q_A = [\alpha_2']\phi_A(Q_0), R_A = [\alpha_3]\phi_A(R_0), S_A = [\alpha_3']\phi_A(S_0)$, and $X_A = [\alpha]\phi_A(X_0)$. Let $\phi_B : E_0 \to E_B$ be an isogeny of degree 3^b. Let $\phi_B' : E_A \to E_{AB}$ be the push-forward $(\phi_A)_*(\phi_B)$. Let $(\beta_2, \beta_2', \beta_x) \in_\$ \mathbb{Z}_{2^a}^* \times \mathbb{Z}_{3^b}^* \times \mathbb{Z}_x^*$. Let $P_B = [\beta_2]\phi_B(P_0), Q_B = [\beta_2']\phi_B(Q_0), P_{AB} = [\beta_2]\phi_B'(P_A), Q_{AB} = [\beta_2']\phi_B'(Q_A)$, and $X_B = [\beta]\phi_B(X_0)$. Let $X_{AB} = \phi_B'(\phi_A)(X_0)$.

Given $(E_A, (P_A, Q_A), (R_A, S_A), X_A), (E_B, (P_B, Q_B), X_B)$, and $(E_{AB}, (P_{AB}, Q_{AB}))$, compute X_{AB}.

The IND-CPA security of P(O)KE is formalized in the following theorem.

Theorem 8. *The $\text{POKE}^{\text{FESTA,SIDH}}$ protocol is IND-CPA secure in the random oracle model under the assumption that the $\text{C-POKE}^{\text{FESTA,SIDH}}$ problem is hard.*

Proof. The proof is analogous to the proof of security of the hashed ElGamal PKE [KM03, Theorem 1]. Let us prove that, if an adversary $\mathcal{A}$ can break the IND-CPA security of the proposed protocol, it is possible to construct an adversary $\mathcal{B}$ that solves an instance $(E_B, P_B, Q_B, X_B, E_{AB}, P_{AB}, Q_{AB}, ct')$ of the $\text{C-POKE}^{\text{FESTA,SIDH}}$ problem. Informally, $\mathcal{B}$ simulates the random oracle model and passes $(E_B, P_B, Q_B, X_B, E_{AB}, P_{AB}, Q_{AB}, ct^*)$ to $\mathcal{A}$, where ct^* is sampled at random. If $\mathcal{A}$ does not query the ROM with X_{AB}, it cannot win the IND-CPA game. If it does, $\mathcal{B}$ can output a random query. With non-negligible probability (since X_{AB} is guaranteed to be among the queries), $\mathcal{B}$ outputs the solution to the $\text{C-POKE}^{\text{FESTA,SIDH}}$ problem. □

It is possible to construct an IND-CPA secure PKE scheme in the standard model from the decisional variant of the $\text{C-POKE}^{\text{FESTA,POKE}}$ problem. Moreover, though the use of the Fujisaki-Okamoto transform [FO99], an IND-CCA secure PKE scheme can be obtained.

Hardness analysis

To solve Problem 9, one would need to recover an isogeny of unknown degree from its action on some torsion bases. Heuristically, this is assumed to be hard from an argument that is very similar to the one proposed for the security of the M-SIDH and MD-SIDH variants. Indeed, the SIDH attacks exploit the action of the secret isogeny on some torsion points, but rely on knowing the degree of the isogeny. Therefore, it seems like Problem 9 reduces to recovering the degree $q(2^a - q)$, given action of the isogeny on some torsion bases. This problem is expected to be computationally hard to solve since the degree cannot be brute-forced in an efficient way (the number of degrees is exponential) nor computed through Weil pairing computations. Indeed, we have $e_{2^a}(P, Q) = e_{2^a}(P', Q') = q(2^a - q)\alpha_2\alpha_2'$ and also $e_{3^b}(R, S) = e_{3^b}(R', S') = q(2^a - q)\alpha_3^2$, where we see that no information can be learned on the degree, that is masked by the unknown scalars α_2, α_2' and α_3. An additional note is that P(O)KE reveals even less information than the M-SIDH and MD-SIDH variants since in the latter cases the degree of the isogeny was known to be a divisor of a public parameter. This is not the case in P(O)KE, where significantly less information is provided, since part of the images of the torsion points are scaled by a diagonal matrix, and no multiple of the degree is known. For these reasons, Problem 9 is expected to be at least as hard as Problem 5, on which MD-SIDH relies.

To solve Problem 10, one would need to recover the secret isogeny from its scaled action on the torsion points. This problem is a *two-instance variant* of Problem 7, known as the CIST problem on which FESTA relies. Since the revealed images of the torsion points are scaled under different unknown scalars, not enough information is revealed to apply the SIDH attacks, and therefore the CIST problem is believed to be hard. The author in [Bas24, §4] states that the additional information that is revealed by Problem 10 compared to the CIST problem does not help in solving the problem. Recall that FESTA relies on the hardness of CIST^2, that is a two-instance version of CIST as well, where the two isogenies are independent and have different degrees. In our case, instead, the two isogenies are the push-forward of one another under the isogeny connecting E and E'. However, the author states that heuristically this feature does not seem to help in solving Problem 10. Not even the fact that the two isogenies are parallel seems to be exploitable, since the isogeny connecting E and E' is part of Alice's secret key and is unknown to an attacker.

6.3.3 Efficiency analysis

Let us present the main advantage of the scheme: it is the most compact post-quantum PKE scheme and also one of the most efficient isogeny-based PKE schemes to date. The compactness and efficiency of the scheme stem from the fact that the parties use different representations of isogenies and work with irrational isogenies. We first discuss the parameters of P(O)KE and then compare its performance to other protocols.

Parameters: In order to guarantee security of the scheme, one must select $2^a \approx 2^\lambda$ (in this way brute-forcing q is hard) and $3^b \approx 2^{2\lambda}$ (in this way we protect against meet-in-the-middle attacks to recover ϕ_B), and $x \approx 2^{\lambda/2}$ (in this way recovering X_{AB} by brute-force is expensive). Therefore, the prime $p = 2^a 3^b f - 1$ is approximately $p \approx 2^{3\lambda}$.
The public key is the tuple $(E_A, P_A, Q_A, R_A, S_A, X_A)$ and therefore is made of one curve and five points. This means that to represent the public key one needs $\approx 38\lambda$ bits – or if one compresses the points $\approx 24\lambda$ bits [Gal12, §11.7]. The ciphertext is the tuple $(E_B, P_B, Q_B, X_B, E_{AB}, P_{AB}, Q_{AB}, ct')$ and therefore is made of two curves, five torsion points, and an encrypted message. This means that to represent the ciphertext one needs $\approx 38\lambda$ bits – or if one compresses the points $\approx 24\lambda$ bits.
If one targets $\lambda = 128$, the compressed public key and ciphertext will require approximately 272 bytes and 384 bytes respectively.

Performance and comparison with other protocols: Before delving into some details, one must acknowledge that the protocol was implemented though its uncompressed version and deploying isogenies between abelian surfaces as proposed in [Dar+23]. Moreover, limitations in the implementation imposes the point X_0 to be defined over $\mathbb{F}_{p^2}$, instead of $\mathbb{F}_{p^4}$, requiring a characteristic about 16% larger than necessary. Therefore, the results presented in Table 6.1 serve as upper bounds as future work will attempt to improve on these details, expecting a speed-up of about 25% once the point X_0 will be defined over $\mathbb{F}_{p^4}$ with a smaller characteristic. Additionally, the author forecasts improvements of one order of magnitude when the same algorithms will be implemented with some assembly-level optimizations.

	Size (bytes)		**Time (ms)**		
λ	$\lvert\mathsf{pk}_{\mathsf{cmp}}\rvert$	$\lvert\mathsf{ct}_{\mathsf{cmp}}\rvert$	KeyGen	Encrypt	Decrypt
128	272	384	554.7	165.0	275.3
192	408	576	1149.3	315.8	559.4
256	544	768	2154.6	576.6	1053.2

Table 6.1: Performance of the P(O)KE PKE.

When comparing this protocol to FESTA, we notice 2.9× smaller ciphertexts and more than one order of magnitude faster computations in P(O)KE. The author in [Bas24, §4.1] also compares this new construction to CSIDH [Cas+18], QFESTA [NO23] and terSIDH-hyb [BF23], highlighting the efficiency and compactness of P(O)KE. However, these schemes are outside the scope of the present survey.

Chapter 7

Conclusions

In conclusion, this report has explored the realm of isogeny-based cryptography, with a particular focus on the Supersingular Isogeny Diffie-Hellman (SIDH) protocol. Initially praised and considered very promising for its efficiency and security properties in the post-quantum era, SIDH's security landscape underwent an abrupt change in the Summer of 2022, after the publication of a series of devastating attacks. These attacks, while highlighting the importance of ongoing cryptanalysis in ensuring secure cryptosystems, caused the development of countermeasures and entirely new cryptographic constructions.

Part I of this report introduced isogeny-based cryptography, meticulously presented the SIDH protocol, and finally explored the series of attacks that rendered SIDH insecure, emphasizing their impact and the different attack scenarios.

Part II explored various responses to the SIDH attacks. We examined two countermeasure approaches, M-SIDH and MD-SIDH, which aim to preserve the SIDH framework while mitigating vulnerabilities. Subsequently, we analyzed two new cryptographic constructions, FESTA and POKE, that were born from a constructive application of the SIDH attacks. These novel schemes leverage the insights gained from the attacks to achieve efficient and compact cryptosystems as the effort to develop secure and efficient post-quantum cryptography remains an ongoing urgency.

The reader must have found extremely interesting the fact that, while the SIDH attacks at the moment might have represented a setback for isogeny-based cryptography, the resulting countermeasures and new constructions demonstrate the resilience and adaptability of the field, that is getting more and more mature and, through its active development, presents itself with many areas to further explore. Indeed, the alternating roles and activities of cryptanalysis and cryptographic design will produce advancements that will ultimately lead to robust and quantum-resistant cryptographic solutions.

Concerning future work, the author hopes to be able to work once again on the topic of isogeny-based cryptography, as the lid has just been opened and the surface has just been scratched. Indeed, isogeny-based cryptography, while being a quite niche topic, carries a large amount of intricacies that were not faced throughout the work dedicated to this report. Therefore, apart from possibly exploring the countermeasures and new constructions that were not covered throughout the project, such as binSIDH, terSIDH [BF23], QFESTA [NO23], and IS-CUBE [Mor23], the author wishes to have the opportunity to study in more detail quadratic imaginary fields, quaternion algebras, and their applications, as well as zero-knowledge proofs based on isogeny assumptions.

Acknowledgments

I extend my sincere gratitude to Professor Serge Vaudenay for providing me with the invaluable opportunity of conducting a research project in the esteemed "Security and Cryptography Laboratory" (LASEC), created at EPFL in 2000.

Moreover, I would like to express my deepest appreciation towards my supervisor, Laurane Marco, as her role within the scope of this project was indispensable to say the least. Indeed, I am grate-

ful for her invaluable guidance, support, and mentorship throughout the preparation of this report. Her expertise, encouragement, and insightful feedback have been crucial in shaping the content and direction of this work.

Lastly, I would also like to extend my thanks to Parsa Tasbihgou for our engaging and enlightening conversations during our collaboration on the team project for the course “COM-506: Student seminar on security protocols and applications” offered at EPFL. His perspectives and insights have enriched my understanding of the subject and have contributed to a rewarding learning experience.

The support and encouragement of the people mentioned above have been immensely appreciated and have made a significant difference in the completion of this work, by supporting and inspiring me throughout this effort.

Bibliography

[Cor08] Giuseppe Cornacchia. "Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} x^{n-h}y^{h}$". In: *Giornale di Matematiche di Battaglini* 46 (1908), pp. 33–90.

[SS66] Daniel Shanks and Larry P Schmid. "Variations on a theorem of Landau. I". In: *Mathematics of Computation* 20.96 (1966), pp. 551–569. DOI: 10.2307/2003544.

[Wat69] William C. Waterhouse. "Abelian varieties over finite fields". In: *Annales Scientifiques De L Ecole Normale Superieure* 2 (1969), pp. 521–560. URL: https://api.semanticscholar.org/CorpusID:264486669.

[Vél71] Jacques Vélu. "Isogénies entre courbes elliptiques". In: *Comptes-Rendus de l'Académie des Sciences* 273 (1971), pp. 238–241. URL: https://aghitza.org/publications/translation-velu/.

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[Lam79] Leslie Lamport. "Constructing digital signatures from a one way function". In: (1979). URL: https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische annalen* 261 (1982), pp. 515–534. DOI: 10.1007/BF01457454.

[RS86] Michael O Rabin and Jeffery O Shallit. "Randomized algorithms in number theory". In: *Communications on Pure and Applied Mathematics* 39.S1 (1986), S239–S256. DOI: 10.1002/cpa.3160390713.

[Len87] Hendrik W Lenstra Jr. "Factoring integers with elliptic curves". In: *Annals of mathematics* (1987), pp. 649–673. DOI: 10.2307/1971363.

[Mon87] Peter L Montgomery. "Speeding the Pollard and elliptic curve methods of factorization". In: *Mathematics of computation* 48.177 (1987), pp. 243–264. DOI: 10.1090/S0025-5718-1987-0866113-7.

[Bro93] Bradley W. Brock. "Superspecial curves of genera two and three". In: 1993. URL: https://api.semanticscholar.org/CorpusID:118408211.

[Gro96] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. "The Magma algebra system I: The user language". In: *Journal of Symbolic Computation* 24.3-4 (1997), pp. 235–265. DOI: 10.1006/jsco.1996.0125.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. "Statistical zero knowledge protocols to prove modular polynomial relations". In: *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer. 1997, pp. 16–30. DOI: 10.1007/BFb0052225.

[Kan97] Ernst Kani. "The number of curves of genus two with elliptic differentials." In: (1997). DOI: 10.1515/crll.1997.485.93.

[Elk+98] Noam D Elkies et al. “Elliptic and modular curves over finite fields and related computational issues”. In: *AMS IP STUDIES IN ADVANCED MATHEMATICS* 7 (1998), pp. 21–76. DOI: https://api.semanticscholar.org/CorpusID:2038806.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. “Secure integration of asymmetric and symmetric encryption schemes”. In: *Annual international cryptology conference*. Springer. 1999, pp. 537–554. DOI: 10.1007/3-540-48405-1_34..

[Gal99] Steven D Galbraith. “Constructing isogenies between elliptic curves over finite fields”. In: *LMS Journal of Computation and Mathematics* 2 (1999), pp. 118–138. DOI: 10.1112/S1461157000000097.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential power analysis”. In: *Advances in Cryptology—CRYPTO’99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer. 1999, pp. 388–397. DOI: 10.1007/3-540-48405-1_25.

[Sho99] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S009753979529317.

[Tes99] Edlyn Teske. “The Pohlig–Hellman method generalized for group structure computation”. In: *Journal of symbolic computation* 27.6 (1999), pp. 521–534. DOI: 10.1006/jsco.1999.0279.

[KM03] Eike Kiltz and John Malone-Lee. “A general construction of IND-CCA2 secure public key encryption”. In: *IMA International Conference on Cryptography and Coding*. Springer. 2003, pp. 152–166. DOI: 10.1007/978-3-540-40974-8_13.

[Smi+05] Benjamin Andrew Smith et al. “Explicit endomorphisms and correspondences”. In: (2005). URL: http://hdl.handle.net/2123/1066.

[Cou06] Jean-Marc Couveignes. “Hard Homogeneous Spaces.” In: *IACR Cryptology ePrint Archive* 2006 (Jan. 2006), p. 291. URL: https://www.researchgate.net/publication/220334486_Hard_Homogeneous_Spaces.

[RS06] Alexander Rostovtsev and Anton Stolbunov. “Public-key cryptosystem based on isogenies”. In: *Cryptology ePrint Archive* (2006). URL: https://www.researchgate.net/publication/220336062_Public-Key_Cryptosystem_Based_on_Isogenies.

[Tes06] Edlyn Teske. “An elliptic curve trapdoor system”. In: *Journal of cryptology* 19 (2006), pp. 115–133. DOI: 10.1007/s00145-004-0328-3.

[BL07] Daniel J Bernstein and Tanja Lange. “Faster addition and doubling on elliptic curves”. In: *Advances in Cryptology–ASIACRYPT 2007: 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007. Proceedings 13*. Springer. 2007, pp. 29–50. DOI: 10.1007/978-3-540-76900-2_3.

[Ber+08] Daniel J Bernstein et al. “Twisted edwards curves”. In: *Progress in Cryptology–AFRICACRYPT 2008: First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings 1*. Springer. 2008, pp. 389–405. DOI: 10.1007/978-3-540-68164-9_26.

[Was08] Lawrence C Washington. *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.

[Brö09] Reinier Bröker. “Constructing supersingular elliptic curves”. In: *J. Comb. Number Theory* 1.3 (2009), pp. 269–273. DOI: 10.48550/arXiv.1301.6875.

[CLG09] Denis X Charles, Kristin E Lauter, and Eyal Z Goren. “Cryptographic hash functions from expander graphs”. In: *Journal of CRYPTOLOGY* 22.1 (2009), pp. 93–113. DOI: 10.1007/s00145-007-9002-x.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*. 2nd edition. Vol. 106. Springer, 2009.

[Tan09] Seiichiro Tani. “Claw finding algorithms using quantum walk”. In: *Theoretical Computer Science* 410.50 (2009), pp. 5285–5297. DOI: 10.1016/j.tcs.2009.08.030.

[LW10] Chung Ki Li and Duncan S Wong. “Signcryption from randomness recoverable public key encryption”. In: *Information Sciences* 180.4 (2010), pp. 549–559. DOI: 10.1016/j.ins.2009.10.015.

[JD11] David Jao and Luca De Feo. “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies”. In: *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4*. Springer. 2011, pp. 19–34. DOI: 10.1007/978-3-642-25405-5_2.

[Gal12] Steven D Galbraith. *Mathematics of public key cryptography.* Cambridge University Press, 2012.

[DA13] Angsuman Das and Avishek Adhikari. “Signcryption from randomness recoverable PKE revisited”. In: *Information Systems Security: 9th International Conference, ICISS 2013, Kolkata, India, December 16-20, 2013. Proceedings 9.* Springer. 2013, pp. 78–90. DOI: 10.1007/978-3-642-45204-8_6.

[Har13] Robin Hartshorne. *Algebraic geometry.* Vol. 52. Springer Science & Business Media, 2013.

[DJP14] Luca De Feo, David Jao, and Jérôme Plût. “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies”. In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247. DOI: 10.1515/jmc-2012-0015.

[Koh+14] David Kohel et al. “On the quaternion-isogeny path problem”. In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 418–432. URL: https://eprint.iacr.org/2014/505..

[Aza+16] Reza Azarderakhsh et al. “Key compression for isogeny-based cryptosystems”. In: *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography.* 2016, pp. 1–10. DOI: 10.1145/2898420.2898421.

[CLN16] Craig Costello, Patrick Longa, and Michael Naehrig. “Efficient algorithms for supersingular isogeny Diffie-Hellman”. In: *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I 36.* Springer. 2016, pp. 572–601. DOI: 10.1007/978-3-662-53018-4_21.

[BL17] Daniel J Bernstein and Tanja Lange. “Montgomery curves and the Montgomery ladder”. In: (2017). URL: Montgomery%20curves%20and%20the%20Montgomery%20ladder.

[CH17] Craig Costello and Huseyin Hisil. “A simple and compact algorithm for SIDH with arbitrary degree isogenies”. In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23.* Springer. 2017, pp. 303–329. DOI: 10.1007/978-3-319-70697-9_11.

[Cos+17] Craig Costello et al. “Efficient compression of SIDH public keys”. In: *Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part I 36.* Springer. 2017, pp. 679–706. DOI: 10.1007/978-3-319-56620-7_24.

[De 17] Luca De Feo. “Mathematics of isogeny based cryptography”. In: *arXiv preprint arXiv: 1711.04062* (2017). DOI: 10.48550/arXiv.1711.04062.

[GPS17] Steven D Galbraith, Christophe Petit, and Javier Silva. “Identification protocols and signature schemes based on supersingular isogeny problems”. In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23.* Springer. 2017, pp. 3–33. DOI: 10.1007/s00145-019-09316-0.

[Jao+17] David Jao et al. *SIKE: Supersingular Isogeny Key Encapsulation.* 2017. URL: https://sike.org/.

[KLV17] Susumu Kiyoshima, Huijia Lin, and Muthuramakrishnan Venkitasubramaniam. "A unified approach to constructing black-box UC protocols in trusted setup models". In: *Theory of Cryptography: 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I 15.* Springer. 2017, pp. 776–809. DOI: 10.1007/978-3-319-70500-2_26.

[Pet17] Christophe Petit. "Faster algorithms for isogeny problems using torsion point images". In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23.* Springer. 2017, pp. 330–353. DOI: 10.1007/978-3-319-70697-9_12.

[ST17] National Institute of Standards and Technology. *NIST: Post-Quantum Cryptography Standardization.* 2017. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[AJL18] Reza Azarderakhsh, David Jao, and Christopher Leonardi. "Post-quantum static-static key agreement using multiple protocol instances". In: *Selected Areas in Cryptography–SAC 2017: 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers 24.* Springer. 2018, pp. 45–63. DOI: 10.1007/978-3-319-72565-9_3.

[Cas+18] Wouter Castryck et al. "CSIDH: an efficient post-quantum commutative group action". In: *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24.* Springer. 2018, pp. 395–427. DOI: doi.org/10.1007/978-3-030-03332-3_15.

[CS18] Craig Costello and Benjamin Smith. "Montgomery curves and their arithmetic: The case of large characteristic fields". In: *Journal of Cryptographic Engineering* 8.3 (2018), pp. 227–240. DOI: 10.1007/s13389-017-0157-6.

[GV18] Steven D Galbraith and Frederik Vercauteren. "Computational problems in supersingular elliptic curve isogenies". In: *Quantum Information Processing* 17.10 (2018), p. 265. DOI: 10.1007/s11128-018-2023-6.

[MR18] Michael Meyer and Steffen Reith. "A faster way to the CSIDH". In: *Progress in Cryptology–INDOCRYPT 2018: 19th International Conference on Cryptology in India, New Delhi, India, December 9–12, 2018, Proceedings 19.* Springer. 2018, pp. 137–152. DOI: 10.1007/978-3-030-05378-9_8.

[Ren18] Joost Renes. "Computing isogenies between Montgomery curves using the action of (0, 0)". In: *International Conference on Post-Quantum Cryptography.* Springer. 2018, pp. 229–247. DOI: 10.1007/978-3-319-79063-3_11.

[Zan+18] Gustavo HM Zanon et al. "Faster key compression for isogeny-based cryptosystems". In: *IEEE Transactions on Computers* 68.5 (2018), pp. 688–701. DOI: 10.1109/TC.2018.2878829.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *International Conference on the Theory and Application of Cryptology and Information Security.* Springer. 2019, pp. 227–247. DOI: doi.org/10.1007/978-3-030-34578-5_9.

[DG19] Luca De Feo and Steven D Galbraith. "SeaSign: compact isogeny signatures from class group actions". In: *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38.* Springer. 2019, pp. 759–789. DOI: 10.1007/978-3-030-17659-4_26.

[FT19] E Victor Flynn and Yan Bo Ti. “Genus two isogeny cryptography”. In: *Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019 Revised Selected Papers 10*. Springer. 2019, pp. 286–306. DOI: 10.1007/978-3-030-25510-7_16..

[FS19] Gerhard Frey and Tony Shaska. *Curves, Jacobians, and cryptography*. Vol. 724. American Mathematical Society Providence, 2019. DOI: 10.1090/conm/724/14596.

[MP19] Chloe Martindale and Lorenz Panny. “How to not break SIDH”. In: *Cryptology ePrint Archive* (2019). URL: https://hdl.handle.net/1983/5f2e15b4-f72c-4b8a-ac31-14ac521ded28.

[Ala+20] Navid Alamati et al. “Cryptographic group actions and applications”. In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*. Springer. 2020, pp. 411–439. DOI: 10.1007/978-3-030-64834-3_14.

[Ber+20] Daniel J Bernstein et al. “Faster computation of isogenies of large prime degree”. In: *Open Book Series* 4.1 (2020), pp. 39–55. DOI: 10.2140/obs.2020.4.39.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. “Oblivious pseudorandom functions from isogenies”. In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*. Springer. 2020, pp. 520–550. DOI: 10.1007/978-3-030-64834-3_18.

[BS20] Dan Boneh and Victor Shoup. “A graduate course in applied cryptography”. In: *Draft 0.5* (2020).

[Cos20a] Craig Costello. “B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion”. In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*. Springer. 2020, pp. 440–463. DOI: 10.1007/978-3-030-64834-3_15.

[Cos20b] Craig Costello. “Supersingular isogeny key exchange for beginners”. In: *Selected Areas in Cryptography–SAC 2019: 26th International Conference, Waterloo, ON, Canada, August 12–16, 2019, Revised Selected Papers 26*. Springer. 2020, pp. 21–50. DOI: 10.1007/978-3-030-38471-5_2.

[De +20] Luca De Feo et al. “SQISign: compact post-quantum signatures from quaternions and isogenies”. In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*. Springer. 2020, pp. 64–93. DOI: 10.1007/978-3-030-64837-4_3.

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. “Chosen ciphertext security from injective trapdoor functions”. In: *Annual International Cryptology Conference*. Springer. 2020, pp. 836–866. DOI: 10.1007/978-3-030-56784-2_28.

[LB20] Jonathan Love and Dan Boneh. “Supersingular curves with small noninteger endomorphisms”. In: *Open Book Series* 4.1 (2020), pp. 7–22. DOI: 10.2140/obs.2020.4.7.

[Bas+21] Andrea Basso et al. “Cryptanalysis of an oblivious PRF from supersingular isogenies”. In: *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*. Springer. 2021, pp. 160–184. DOI: 10.1007/978-3-030-92062-3_6.

[De +21] Luca De Feo et al. “Séta: Supersingular encryption from torsion attacks”. In: *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27*. Springer. 2021, pp. 249–278. DOI: 10.1007/978-3-030-92068-5_9.

[FP21] Tako Boris Fouotsa and Christophe Petit. “SHealS and HealS: isogeny-based PKEs from a key validation method for SIDH”. In: *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27*. Springer. 2021, pp. 279–307. DOI: 10.1007/978-3-030-92068-510.

[KTW21] Sabrina Kunzweiler, Yan Bo Ti, and Charlotte Weitkämper. “Secret keys in genus-2 SIDH”. In: *International Conference on Selected Areas in Cryptography*. Springer. 2021, pp. 483–507. DOI: 10.1007/978-3-030-99277-4_23.

[Pan21] Lorenz Panny. “Cryptography on isogeny graphs”. PhD thesis. 2021. URL: https://yx7.cc/.

[PDJ21] Geovandro Pereira, Javad Doliskani, and David Jao. “x-only point addition formula and faster compressed SIKE”. In: *Journal of Cryptographic Engineering* 11.1 (2021), pp. 57–69. DOI: 10.1007/s13389-020-00245-4.

[Que+21] Victoria de Quehen et al. “Improved torsion-point attacks on SIDH variants”. In: *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*. Springer. 2021, pp. 432–470. DOI: 10.1007/978-3-030-84252-9_15.

[Sut21] Andrew Sutherland. *Elliptic curves – Chapter 5: Isogeny kernels and division polynomials*. MIT: Massachusetts Institute of Technology. 2021. URL: https://math.mit.edu/classes/18.783/2021/LectureNotes5.pdf.

[Cor+22] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2022.

[Ebr22] Ehsan Ebrahimi. “Post-quantum security of plain OAEP transform”. In: *IACR International Conference on Public-Key Cryptography*. Springer. 2022, pp. 34–51. DOI: doi.org/10.1007/978-3-030-97121-2_2.

[FP22] Tako Boris Fouotsa and Christophe Petit. “A new adaptive attack on SIDH”. In: *Cryptographers’ Track at the RSA Conference*. Springer. 2022, pp. 322–344. DOI: 10.1007/978-3-030-95312-6_14.

[Ler22] Antonin Leroux. “Quaternion Algrebras and isogeny-based cryptography”. PhD thesis. Institut polytechnique de Paris, 2022. URL: https://www.researchgate.net/publication/366092462_Quaternion_Algebra_and_isogeny-based_cryptography.

[MM22] Luciano Maino and Chloe Martindale. *An attack on SIDH with arbitrary starting curve*. Cryptology ePrint Archive, Paper 2022/1026. 2022. URL: https://eprint.iacr.org/2022/1026.

[MMP22] Marzio Mula, Nadir Murru, and Federico Pintore. “On random sampling of supersingular elliptic curves”. In: *Cryptology ePrint Archive* (2022). URL: https://eprint.iacr.org/2022/528.

[OP22] Rémy Oudompheng and Giacomo Pope. “A note on reimplementing the Castryck-Decru attack and lessons learned for SageMath”. In: *Cryptology ePrint Archive* (2022). URL: https://eprint.iacr.org/2022/1283.

[Uen+22] Rei Ueno et al. “Curse of re-encryption: a generic power/EM analysis on post-quantum KEMs”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2022), pp. 296–322. DOI: 10.46586/tches.v2022.i1.296-322.

[Wes22] Benjamin Wesolowski. “Understanding and improving the Castryck-Decru attack on SIDH”. In: (2022). URL: https://www.bweso.com/papers.php.

[BF23] Andrea Basso and Tako Boris Fouotsa. “New sidh countermeasures for a more efficient key exchange”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 208–233. DOI: 10.1007/978-981-99-8742-9_7.

[BMP23] Andrea Basso, Luciano Maino, and Giacomo Pope. "FESTA: fast encryption from supersingular torsion attacks". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 98–126. DOI: `10.1007/978-981-99-8739-9_4`.

[Bas+23] Andrea Basso et al. "Supersingular curves you can trust". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 405–437. DOI: `10.1007/978-3-031-30617-4_14`.

[CD23] Wouter Castryck and Thomas Decru. "An efficient key recovery attack on SIDH". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 423–447. DOI: `10.1007/978-3-031-30589-4_15`.

[CV23] Wouter Castryck and Frederik Vercauteren. "A polynomial time attack on instances of M-SIDH and FESTA". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 127–156. DOI: `10.1007/978-981-99-8739-9_5`.

[CPR23] Jesús-Javier Chi-Domínguez, Amalia Pizarro-Madariaga, and Edgardo Riquelme. "Computing Quotient Groups of Smooth Order with Applications to Isogenies over Higher-Dimensional Abelian Varieties." In: *IACR Cryptol. ePrint Arch.* 2023 (2023), p. 508. URL: `https://eprint.iacr.org/2023/508`.

[Dar+23] Pierrick Dartois et al. *An Algorithmic Approach to* $(2,2)$*-isogenies in the Theta Model and Applications to Isogeny-based Cryptography.* Cryptology ePrint Archive, Paper 2023/1747. 2023. URL: `https://eprint.iacr.org/2023/1747`.

[Feo+23] Luca De Feo et al. "SCALLOP: scaling the CSI-FiSh". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2023, pp. 345–375. DOI: `10.1007/978-3-031-31368-4_13`.

[FMP23] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. "M-SIDH and MD-SIDH: countering SIDH attacks by masking information". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 282–309. DOI: `10.1007/978-3-031-30589-4_10`.

[LTZ23] Jason LeGrow, Yan Bo Ti, and Lukas Zobernig. "Supersingular Non-Superspecial Abelian Surfaces in Cryptography". In: *Mathematical Cryptology* 3.2 (2023), pp. 11–23. URL: `https://eprint.iacr.org/2022/650`.

[Mai+23] Luciano Maino et al. "A Direct Key Recovery Attack on SIDH". In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by Carmit Hazay and Martijn Stam. Cham: Springer Nature Switzerland, 2023, pp. 448–471. ISBN: 978-3-031-30589-4. DOI: `10.1007/978-3-031-30589-4_16`.

[Mor23] Tomoki Moriya. "IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram". In: *IACR Cryptol. ePrint Arch.* 2023 (2023), p. 1506. URL: `https://api.semanticscholar.org/CorpusID:263708980`.

[NO23] Kohei Nakagawa and Hiroshi Onuki. "QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras". In: *Cryptology ePrint Archive* (2023). URL: `https://eprint.iacr.org/2023/1468`.

[Rob23a] Damien Robert. "Breaking SIDH in polynomial time". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 472–503. DOI: `10.1007/978-3-031-30589-4_17`.

[Rob23b] Damien Robert. "Evaluating isogenies in polylogarithmic time". In: (2023). URL: `https://hal.science/hal-03943970v2`.

[Bas24] Andrea Basso. "POKE: A Framework for Efficient PKEs, Split KEMs, and OPRFs from Higher-dimensional Isogenies". In: *Cryptology ePrint Archive* (2024). URL: `https://eprint.iacr.org/2024/624`.

[Sag24] The Developers of Sage. *SageMath, the Sage Mathematics Software System (Version 10.1)*. 2024. URL: `https://www.sagemath.org`.