EPFL

SQIPrime & SILBE: New isogeny based cryptographic protocols

Max Duparc

School of Computer and Communication Sciences

Master Thesis

January 2024

Responsible
Prof. Serge Vaudenay
EPFL / LASEC

Supervisor
Dr. Tako Boris Fouotsa
EPFL / LASEC

LASEC

Abstract:
We present two new isogeny based cryptographic protocols: SQIPrime and SILBE. The first is a signature scheme inspired by SQISign that relies exclusively on isogenies of prime degree, while the second is an Updatable Public Key Encryption scheme based on M-SIDH and on the generalized lollipop attack. Both protocols make extensive usage of the multiple isogeny representations used in cryptography.

Introduction

The discovery in Shor's seminal paper [Sho94] of a quantum algorithm able to efficiently solve both factoring and discrete logarithm problems highlighted the security risks presented by the development of quantum computers and propelled the development of Post Quantum Cryptography. Isogeny-Based Cryptography is a relative newcomer in the field of Post-Quantum Cryptography, as although it traces its roots to Couveignes' 1997 rejected paper [Cou06], it only started gaining serious traction in the late 2000s due to its inherent compactness and seemingly heightened resistance to quantum cryptanalysis, reminiscent of Elliptic Curve Cryptography on which it builds upon. Instead of relying on scalar multiplication over elliptic curves, Isogeny-Based Cryptography employs rational maps between curves, aptly named isogenies. These isogenies are interesting as they remain efficiently computable and have many interesting structures. For example, the first efficient key exchange protocols proposed in [RS06] relied on isogenies induced commutative group action, as this commutative group action stands resilient against Shor's algorithm while preserving essential properties to perform Diffie-Hellman key exchange.
Since its inception, the field has rapidly expanded and diversified. Isogenies, echoing Henri Poincaré's maxim that "Mathematics is the art of giving the same name to different things," assume various forms — rational maps, torsion subgroups, matrices, ideals between orders of quadratic or quaternion algebra, edges in regular graphs, morphisms of lattices, and more. Leveraging this multitude of isogeny representation has been instrumental in constructing key exchange mechanisms [RS06, FJP11, CLM$^+$18, FMP23, ...], public key encryption schemes [Mor23, BMP23, NO23, ...], signature schemes [GPS16, FKL$^+$20, DLRW23, ...] or hash functions [CGL06].

In this thesis, we contribute to this evolving landscape by presenting a new isogeny-based signature scheme named SQIPrime, alongside a novel updatable public key encryption scheme named SILBE. Both leverage the versatility of the multiple isogeny representations. This thesis is organized into the following chapters:

- Chapter 1, introduces all the necessary preliminaries on elliptic curves and isogenies that we will use throughout this thesis.
- Chapter 2 presents a family of algorithms that are used to efficiently utilize isogenies under their different representations. We build upon these algorithms to construct our cryptographic protocols.
- Chapter 3 details SQIPrime, a variant of the signature scheme SQISignHD based only on prime degree isogenies.
- Chapter 4 details SILBE, an Updatable public key encryption scheme based on M-SIDH and on the generalised lollipop attacks.

As you navigate through these chapters, we wish you a pleasant and insightful reading.

Contents

Chapter 1

Mathematical background on elliptic curves

Cryptography has always had deep roots in mathematics, a connection that became particularly apparent with Public Key Encryption as the security of such cryptographic schemes is inherently tied to mathematical problems that are computationally hard in one way, such as the discrete logarithm or the factoring problem. Understanding the mathematical underpinnings of cryptographic primitives and algorithms is paramount, and this necessity is further underlined in the era of post-quantum cryptography. While the factoring problem may be relatively straightforward to comprehend[1], code-based or lattice-based cryptography already demand a more profound mathematical knowledge and experience. The same holds for isogeny-based cryptography, as it is grounded in algebraic geometry—an advanced domain of mathematics. The objective of this chapter is thus to define central concepts and highlight the core mathematical properties of elliptic curves and isogenies. These properties will serve as the bedrock upon which we will construct subsequent chapters.

Throughout this chapter, we denote K a general field and $\overline{K}$ its algebraic closure. Additionally, we also consider field such that $\text{char}(K) \neq 2, 3$.

1.1 Elliptic curves

1.1.1 Equations

Before defining the notion of elliptic curve, we need to define the notion of projective space. Although we most often work in the affine space, some properties of elliptic curves are easier to understand when seeing them as subspaces of projective spaces.

Definition 1.1.1: Projective spaces

The n-th **projective space** $\mathbb{P}^n$ is the set of equivalent classes over $\overline{K}^{n+1}\backslash\{0\}$ under the following equivalence relation.

$$(x_0, \cdots, x_n) \sim (y_0, \cdots, y_n) \iff \exists \lambda \in \overline{K}, \lambda \neq 0 \text{ such that } x_i = \lambda y_i$$

[1] which is not at all the case of its cryptanalysis.

A point of $\mathbb{P}^n$ is noted $[X_0 : \cdots : X_n]$ and the K-**rational space** $\mathbb{P}^n_K$ is the set of all $[X_0 : \cdots : X_n]$ such that $X_i \in K$ for all $i = 0, \cdots, n$.

To go from the projective space to the affine space $\overline{K}^n$ (usually noted as $\mathbb{A}^n$), we use the standard (de)homogenisation maps ϕ_i, that link $\mathbb{A}^n$ with the projective subset $U_i = \{[X_0 : \cdots : X_n] \in \mathbb{P}^n | X_i \neq 0\}$

$$\phi_i : \mathbb{A}_n \longrightarrow U_i \subset \mathbb{P}^n$$
$$\phi_i(x_1, \cdots, x_n) \longrightarrow [x_1 : \cdots, x_{i-1} : 1 : x_{i+1} : \cdots : x_n]$$
$$\left(\frac{X_0}{X_i}, \cdots, \frac{X_{i-1}}{X_i}, \frac{X_{i+1}}{X_i} \cdots, \frac{X_n}{X_i}\right) \longleftarrow \phi_i^{-1}([X_0 : \cdots, X_{i-1} : X_i : X_{i+1} : \cdots : X_n])$$

We now define elliptic curves using the Weierstrass equation.

Definition 1.1.2: Elliptic curves

An **elliptic curve** E is defined as the subset of $\mathbb{P}^2$ given by the zeros of the **Weierstrass equation**:

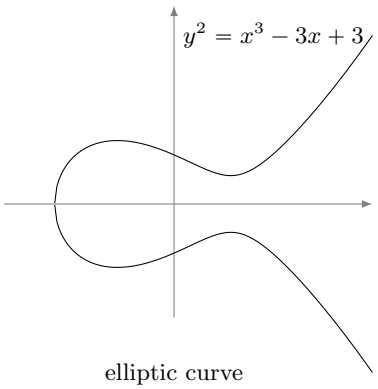
$$Y^2Z = X^3 + AXZ^2 + BZ^3$$

with $A, B \in \overline{K}$ and such that $4A^3 + 27B^2 \neq 0$.
$[0 : 1 : 0]$ is the **base point** of E.

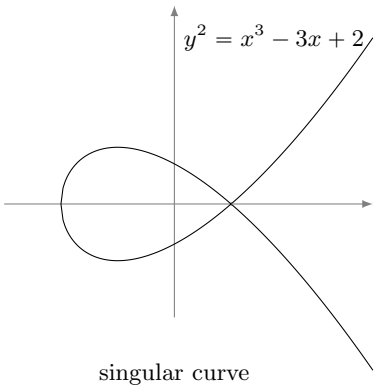
Additionally, we say that E is defined over K whenever $A, B \in K$ and denote as $E(K)$ the K-rational points of E. Note that $[0 : 1 : 0]$ is the only point of E such that $Z = 0$.
The Weierstrass equation given above describe all elliptic curves whenever $\text{char}(K) \neq 2, 3$. Using the dehomogenizing over U_z, we can also define an elliptic curve over the affine space $\mathbb{A}^2$ as follows.

$$E = \{(x, y) \in \mathbb{A}^2 | y^2 = x^3 + Ax + B\} \cup \{\infty\}$$

with ∞ the base point of E.
The reason we ask for $4A^3 + 27B^2$ to be non-zero is that we want E to be a smooth curve whose tangent space is well-defined at any points of E, which is equivalent to having $4A^3 + 27B^2 \neq 0$. Curves given by the Weierstrass equation such that $4A^3 + 27B^2 = 0$ are called singular curves.



elliptic curve



singular curve

1.1.2 Group structures

Definition 1.1.2 establishes that elliptic curves are regular projective varieties of dimension 1, like parabolas or hyperbolas. However, what distinguishes elliptic curves from other such varieties is their abelian group structure. To be more precise, elliptic curves are categorized as abelian varieties of dimension 1.

Definition 1.1.3: abelian varieties

An **abelian variety** V is an algebraic variety with:

- a **zero point** $0 \in V$.
- a **group law** morphism[a]
$$+ : V \times V \longrightarrow V$$
- an **inversion** morphism
$$i : V \to V$$

and such that $(V, 0, +, i)$ has an abelian group structure.

[a] A morphism of varieties essentially consists in a change of variable given by homogeneous rational maps of multivariable polynomials. See [Sil09, I.3] for greater details.

This definition is straightforward but it is not clear how to find an abelian group structure on elliptic curves. If we take two distinct points $P, Q \in E$, what should be $P + Q$? and how to do in such a way that this is commutative?
The answer is given by the fact that there exists a unique line that pass throw both P and Q. If we assume that this line passes through E at exactly one another point R. Then this point would be a good choice to be defined as $P + Q$. This is almost how the group law is defined as to ensure associativity, if $R = [X : Y : Z]$, then we must define $P + Q$ as its inverse $[X : -Y : Z]$. Now, what occurs when $P = Q$? As Q gets closer to P, we have that the line passing thought both points becomes the tangent of E at the point P. If this line intersects E at another point, then we could define $P + P$ similarly to how we defined $P + Q$. This later point explain why we asked for elliptic curve to be smooth.

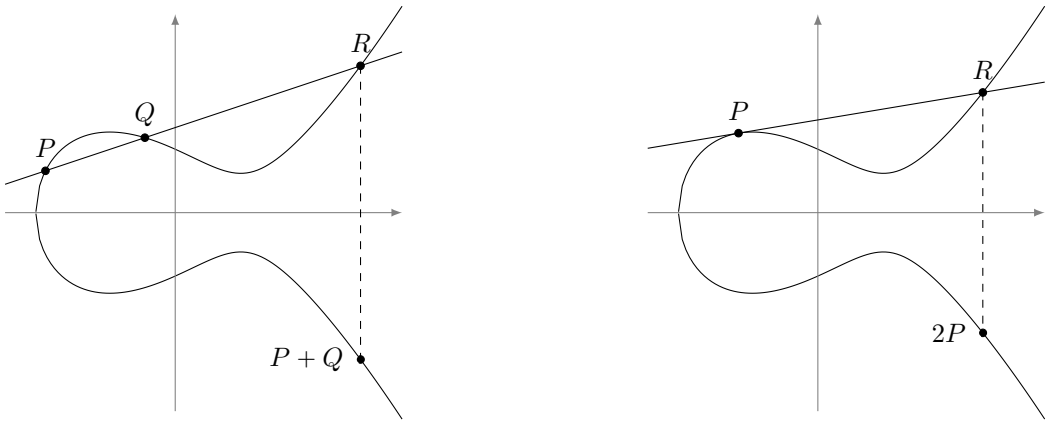


Figure 1.1: [DF17] Visualization of the group law of elliptic curve.

Now, the reason we can assume that the elliptic curve E and the line defined by P and Q will have an additional intersection points comes from Bezout's theorem.

Theorem 1.1.4: [Sha16, III.2.2.2] Bezout's theorem

Let V_1 and V_2 be two subsets of $\mathbb{P}^2$ defined as the zeros of two distinct prime homogeneous polynomials f_1 and f_2. Then, V_1 and V_2 intersect at exactly $\deg(f_1)\deg(f_2)$ points, counted with their respective multiplicities.

As elliptic curves are given by Weierstrass equations, itself a polynomial of degree 3 and that a line is of degree 1, the Bezout's theorem ensures us that there are always 3 intersection points, P, Q and R, meaning that our idea of addition is well-defined. By looking at the equation of both E and of the line defined by P and Q, we can properly define the group law and more generally the abelian group as follows.

Theorem 1.1.5: Elliptic curve group law

Any elliptic curve $E : y^2 = x^3 + Ax + B$ is an abelian variety of dimension 1 with:

- 0 as the base point ∞
- the inversion map given by $i(x, y) = (x, -y)$
- the group law given by $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ such that

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = (x_1 - x_3)\lambda - y_1$$
$$\text{with } \lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\ \frac{3x_1^2 + A}{2y_1} & \text{otherwise} \end{cases}$$

with the additional rule that $P + i(P) = 0$.

All the proof details can be found in [Sil09, section III.2] and even more as this theorem is proven for elliptic curves over field of any characteristic. This significantly complexifies the equations.
Elliptic curves are thus abelian varieties of dimension 1, often called abelian curves. They are in fact THE abelian curve, as following [Sil09, section II.3], we can show that any abelian curve of genus 1 is in fact isomorphic to an elliptic curve. This essentially comes from the fact that there is an equivalence between abelian curves and an object called the Picard group.[2] This result will be useful for the notion of dual isogeny and to understand the difference between elliptic curves and higher dimensional abelian varieties.

Theorem 1.1.6: [Sil09, III.3.4] Abel-Jacobi isomorphism

Let E be an elliptic curve. Then, the **Abel-Jacobi map**

$$\lambda : E \to \mathrm{Pic}^0(E)$$
$$\lambda(P) = [P] - [0]$$

is an isomorphism.

[2] See [Sha16, IV.1 & IV.4] for a proper definition using Weil's and Cartier's divisors.

1.1.3 j-invariant

Still, two curves given by different Weierstrass equations can be isomorphism, so we need a method to characterize elliptic curve up to isomorphism. This is why we introduce the notion of j-invariant.

Definition 1.1.7: j-invariant

Let E be an elliptic curve induced by $y^2 = x^3 + Ax + B$. The **j-invariant** of E is defined as

$$j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}$$

At first glance, the notion of j-invariant describes a link between A and B but it is not striking that it characterize isomorphism of elliptic curves, but it is the case.

Theorem 1.1.8: [Sil09, III.1.4] j-invariant theorem

Let $E : y^2 = x^3 + A_1x + B_1$ and $F : y^2 = x^3 + A_2x + B_2$ be two elliptic curves defined over K, then

$$E \cong F \iff A_2 = \mu^4 A_1 \text{ and } B_2 = \mu^6 B_1 \text{ with } \mu \in K^*$$

This induces that

$$E \cong F \Rightarrow j(E) = j(F)$$

$$j(E) = j(F) \Rightarrow E \cong F \text{ over a field L}$$

with L a K-field extension of degree:

- dividing 6 if $j(E) = 0$.
- dividing 4 if $j(E) = 1728$.
- dividing 2 for any other $j(E) \in K$.

Note that if $K = \overline{K}$, then j-invariant and isomorphism are equivalent. The j-invariant is especially interesting in cryptography as it enables us to define canonical representation of isomorphism class, i.e., fixing one Weierstrass equation among all possible to define elliptic curves of j-invariant j_0. A common canonical representation is given below but other representations are often used.

j_0	E_{j_0}
0	$y^2 = x^3 + 3$
1728	$y^2 = x^3 + x$
otherwise	$y^2 = x^3 + 3j_0(1728 - j_0)x + 2j_0(1728 - j_0)^2$

Additionally, it is relatively straightforward to construct a curve that is not isomorphic to E but with the same j-invariant. This curve is called the quadratic twist. This notion of quadratic twist is very handy to work with torsion points in finite fields, as we shall see in section 1.3.2.

Definition 1.1.9: Quadratic twist

Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve defined over K and let $d \in K$ be a not square number. the **quadratic twist** of E, noted E^d is the elliptic curve given by

$$E^d : dy^2 = x^3 + Ax + B \iff y^2 = x^3 + d^2Ax + d^3B$$

Note that $j(E^d) = j(E)$ and that E^d and E are only isomorphic in $K[\sqrt{d}]$, not K.

1.2 Isogenies

The challenge we now face is that the concept of projective morphism does not adequately consider the group structure inherent in elliptic curves. Therefore, we need to enhance our definition of morphisms for elliptic curves. These refined morphisms are called isogenies. They possess significant and foundational properties, which we will elaborate on in detail throughout this section.

1.2.1 Rational maps

We will initially introduce isogenies through their structure as morphisms of varieties, i.e. as rational maps. While this representation is heavy, it is the most natural and facilitates the definition and comprehension of several central concepts in isogenies.

Definition 1.2.1: Isogenies

Given X, Y two abelian varieties of same dimension defined over K, an **isogeny** is a map

$$\phi : X \to Y$$

such that:

- ϕ is a projective morphism defined over K.[a]
- ϕ is also a group morphism.
- $\ker(\phi)$ is finite.

Additionally, two isogenies $\phi : E \to F$ and $\psi : E' \to F'$ are **isomorphic** if there are isomorphisms $\iota : E \cong E'$ and $\kappa : F \cong F'$ defined over K such that the following diagram is commutative

$$\begin{array}{ccc} E & \xrightarrow{\phi} & F \\ \| \iota & & \| \kappa \\ E' & \xrightarrow[\psi]{} & F' \end{array}$$

[a] meaning that it is given by homogeneous rational maps of multivariable polynomial that are defined over K.

Examples 1.2.2:

- The **inverse map**
$$[-1] : E \to E$$
$$(x, y) \to (x, -y)$$
- The **[2] map**
$$[2] : E \to E$$

$$(x,y) \to \left(\frac{(3x^2+A)^2 - 8xy^2}{4y^2}, \frac{12xy^2(3x^2+A) - (3x^2+A)^3 - 8y^4}{8y^3} \right)$$

- If K is of characteristic p, then, given $E : y^2 = x^3 + Ax + B$, we define $E^{(p)} : y^2 = x^3 + A^p x + B^p$. We have that $j(E^{(p)}) = j(E)^p$. Furthermore, the **Frobenius map**

$$\pi : E \to E^{(p)}$$

$$\pi : (x,y) \to (x^p, y^p)$$

  is an isogeny.

In general, the equations that generate projective morphisms are often not practical to work with. See for example [Was08, section 3.2] for the equations of the scalar maps $[n]$, with $n \in \mathbb{Z}$. Nevertheless, due to the group preserving nature of isogenies, they exhibit a *canonical form*.

Lemma 1.2.3: [Sut15, Lemma 5.25] Canonical form of isogenies

Let $E_1 : y^2 = f_1(x)$ and $E_2 : y^2 = f_2(x)$ be two elliptic curves and let $\phi : E_1 \to E_2$ be an isogeny between the two. Then ϕ is uniquely characterized by the map

$$\phi(x,y) = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)} y \right)$$

with $u(x)$ coprime to $v(x)$ and $s(x)$ coprime $t(x)$, $v^3(x) \big| t^2(x)$ and $t^2(x) \big| v^3(x) f_1(x)$.

Some isogeny properties are linked to properties of their canonical form. This is the case for the following notions.

Definition 1.2.4: Degree and separability of isogenies

Given $\phi : E \to F$ an isogeny in canonical form $\phi(x,y) = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)} y \right)$.

- The **degree** of ϕ is given by

$$\deg \phi = \max \{ \deg u(x), \deg v(x) \}$$

- ϕ is said **separable** if

$$\left(\frac{u}{v} \right)' = \frac{u'v - v'u}{v^2} \neq 0$$

  Otherwise, it is **inseparable**

Following this definition, we have that $[-1]$ is separable and of degree 1, that the $[2]$ map is of degree 4 and separable while the Frobenius isogeny is of degree p and is *inseparable*. The Forbenius isogeny is in fact the quintessential insperable isogeny as given by the following theorem.

Theorem 1.2.5: [Sut15, Theorem 6.4] Decomposition of isogeny

Let $\phi : E \to F$ be any isogeny. ϕ can be decomposed as

$$\phi = \phi_{\mathrm{s}} \circ \pi^n$$

with ϕ_{s} a separable isogeny and π the Frobenius isogeny.

This induces that $\deg \phi = \deg(\phi_s) \cdot p^n$. This value p^n is sometimes called the *inseparable degree* $\deg_i(\phi) = p^n$. π^n is a slight abuse of notation to represent the isogeny between E and $E^{(p^n)}$ given by the composition of Frobenius isogeny. Theorem 1.2.5 has many corollaries such that it induces that all isogenies are separable if $\mathrm{char}(K) = 0$. Among all other, the following two are the bedrock of Isogeny Based Cryptography.

Corollary 1.2.6: [Sut15, Corollary 6.8] Kernel-degree connection

Let $\phi : E \to F$ be any isogeny defined over $\overline{K}$. Then

$$|\ker \phi| = \deg \phi_s$$

Corollary 1.2.7: [Sut15, Corollary 6.10] Degree of composition

Let $\phi : E \to F$ and $\psi : F \to G$ be two isogenies. Then:

$$|\ker(\psi \circ \phi)| = |\ker \psi| \cdot |\ker \phi|$$

$$\deg_i(\psi \circ \phi) = (\deg_i \psi)(\deg_i \phi)$$

$$\deg(\psi \circ \phi) = (\deg \psi)(\deg \phi)$$

1.2.2 Vélu's formulas

The canonical form characterize isogenies as rational maps, but this representation remains somewhat impractical. We would therefore like to represent isogenies in a more efficient and compact manner. We have seen in theorem 1.2.5 that isogenies are composed of a separable part and of the Frobenius isogeny. Furthermore, we have shown in corollary 1.2.6 that their degree was fully determined by their kernel, but this link between isogeny and kernel is deeper. If we just see isogenies as surjective group morphism, then the fundamental theorem of isomorphism tells us that the image curve must be isomorphic to $E/\ker(\phi)$ i.e. that ϕ is entirely determined, as a group morphism, by its domain and its kernel. This intuition can be proven right using the Vélu's formulas. We will first state the following theoretical theorem.

Theorem 1.2.8: [Sil09, III.4.12]

Let E be an elliptic curve defined over $\overline{K}$ and let G be a finite subgroup of E. Then, there exists an isogeny ϕ

$$\phi : E \to E/_G$$

with ϕ unique up to isomorphism.

Corollary 1.2.9: Prime factorisation of isogenies

Let ϕ be a separable isogeny of degree d. Seeing d as $\prod_{i=1}^{n} p_i$ with p_i prime numbers, then ϕ can be written as

$$\phi = \bigcirc_{i=1}^{n} \phi_i$$

with ϕ_i being isogenies of prime degree p_i.

Proof of Corollary 1.2.9:
This proof is done recursively over d. Consider $G = \ker(\phi)$. It is a finite abelian subgroup of E. Therefore, using Cauchy's theorem, we can find $P_1 \in G$ of order p_1. Using theorem 1.2.8, we can thus define.

$$\phi_1 : E \to E/_{\langle P_1 \rangle}$$

Then, the set $\phi_1(G)$ is a subgroup of $E/_{\langle P_1 \rangle}$ of order $d/_{p_1} = \prod_{i=2}^{n} p_i$ so we can recursively write the isogeny generated by $\phi_1(G)$ as $\bigcirc_{i=2}^{n} \phi_i$. By composing both maps, we get our desired decomposition.
□ 1.2.9

Theorem 1.2.8 is mathematically elegant but it remains somewhat theoretical. This is where Vélu's formulas come into play, significantly enhancing practicality. Vélu's formulas render theorem 1.2.8 computable by providing the following equations.

Theorem 1.2.10: [Vél71] Vélu's formulas

Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve defined over $\overline{K}$.

- Let $(x_0, 0) \in E$. Set $t = 3x_0^2 + A$ and $w = x_0 t$

$$\phi : E \to F$$

$$\phi(x, y) = \left(\frac{x^2 - x_0 x + t}{x - x_0}, \frac{(x^2 - x_0)^2 + t}{(x - x_0)^2} y \right)$$

  is a separable isogeny with $\ker \phi = \{0, (x_0, 0)\}$ and F given by $y^2 = x^3 + A'x + B'$ with $A' = A - 5t$ and $B' = B + 7w$.

- Let $G \subset E$ be a finite group of odd order. Then, for all $Q \in G$, set

$$t_Q = 3x_Q + A \qquad w_Q = 2y_Q^2 + t_Q x_Q$$

$$r(x) = x + \sum_{G \in G \setminus 0} \left(\frac{t_Q}{x - x_Q} + \frac{2y_Q^2}{(x - x_Q)^2} \right)$$

  Then,

$$\phi : E \to F$$

$$\phi(x, y) = \big(r(x), r(x)' y\big)$$

  is a separable isogeny with $\ker \phi = G$ and F given by $y^2 = x^3 + A'x + B'$ with:

$$A' = A - 5 \left(\sum_{G \in G \setminus 0} t_Q \right) \qquad B' = B + 7 \left(\sum_{G \in G \setminus 0} w_Q \right)$$

Note that if the points of G are defined over K, then ϕ is defined over K. In fact, Vélu's formulas show that, up to isomorphism, any isogeny is defined over K if and only if its kernel is a subgroup of $E(K)$.

1.2.3 Dual isogeny

Another central notion of isogenies of elliptic curves is that they induce an inverse like isogeny named the dual isogeny and defined as such

Definition 1.2.11: Dual isogeny

Let $\phi : E \to F$ be an isogeny between two elliptic curves. The **dual isogeny** of ϕ noted $\widehat{\phi}$ is an isogeny defined as such.

$$\widehat{\phi} : F \longrightarrow E$$
$$P \to \sum_{Q=\phi^{-1}(P)} [\deg_i(\phi)]Q - \sum_{Q\in\ker(\phi)} [\deg_i(\phi)]Q$$

This definition is in fact induced by the composition $F \xrightarrow{\lambda} \mathrm{Pic}^0(F) \xrightarrow{\phi^*} \mathrm{Pic}^0(E) \xrightarrow{\sum} F$, with λ the Abel-Jacobi map. Note that, thanks to the Vélu's formulas, if ϕ is defined over K, then so does $\widehat{\phi}$.

Theorem 1.2.12: [Sil09, III.6.2]

Let $\phi, \kappa : E_0 \to E_1$ and $\psi : E_1 \to E_2$ be isogenies.

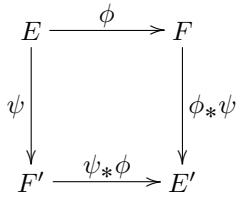
1. *Inverse*: $\widehat{\phi}$ is the unique isogeny up to isomorphism such that
$$\widehat{\phi} \circ \phi = [\deg(\phi)] \text{ and } \phi \circ \widehat{\phi} = [\deg(\phi)]$$
2. *Composition*:
$$\widehat{\phi \circ \psi} = \widehat{\psi} \circ \widehat{\phi}$$
3. *Sum*:
$$\widehat{\phi + \kappa} = \widehat{\phi} + \widehat{\kappa}$$
4. *Multiplicative maps*:
$$\widehat{[m]} = [m]$$
This implies that $\deg([m]) = m^2$.
5. *Duality*:
$$\widehat{\widehat{\phi}} = \phi$$
6. *Degree*:
$$\deg \widehat{\phi} = \deg \phi$$

Another important notion in isogeny based cryptography consists in isogeny pushforwards. They are defined using Vélu's formulas.

Definition 1.2.13: Pushforwards

Let $\phi : E \to F$ and $\psi : E \to F'$ be two isogenies of coprime degree. The **pushforward** of ψ by ϕ is the isogeny $\phi_*\psi : F \to E'$ defined by $\ker(\phi_*\psi) = \phi\big(\ker(\psi)\big)$.

Note that this definition is in line with the universal pushforward property, meaning pushforwards they are the unique isogenies up to isomorphism such that the following diagram is commutative.



1.3 Endomorphism rings

We now study the ring structure of elliptic curve endomorphisms. It is indeed central in many cryptosystems such as [CLM+18, FKL+20, ...]. This section consists only of a basic introduction, as we will delve more into details during section 1.4.2. We also detail some properties of torsion points.

1.3.1 Order

Definition 1.3.1: Endomorphism ring

Let E be any elliptic curve defined over $\overline{K}$, The **endomorphism ring** $\mathrm{End}(E)$ consists in

$$\{\varphi : E \to E \text{ an isogeny}\} \cup \{0\}$$

with 0 corresponding to the zero map and the multiplication induced by composition.
$\mathrm{End}_K(E)$ is the subring consisting of endomorphism defined over K.

We have by definition that $\mathbb{Z} \cong \{[n] | n \in \mathbb{Z}\} \subseteq \mathrm{End}(E)$. This induces that $\mathrm{End}(E)$ is of characteristic 0. The notion of degree for 0 is usually defined as 0 to remain consistent with corollary 1.2.6. The degree of 0 this can be used to show that $\mathrm{End}(E)$ is an integral ring as $\phi\psi = 0$ would be impossible if one of ϕ and ψ were not 0. In addition to the degree, endomorphisms are described by another quantity, their trace. The trace and degree are in fact sufficient to uniquely characterize any endomorphism up to duality.

Definition 1.3.2: Trace of an endomorphism

Let E be any elliptic curve and let $\mathrm{End}(E)$ be its endomorphism ring. The trace of an endomorphism $\alpha \in \mathrm{End}(E)$ is defined as

$$\mathrm{tr} : \mathrm{End}(E) \to \mathbb{Z}$$

$$\mathrm{tr}(\alpha) = \widehat{\alpha} + \alpha = \deg(\alpha + 1) - \deg\alpha - 1$$

Following theorem 1.2.12 and corollary 1.2.7, we have that $\mathrm{tr}(\alpha + \beta) = \mathrm{tr}(\alpha) + \mathrm{tr}(\beta)$ and $\deg(\alpha\beta) = \deg(\alpha)\deg(\beta)$. We now restrict our field K to be a finite field $\mathbb{F}_q$ with $q = p^n$. We then have that $E(\mathbb{F}_q)$ can be seen as $\ker(\pi^n - 1)$ as any element $x \in \mathbb{F}_q$ can be seen as an element $x \in \overline{\mathbb{F}_q}$ such that $x^q - x = 0$. Let π_E be π^n with $\pi_E \in \mathrm{End}(E)$. π_E can be used to compute the size $E(\mathbb{F}_p)$.

Theorem 1.3.3: Hasse's Theorem

Let E be any elliptic curve defined over $\mathbb{F}_q$. Then,

$$|E(\mathbb{F}_q)| = q + 1 - t$$

with $t = tr(\pi_E)$ and $|t| \leqslant 2\sqrt{q}$.

Proof of Theorem 1.3.3:
As previously explained, we have that

$$E(\mathbb{F}_q) = \left\{P \in E \text{ over } \overline{\mathbb{F}_q} | \pi_E(P) - P = 0\right\} = \ker(\pi_E - 1)$$

Furthermore, as π_E is inseparable and $[-1]$ is separable, we have that $\pi_E - 1$ is separable, meaning by corollary 1.2.6 that $\deg(\pi_E - 1) = |\ker(\pi_E - 1)| = |E|$. As $\deg(\pi_E - 1) = (\pi_E - 1)(\widehat{\pi_E - 1}) = \deg \pi_E - \mathrm{tr}(\pi_E) + 1$, we get the equation

$$|E(\mathbb{F}_q)| = q + 1 - \mathrm{tr}(\pi_E)$$

Now, let us show that $|\mathrm{tr}(\pi_E)| \leqslant 2\sqrt{t}$. To do so, consider $a, b \in \mathbb{Z} \times \mathbb{Z}^*$. The endomorphism $a\pi_E - b$ has degree

$$\begin{aligned}
\deg(a\pi_E - b) &= (a\pi_E - b)(\widehat{\pi_E}\widehat{a} - \widehat{b}) \\
&= a\pi_E\widehat{\pi_E}\widehat{a} - a\pi_E\widehat{b} - b\widehat{\pi_E}\widehat{a} + b\widehat{b}) \\
&= a^2 \deg(\pi_E) - a\pi_E b - b\widehat{\pi_E}a + b^2 \\
&= a^2 q - ab\mathrm{tr}(\pi_E) + b^2
\end{aligned}$$

As $\deg(a\pi_E - b) \geqslant 0$, we have that

$$0 \leqslant \left(\frac{a}{b}\right)^2 q + \left(\frac{a}{b}\right)\mathrm{tr}(\pi_E) + 1$$

i.e. that the for all values of $\nu \in \mathbb{Q}$, $q\nu^2 - \nu\mathrm{tr}(\pi_E) + 1 \geqslant 0$. As $\mathbb{Q}$ is dense in $\mathbb{R}$, this means that the determinant of the polynomial is strictly smaller than 0 i.e, that

$$\mathrm{tr}(\pi_E)^2 - 4q \leqslant 0 \iff |\mathrm{tr}(\pi_E)| \leqslant 2\sqrt{q}$$

□ 1.3.3

Furthermore, the bound of $|\mathrm{tr}(\pi_E)| \leqslant 2\sqrt{q}$ is tight, as we can find elliptic curve of any trace inside this bound.

Corollary 1.3.4:

Let E be any elliptic curve defined over $\mathbb{F}_q$ and let E^d be its quadratic twist. Then

$$\mathrm{tr}(\pi_{E^d}) = -\mathrm{tr}(\pi_E)$$

Proof of Corollary 1.3.4:

Inside $\mathbb{F}_q$ the product of two non-square numbers is a square. Therefore, for any value $x \in \mathbb{F}_q$, we either have that $x^3 + Ax + B$ has a root, meaning that x defines two points in E, or that it has not, and it then defines two points in E^d. Using Hasse's Theorem, we get that

$$2q = 2|\mathbb{F}_q| = |E(q)| + |E^d(q)| - 2 = 2q - \mathrm{tr}(\pi_E) - \mathrm{tr}(\pi_{E^d})$$

proving this corollary. □ 1.3.4

Now that we have introduced the trace of endomorphims and its immediate properties, we see it strongly restricts the possible form of $\mathrm{End}(E)$, as it can only be an order of very specific algebras. Let us first define the notion of order.

Definition 1.3.5: Order

Let $\mathcal{A}$ be a K-algebra of finite dimension and of characteristic 0. An **order** of $\mathcal{A}$, denoted $\mathcal{O}$, is a strict subring of characteristic 0 such that $\mathcal{O} \otimes_{\mathbb{Z}} K = \mathcal{A}$. An order is furthermore said to be **maximal** if it is not contained in another order.

Among all algebras, we are particularly interested in the following two families:

- **Quadratic fields** $\mathbb{Q}(\sqrt{d})$: A 2-dimensional $\mathbb{Q}$-algebra with basis $(1, \alpha)$ and

$$\alpha^2 = d$$

  If $d > 0$, it is a **real quadratic field**, otherwise, it is a **complex quadratic field**.

- **Quaternion algebras** $\mathbf{B}_{p,\infty}$[3]: A 4 dimensional $\mathbb{Q}$-algebra with basis $(1, \alpha, \beta, \alpha\beta)$ and

$$\alpha^2 = a \qquad \beta^2 = -p \qquad \alpha\beta = -\beta\alpha$$

Theorem 1.3.6: [Sil09, III.9.3]

Let E be an elliptic curve. Then,

- $\mathrm{End}(E) \cong \mathbb{Z}$ if $\mathrm{char}(K) = 0$.
- $\mathrm{End}(E)$ is an order of a complex quadratic field $\mathbb{Q}(\sqrt{d})$
- $\mathrm{End}(E)$ is an order of a quaternion algebra $\mathbf{B}_{p,\infty}$

This theorem can also be refined over $\mathrm{End}_K(E)$ when $K = \mathbb{F}_q$. For any curve E defined over $\mathbb{F}_q$ such that $\pi_E \notin \mathbb{Z}$, we can prove that $\mathrm{End}_{\mathbb{F}_q}(E)$ is an order of $\mathbb{Q}[\sqrt{d}]$, with $d = \mathrm{tr}(\pi_E) - 4q$. See [Sut15, theorem 14.6] for a proof.

1.3.2 Torsion points

To close this section, we will discuss torsion points. Following Vélu's formulas, we have that isogenies are given by their kernels, finite subgroups of elliptic curves. Furthermore, following theorem 1.2.12, we have that the kernel of isogeny of degree d are subgroups of the kernel of the scalar endomorphism $[d]$, i.e. they are subgroup of the torsion subgroups.

[3]This notation is inherited from ramification theory. Indeed, we are especially interested in quaternion algebra that are ramified at points p and ∞. See [Voi21, chapter 14] for more details.

Definition 1.3.7: n-torsion subgroup

let E be an elliptic curve, the **n-torsion subgroup** $E[n]$ is given as

$$E[n] = \ker\left([n]\right)$$

Theorem 1.3.8: [Sut15, Theorem 7.1]

Let p the characteristic of K and q be a prime number.

$$E[q^\ell] = \begin{cases} \mathbb{Z}_p^\ell \text{ or } 0 & \text{if } p = q \\ \mathbb{Z}_{q^\ell} \times \mathbb{Z}_{q^\ell} & \text{otherwise} \end{cases}$$

Note that using the Chinese remainder theorem, theorem 1.3.8 characterizes the group structure of $E[N]$ for any $N \in \mathbb{N}^*$. The group structure of $E[N]$ in conjunction with Vélu's formulas tells us that, up to isomorphism, there are exactly $\ell + 1$ isogenies of prime degree ℓ, as there exists only $\ell + 1$ subgroups of order ℓ in $\mathbb{Z}_\ell^2$. If we set P, Q a basis of $E[\ell]$, we can characterize all these groups as follows.

$$\langle P\rangle, \langle P + Q\rangle, \langle P + 2Q\rangle, \cdots, \langle P + (\ell - 1)Q\rangle, \langle Q\rangle$$

Furthermore, any isogeny $\phi : E \to F$ restricted over torsion points define a $\mathbb{Z}_N$-linear applications $\phi|_{E[N]} : E[N] \to F[N]$, meaning that we can also represent the action of isogenies over torsion points as matrices of dimension 2.
This representation is especially useful when correlated with the Cauchy interpolation theorem [Bie53], as it enables easy characterization of isogenies. To do so, we use the fact that the map $\sqrt{\deg(-)}$ defines a norm over the space of all isogenies between two elliptic curves E and F. This is because the degree map is positive definite quadratic form [Sil09, V.1.2].

Corollary 1.3.9:

Let $\phi, \psi : E \to F$ be two isogenies of maximal degree m and let N be an integer such that $N \geqslant 2\sqrt{m} + 1$. Then

$$\phi|_{E[N]} = \psi|_{E[N]} \iff \phi = \psi$$

Proof of Corollary 1.3.9: Assume that $\phi \neq \psi$. Then, using the triangular inequality over ϕ and ψ, we get that

$$\deg(\phi - \psi) \leqslant \left(\sqrt{\deg(\phi)} + \sqrt{\deg(\psi)}\right)^2 \leqslant 4m$$

meaning using corollary 1.2.6 that $|\ker(\phi - \psi)| \leqslant 4m$ but $\phi|_{E[N]} = \psi|_{E[N]}$ implies that $E[N] \subseteq \ker(\phi - \psi)$ and thus that $4\sqrt{m} + 1 \leqslant 0$, a contradiction. □ 1.3.9

Finally, to finish this section, we will have a small word about elliptic curve pairing. Pairing are bilinear and non-degenerative maps between curves to finite subgroup. Among all pairing, we will essentially use the **Weil's pairing** e_N as defined in [Sil09, III.8], but other pairing exists and are used in cryptography such as the Tate-Lichtenbaum pairings.[Was08, 3.4]

1.4 Supersingularity

We have seen in theorem 1.3.8 that there are two types of elliptic curves, those such that $E[p] = \mathbb{Z}_p$ and those such that $E[p] = 0$. Similarly, we also saw in theorem 1.3.6 that they were curves whose

endomorphism ring was an order of an imaginary field and those whose endomorphism ring was an order of a quaternion algebra. It occurs that those separations are in fact characteristic of the same notion, the supersingularity.

1.4.1 Group structure

Definition 1.4.1: Ordinary/Supersingular curves

Let E be an elliptic curve over K, with $p = \text{char}(K)$.

- E is **ordinary** if $E[p] = \mathbb{Z}_p$
- E is **supersingular** if $E[p] = 0$

A central observation is that supersingularity is preserved by isogenies. This means that due to their group preserving structure, for any isogeny $\phi : E \to F$, F is supersingular if and only if E is supersingular.

Proposition 1.4.2:

Let E an elliptic curve defined over K.

- If E is supersingular, then $j(E) \in \mathbb{F}_{p^2}$.
- E is supersingular and is defined over $\mathbb{F}_{p^n} \iff \text{tr}(\pi_E) = 0 \mod p$.

Proof of Proposition 1.4.2:

- Using theorem 1.3.8 and corollary 1.2.7, we have that

$$E[p] = 0 \iff [p] = \iota \circ \pi^2 \iff \widehat{\pi} = \iota \circ \pi$$

with $\iota : \pi^2(E) \cong E$ an isomorphism, meaning that $j(E) = j(\pi^2(E))$ and therefore that

$$j(E) = j\big(\pi^2(E)\big) = j(E)^{p^2}$$

Thus, $j(E) \in \mathbb{F}_{p^2}$.

- We note that $\text{tr}(\pi_E)$ is inseparable if and only if $\widehat{\pi_E}$ inseparable if and only if $\widehat{\pi}$ is inseparable, i.e. if and only if E is supersingular. Then
  - If $\text{tr}(\pi_E)$ is inseparable, then, using corollary 1.2.7, p cannot be coprime to $\text{tr}(\pi_E)$. Thus, $\text{tr}(\pi_E) = 0 \mod p$
  - If $\text{tr}(\pi_E) = 0 \mod p$, then $\text{tr}(\pi_E) = [k] \circ [p] = [k] \circ \widehat{\pi} \circ \pi$, meaning that it is separable.

□ 1.4.2

Proposition 1.4.2 tells us that supersingular curves are rare. Indeed, the first point gives us that they are finitely many supersingular curves up to isomorphism and the second point gives us that they are restricted to only a handful of possible traces. Nevertheless, those strong restrictions enable us to precisely characterize supersingular curve in both number and group structure.

Theorem 1.4.3: Number of supersingular curves

$$\left|\left\{j \in \mathbb{F}_{p^2} | E_j \text{ is supersingular}\right\}\right| = \left\lfloor \frac{p}{12} \right\rfloor + \begin{cases} 2 & p = 11 \mod 12 \\ 1 & p = 5, 7 \mod 12 \\ 0 & p = 1 \mod 12 \end{cases}$$

$$\left|\left\{j \in \mathbb{F}_{p} | E_j \text{ is supersingular}\right\}\right| = \Theta(\sqrt{p})$$

A full proof using Hasse invariant is given in [Sil09, IV 4.1]. The additional points of the first equation are in fact given by $j(E) = 1728$ if $p = 3 \mod 4$ and $j(E) = 0$ if $p = 2 \mod 3$. Following theorem 1.1.8, both of these curves have several additional properties.

Theorem 1.4.4: Group structure of supersingular curves

Let E be a supersingular curve defined over $\mathbb{F}_{p^2}$ [a]. Let $E(\mathbb{F}_q)$ be the $\mathbb{F}_q$-rational points with $q = p^n$.

- If n is odd, then $\text{tr}(\pi_E) = 0$ and
$$E(\mathbb{F}_q) \cong \mathbb{Z}_{q+1}$$
- If n is even, then
  - $\text{tr}(\pi_E) = +2\sqrt{q}$ and
  $$E(\mathbb{F}_q) \cong \mathbb{Z}_{\sqrt{q}-1} \times \mathbb{Z}_{\sqrt{q}-1}$$
  - $\text{tr}(\pi_E) = -2\sqrt{q}$ and
  $$E(\mathbb{F}_q) \cong \mathbb{Z}_{\sqrt{q}+1} \times \mathbb{Z}_{\sqrt{q}+1}$$

[a] This ensures that we do not consider the special twists of $j = 0$ and $j = 1728$.

There are different group structures when we consider $p = 2, 3$ or when we look at the special twists of $j = 1728$ and $j = 0$. See [AAM18, MVO91] for a complete list and proof. Supersingular curves and isogenies can be represented as graphs.

Definition 1.4.5: Supersingular isogeny graphs

Given p and l two prime number. We define the **ℓ-th supersingular isogeny graphs** $\mathcal{G}_p^\ell$ as a graph with

- vertices corresponding to the supersingular j-invariant in $\mathbb{F}_p^2$.
- edges corresponding to isogenies of degree ℓ up to isomorphism.

Following theorem 1.3.8, we have that $\mathcal{G}_p^\ell$ are $\ell+1$-regular graph and are in fact Ramanujan graphs [Piz90]. Those graphs have many applications in theoretical computer science, as expander graphs, thanks to their pseudo-randomness properties.

1.4.2 Deuring correspondence

Now that we have properly defined supersingularity, we now go back to endomorphism ring and see that supersingularity is intrinsically linked with quaternions. In fact, Deuring proved in [Deu41] that both supersingular curves and their isogenies were equivalent to maximal order and linking integral ideals of quaternion algebras.

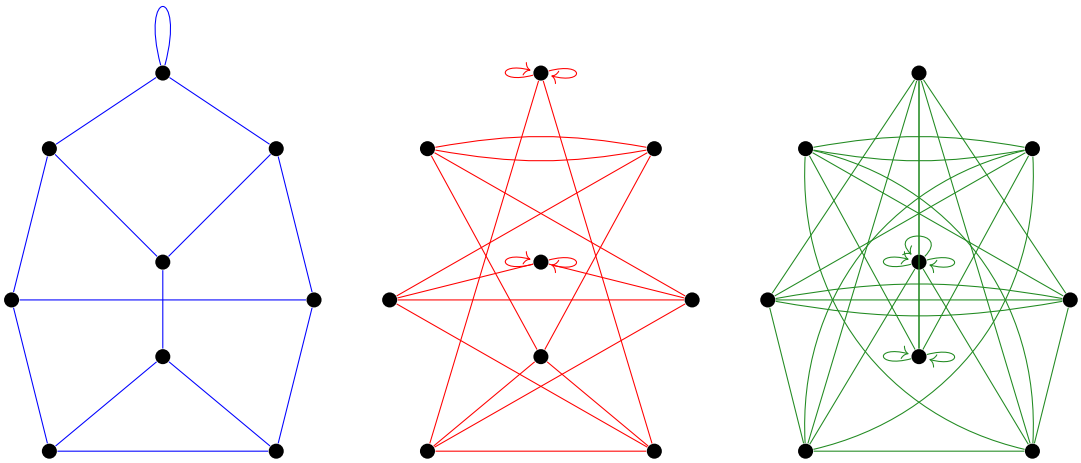


Figure 1.2: Representation of $\mathcal{G}_{109}^2$, $\mathcal{G}_{109}^3$ and $\mathcal{G}_{109}^5$

Theorem 1.4.6: [Voi21, Theorem 42.1.9]

Let E be a supersingular curve and let $\mathcal{O}_E \cong \mathrm{End}(E)$ be the corresponding order of a quaternion algebra $\mathbf{B}_{p,\infty}$. Then:

$$\mathcal{O}_E \text{ is a maximal order of } \mathbf{B}_{p,\infty}$$

with $\mathbf{B}_{p,\infty}$ given by $b^2 = -p$ and $a = \begin{cases} -1 & p = 3 \mod 4 \\ -2 & p = 5 \mod 8 \\ -q & p = 1 \mod 8 \end{cases}$ with q a prime such that $\left(\frac{-q}{p}\right) = -1$.

Example 1.4.7: Endomorphism ring of $j(E) = 1728$

Let $p = 3 \mod 4$, then the curve E_{1728} is supersingular and its endomorphism ring correspond to the maximal order $\mathcal{O}_{1728}$:

$$\mathcal{O}_{1728} = \mathbb{Z} + i\mathbb{Z} + \frac{i+j}{2}\mathbb{Z} + \frac{1+ij}{2}\mathbb{Z}$$

with $i : (x, y) \to (-x, \sqrt{-1}y)$ and $j = \pi$ the Frobenius endomorphism.

The algebra $\mathbf{B}_{p,\infty}$ is unique up to isomorphism. We can thus see supersingular curves as maximal orders of $\mathbf{B}_{p,\infty}$. We now need an oriented algebraic object that links two maximum order of $\mathbf{B}_{p,\infty}$, similarly to what isogenies do. This object is the notion of integral ideals.

Definition 1.4.8: Integral ideals

Let $\mathbf{B}_{p,\infty}$ be a quaternion algebra. Let $a_1, \cdots, a_4 \in \mathbf{B}_{p,\infty}$ be linearly independent elements. $I = \langle a_1, \cdots, a_4 \rangle$ is a **fractional ideal**.

- **the norm of I**:

$$n(I) = \gcd\Big(\{n(\alpha) | \alpha \in I\}\Big)$$

- **The left/right order of I:**

$$\mathcal{O}_L(I) = \{\alpha \in \mathbf{B}_{p,\infty} | \alpha I \subseteq I\}$$

$$\mathcal{O}_R(I) = \{\alpha \in \mathbf{B}_{p,\infty} | I\alpha \subseteq I\}$$

  I is denoted as an $(\mathcal{O}_L(I), \mathcal{O}_R(I))$-ideal

- We say that I is **integral** if $I \subseteq \mathcal{O}_L(I)$.

We see that order of $\mathbf{B}_{p,\infty}$ are just integral ideals that hold the unit element, i.e. $1 \in I$. Integral rings links maximal orders using the following proposition

Proposition 1.4.9

Let I, J be two integral ideals.

- Both $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$ are maximal orders.
- $$\mathcal{O}_L(IJ) = \mathcal{O}_L(I) \text{ and } \mathcal{O}_R(IJ) = \mathcal{O}_R(J)$$
- $$\mathcal{O}_L(\overline{I}) = \mathcal{O}_R(I) \text{ and } \mathcal{O}_R(\overline{I}) = \mathcal{O}_L(I)$$

The proof of the first point can be found in [Voi21, 10.4.2], while the others points are straightforward consequences of the definition.
Coming back to our elliptic curves, we can go from integral ideals to isogenies and reversely using the following tools.

Definition 1.4.10:

- Let $\phi : E \to F$ be an isogeny between two supersingular curves. Let $\mathcal{O}_E$ and $\mathcal{O}_F$ be the maximal orders of $\mathbf{B}_{p,\infty}$ corresponding to $\mathrm{End}(E)$ and $\mathrm{End}(F)$. The **kernel ideal** of ϕ is defined as
$$I_\phi = \left\{\alpha \in \mathcal{O}_E \middle| \ \alpha(\ker(\phi)) = 0\right\}$$
- Conversely, given I an $(\mathcal{O}_E, \mathcal{O}_F)$-ideal, it induces an isogeny $\phi_I : E \to F$ given by
$$\ker \phi_I = E[I] = \left\{P \in E \ \middle| \ \alpha(P) = 0 \ \forall \alpha \in I\right\}$$

In both cases, we have that, up to isomorphism, $I_{\phi_I} = I$ and $\phi_{I_\phi} = \phi$. Deuring showed in [Deu41][4] that this transformation induced a contravariant equivalence.

[4] See [Voi21, section 42.2 & 42.3] for a detailed proof.

supersingular j-invariants over $\mathbb{F}_{p^2}$	**maximal orders in $\mathbf{B}_{p,\infty}$**
E	$\mathcal{O}_E$
$\phi \circ \psi$	$I_\psi I_\phi$
$\deg(\phi)$	$n(I_\phi)$
$\widehat{\phi}$	$\overline{I_\phi}$
$\psi_* \phi$	$[I_\psi]_* I_\phi = \frac{1}{n(I_\psi)} \overline{I_\psi}(I_\psi \cap I_\phi)$
$\gamma \in \mathrm{End}(E)$	$\mathcal{O}_E \gamma$

Furthermore, a property that we will often use is the fact that for any $\phi : E \to F$ an isogeny such that $\phi = \rho \circ \tau$ with $q = \deg \tau$ coprime to $\deg \rho$, then as $\ker(\tau) = \ker(\phi) \cap E[q]$, we get that

$$I_\tau = I_\phi + \mathcal{O}_E q$$

This overview of the Deuring correspondence concludes this chapter. Among the important fields of elliptic curves that we did not discuss is the notion of Complex Multiplication (CM). For the interested, a good reference is [Sil94, chapter 2]. CM and volcanology is the basis of an important part of Isogeny Based Cryptography, based on the group action of the endomorphism ring over $\mathcal{G}_p^\ell$. See for example [RS06, CLM+18, BKV19, ...]. Speaking of cryptography, to manipulate all theses mathematical objects, we need efficient algorithms which is the subject of our next chapter.

Chapter 2

Toolbox: isogeny representations

Thanks to the previous chapter, we are now familiar with the mathematical properties of isogenies, but we omitted their algorithmic aspects. Indeed, as we desire to design isogeny based cryptosystems, we need standard tools and algorithms. This necessity is further accentuated by the multiple representations of isogenies that we saw in chapter 1, as separable isogenies can be represented up to isomorphism as rational maps (lemma 1.2.3), kernels (theorem 1.2.10), matrices (corollary 1.3.9) or as ideals (section 1.4.2). However, before delving into the intricacies of these representations, it is essential to establish a clear definition of what we mean by the term "representation."

Definition 2.0.1: Efficient isogeny representation

Let $\phi : E \to E'$ be an isogeny defined over $\mathbb{F}_q$. An **efficient representation** of ϕ is given by a couple $(D, \mathcal{A})$ with

- D some data of size $\mathsf{poly}\big(\log(\deg(\phi)), \log(q)\big)$ that define uniquely the isogeny ϕ.
- $\mathcal{A}$ an universal algorithm independent of ϕ that, on input P returns $\phi(P)$ with $P \in E(\mathbb{F}_{q^k})$ in $\mathsf{poly}\big(k\log(q), \log(\deg\phi)\big)$.

An efficient representation that can just compute points P of order coprime to N is called an **N-efficient representation**.

Examples of standard efficient isogeny representation are:

- $\pi : E \to E^{(p)}$ the Frobenius morphism as given by its rational maps.
- $[n] : E \to E$ all scalar maps with $n \in \mathbb{Z}$ using, for example, Montgomery method [HM21, section 3.3].

Both of theses efficient representations are based on rational maps, but this does not scale to all isogenies as following lemma 1.2.3, the canonical representation of a separable isogeny needs $O\big(\deg(\phi)\log(q)\big)$ space, which is not in line with definition 2.0.1. In this chapter, we will discuss three different efficient isogeny representations.

- The kernel representation, arguably the most widely used in Isogeny-Based Cryptography due to its compactness.
- The ideal representation, that is inherited from the Deuring correspondence[1].

[1] and more generally, from complex multiplication, but all our scheme will be based over quaternion algebras.

- The HD representation, that is based on matrices and on higher-dimensional isogenies.

2.1 Kernel representation

The first and arguably main representation of isogenies that is used in cryptography is the kernel representation. It makes full usage of theorem 1.2.8. The idea is to represent an isogeny $\phi : E \to E'$ of degree d as a point $K \in E[d]$ such that $\langle K \rangle = \ker(\phi)$. To compute $\phi(P)$, we just use Vélu's formulas but this method is not an effective representation of ϕ as Vélu's formulas are in $O(\deg \phi)$ and not in $O\big(\log(\deg \phi)\big)$. The trick is to use corollary 1.2.9. Instead of using Vélu's formulas once, it is far more efficient to use Vélu's formulas over the decomposition of $\phi = \bigcirc_{i=1}^{m} \phi_i$ with ϕ_i isogenies of degree p_i such that $\deg \phi = \prod_{i=1}^{m} p_i$. This can be done using the following algorithm.

Algorithm 1 KernelToIsogeny

Input: E the domain curve, K a generator of $\ker(\phi)$ and $d = \prod_{i=1}^{m} p_i$ the degree of ϕ.
Output: ϕ the isogeny, F the codomain of ϕ.

1: Set $E^0 = E$, $K^0 = K$
2: **For** $j = 1$ to m:
3: $\quad \phi_j, E^j \longleftarrow$ Vélu's formulas$\big(E_{j-1}, [d/\prod_{i>j} p_i]K^{j-1}, p_j\big)$
4: $\quad K^j \longleftarrow \phi_j(K^{j-1})$
5: **return** $\bigcirc_{i=1}^{m} \phi_j$, E_m.

$\bigcirc_{i=1}^{m} \phi_j$ is a slight abuse of notation, as it does not compute the composition of all ϕ_i which would be a wasteful operation. Instead, it sends all distinct ϕ_i separately that are then applied sequentially when evaluating.
Let $\deg(\phi) = d$ be B-smooth, meaning that all prime factors of d are smaller than B, then **KernelToIsogeny** returns an evaluation of ϕ that is in $O(\log_B(d)B) = O\big(B \log(d) \log(B)\big)$ field operations[2]. This induces that if d is $\tilde{O}(\log(d))$ smooth, then **KernelToIsogeny** is an efficient evaluation algorithm. We can thus define the kernel representation as such.

Definition 2.1.1: Kernel representation

Let $\phi : E \to E'$ be a cyclic[a] isogeny of *smooth* degree d. Its **kernel representation** consists in:

- $K \in E[d]$ such that $\langle K \rangle = \ker(\phi)$.
- The **KernelToIsogeny** algorithm.

[a] i.e. its kernel is a cyclic group.

It is important to say that the algorithm **KernelToIsogeny** that we presented here is by no means optimal, and many acceleration mechanisms have been developed.

1. [FJP11, section 4.2.2] proposed to use computational strategy based on discrete equilateral triangle to not perform wasteful multiplication and additions.
2. [BFLS20] proposed $\sqrt{\text{élu}}$, a speedup that enabled computation of the Vélu's formulas in $\tilde{O}(\sqrt{\ell})$. Due to some hidden constant, this speed-up is only significant for primes greater than 100.

[2] Note that this computation does not consider the computational cost of adding and multiplying over elliptic curves.

3. Finally, other interesting speedup method consists in using other parameterization of elliptical curves, such as Montgomery curves [OM21].

Using kernel representation, we can thus easily compute smooth isogenies between two curves. On the contrary, finding an isogeny between two curves is believed to be hard. This gap induces the central problems in Isogeny Based Cryptography. We give here its restriction to supersingular curves.

Problem 2.1.2: Isogeny walk problem

Given E, E' two supersingular curves, find $\phi : E \to E'$ an isogeny of smooth degree.

The term walk comes from the fact that one can see an isogeny of degree ℓ^α as a walk over the graph $\mathcal{G}_p^\ell$. In some cases, the degree of the linking isogeny is known. This induces the following problem.

Problem 2.1.3: Explicit isogeny problem

Given E, E' two curves isogenenous of degree d, compute $\phi : E \to E'$ of degree d.

The best known generic algorithm [DFHPS16] that solve explicit isogeny problem is in $O(d^2)$. A question nevertheless remains. How can we ensure that our torsion points and thus our kernels are available in $\mathbb{F}_{p^{2k}}$, with $k = \tilde{O}(\log p)$ a small power?

2.1.1 Accessible torsion points

To answer that question, we will make use of the supersingularity and more especially of the group structure of $E(\mathbb{F}_{p^2})$, as given by theorem 1.4.4. The following point is one of the reason we often use supersingular curves when working in Isogenies Based Cryptography.
First, we have that all supersingular curves E are defined over $\mathbb{F}_{p^2}$. Thus, if we assume that $\mathrm{tr}(\pi_E) = 2\sqrt{p}$, then via theorem 1.4.4 we have that

$$E(\mathbb{F}_{p^2}) = \mathbb{Z}_{p-1}^2 = E[p-1]$$

Furthermore, using proposition 1.3.4, we have that the quadratic twist of E, E^d is such that $\mathrm{tr}(\pi_E) = -2\sqrt{p}$ and thus that

$$E^d(\mathbb{F}_{p^2}) = \mathbb{Z}_{p+1}^2 = E[p+1]$$

Then, by using theorem 1.1.8, we can map the points in $E^d[p+1]$ to points in $E(\mathbb{F}_{p^4})$ in such a way that the x-coordinate remains in $\mathbb{F}_{p^2}$. This therefore means that we have the guaranty that all torsion points whose order divide p^2-1 are easy to access. This ease of access is extremely valuable in Cryptography as we can choose prime number p such that the desired torsion points are divisors of p^2-1 and thus ensure that they are easy to access. This is the basis for the following algorithm.

CanonicalTorsionBasis Using those easy to access torsion points, the **CanonicalTorsionBasis** efficiently computes a basis $\langle P, Q\rangle = E[N]$. To do so, it simply samples points at random in $E(\mathbb{F}_{p^2})$ or $E^d(\mathbb{F}_{p^2})$. To ensure that this method is deterministic, the sampling is performed deterministically. There exists many algorithms to find torsion basis, depending on the cases. A good example in the general case is [MMRV09], while [ZJP+17] is very efficient for large power of 2 or 3.
Let's now give an example to see how the kernel representation can be used in cryptography with SIDH.

2.1.2 SIDH

Initially proposed in [FJP11], SIDH[3] is an isogeny based key exchange mechanism. The main idea behind SIDH is to use the universal property of pushforwards. Indeed, if Alice and Bob compute ϕ_A and ϕ_B two isogenies of coprime degree with the same domain, then they can compute their respective pushforwards, namely $(\phi_A)_*\phi_B$ and $(\phi_B)_*\phi_A$ and gain a shared secret, the codomain of both pushforwards.
Let the SIDH public parameter be as follows:

- $p = \ell_A^{e_A}\ell_B^{e_B} f - 1$ a prime number with ℓ_A and ℓ_B coprime and $\ell_A^{e_A} \approx \ell_B^{e_B}$.
- E a supersingular curve defined over $\mathbb{F}_{p^2}$.
- $\langle P_A, Q_A \rangle$ a basis of $E_0[\ell_A^{e_A}]$
- $\langle P_B, Q_B \rangle$ a basis of $E_0[\ell_B^{e_B}]$.

SIDH

Alice(pp)		**Bob**(pp)
$s_A \leftarrow_\$ \mathbb{Z}_{\ell_A^{e_A}}$		$s_B \leftarrow_\$ \mathbb{Z}_{\ell_B^{e_B}}$
$R_A \leftarrow P_A + [s_A]Q_A$		$R_B \leftarrow P_B + [s_B]Q_B$
$\phi_A, E_A \leftarrow$ **KernelToIsogeny**$(E, R_A, \ell_A^{e_A})$		$\phi_B, E_B \leftarrow$ **KernelToIsogeny**$(E, R_B, \ell_B^{e_B})$
$S_A \leftarrow \phi_A(P_B), T_A \leftarrow \phi_A(Q_B)$		$S_B \leftarrow \phi_B(P_A), T_B \leftarrow \phi_B(Q_A)$
	$\xrightarrow{E_A, S_A, T_A}$	
	$\xleftarrow{E_B, S_B, T_B}$	
$U_A \leftarrow S_B + [s_A]T_B$		$U_B \leftarrow S_A + [s_B]T_A$
$\psi_A, E_K \leftarrow$ **KernelToIsogeny**$(E_B, U_A, \ell_A^{e_A})$		$\psi_B, E_K \leftarrow$ **KernelToIsogeny**$(E, U_B, \ell_B^{e_B})$
$K \leftarrow \mathsf{KDF}(j(E_K))$		$K \leftarrow \mathsf{KDF}(j(E_K))$

This protocol is correct because

$$\ker \psi_A = \langle \phi_B(P_A) + [s_A]\phi_B(Q_A) \rangle = \langle \phi_B(P_A + [s_A]Q_A) \rangle = \phi_B\big(\ker(\phi_A)\big) = \ker\big([\phi_B]_*\phi_A\big)$$

$$\ker \psi_B = \langle \phi_A(P_B) + [s_B]\phi_A(Q_B) \rangle = \langle \phi_A(P_B + [s_B]Q_B) \rangle = \phi_A\big(\ker(\phi_B)\big) = \ker\big([\phi_A]_*\phi_B\big)$$

Therefore, due to definition 1.2.13, the following diagram is commutative

$$\begin{array}{ccc} E & \xrightarrow{\phi_B} & E_B \\ {\scriptstyle \phi_A}\downarrow & & \downarrow{\scriptstyle \psi_A} \\ E_A & \xrightarrow{\psi_B} & E_K \end{array}$$

SIDH has many advantages. It is simple to understand, easy to implement and has very small key size. Its key security reduces to the following variant of the explicit isogeny problem.

[3] Supersingular Isogeny Diffie-Hellman

Problem 2.1.4: Supersingular isogeny problem with torsion point information

Let $\phi : E \to E'$ be an isogeny of degree d between supersingular curves and let $\langle P, Q \rangle = E[N]$ with N coprime to d.
Given $P, Q, \phi(P), \phi(Q)$, retrieve ϕ.

It is important for N and d to be coprime, as otherwise, we would trivially gain partial knowledge of the kernel of ϕ. For example, if we know $\phi(E[d])$, then as $\phi(E[d]) = \ker \widehat{\phi}$, we can compute $\widehat{\phi}$, evaluate $\widehat{\phi}(E'[d])$ and thus retrieve $\ker(\phi)$.
Since SIDH was the underlying architecture basis for SIKE, a candidate to the NIST Post-Quantum cryptography standardization effort, its security analysis was widely studied. Some important works are [GPST16, FP21] that proposed adaptative attacks with one dishonest party. [Pet17] proposed the idea of *lollipop attacks* that is efficient over unbalanced SIDH, meaning that $\ell_A^{E_A} \gg \ell_B^{E_B}$. Those attacks were further improved in [dQKL+20]. We will detail them more precisely in chapter 4. Finally, it was proven in [CD23, MMP+23, Rob22a] that supersingular isogeny problem with torsion point information was easy using higher dimension isogenies and high dimension representation of isogenies, as we shall see in section 2.3. Countermeasures have been proposed [FMP23, BF23, ...] but they all come with a huge overhead.

2.2 Ideal representation

Using the Deuring correspondence, we know that an isogeny can be represented as an integral ideal of $\mathbf{B}_{p,\infty}$ linking maximal orders of $\mathbf{B}_{p,\infty}$. Finding which order correspond to which curve is believed to be hard. This problem is in fact a central problem in Isogeny Based Cryptography.

Problem 2.2.1: Endomorphism problem

Let E be any supersingular curve defined over $\mathbb{F}_{p^2}$, find a nontrivial[a] endomorphism of E.

[a] i.e. not $\alpha \in \mathbb{Z}$

It was proven in [PW23] that the endomorphism problem was equivalent to the problem of retrieving a full basis of $\mathcal{O}_E$. It was also proven in [Wes22] that the endomorphism problem and the isogeny walk problem are equivalent.

2.2.1 Endomorphism basis

An important point to see is that if we have an efficient representation for $\alpha_1, \cdots, \alpha_4$ a basis of $\mathcal{O}_E$, then we can construct an efficient representation for all $\gamma \in \mathrm{End}(E)$, as $\gamma = \sum_{i=1}^{4} [a_i]\alpha_i$. This central property is the motivation for the notion of evaluation basis.

Definition 2.2.2: Evaluation Basis

Let $\mathcal{O}_E$ be a maximal order of $\mathbf{B}_{p,\infty}$, we define an **evaluation basis** of $\mathcal{O}_E$, denoted $\mathfrak{O}_E$ as:

- A basis $\alpha_1, \cdots, \alpha_4$ such that
$$\mathcal{O}_E = \alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z}$$
- An isomorphism $\delta : \mathrm{End}(E) \cong \mathcal{O}_E$ such that any $\delta^{-1}(\alpha_i)$ has an efficient evaluation.

An evaluation basis that can just compute points P of order coprime to N is called an N-**evaluation basis**.

Apart from $\mathcal{O}_0$ and $\mathcal{O}_{1728}$, the latter given in example 1.4.7, finding ex nihilo an evaluation basis over a curve E is believed to be hard. The best method to compute evaluation basis is to use isogenies.

PushEndRing

We present here the **PushEndRing**, as given in [DLRW23, Algorithm 8]. If we know $\mathfrak{O}_E$ an evaluation basis of $\mathrm{End}(E)$ together with an isogeny $\varphi : E \to F$ and its kernel ideal I_φ, we can then construct an evaluation basis of $\mathrm{End}(F)$. The main idea is to see that for any $\theta \in \mathrm{End}(F)$, $\widehat{\varphi} \circ \theta \circ \varphi \in \mathrm{End}(E)$. This makes the map

$$\iota : \mathrm{End}(F) \to \mathbf{B}_{p,\infty}$$
$$\iota(\theta) \to \frac{1}{d}\delta(\widehat{\varphi} \circ \theta \circ \varphi)$$

an injective morphism with $\iota\big(\mathrm{End}(F)\big) = {}^1\!/_d\overline{I_\varphi} I_\varphi = \mathcal{O}_F$.

Algorithm 2 PushEndRing

Input: $\mathfrak{O}_E = \big(\{\alpha_i\}_{i=1}^4, \delta\big)$ an evaluation basis of $\mathrm{End}(E)$, $\varphi : E \to F$ an isogeny of degree d with an efficient representation together with its ideal I_φ.
Output: $\mathfrak{O}_F$ a d-representation basis of $\mathrm{End}(F)$.

1: Find $\{\beta_i\}_{i=1}^4$ the basis of the order $\mathcal{O}_F = {}^1\!/_d\overline{I_\varphi} I_\varphi$.
2: Find $\{c_{i,j}\}_{i,j=1}^4$ such that $d\beta_i = \sum_{j=1}^4 c_{i,j}\alpha_j$
3: Set $\epsilon : \mathcal{O}_F \to \mathrm{End}(F)$ as $\beta_i \to {}^1\!/_d \sum_{j=1}^4 [c_{i,j}](\widehat{\varphi} \circ \delta(\alpha_j) \circ \varphi)$
4: **return** $\mathfrak{O}_F = \big(\{\beta_i\}_{i=1}^4, \epsilon^{-1}\big)$

It is thus possible, knowing an isogeny between E_{1728} and E to find an evaluation basis of E. This method is useful and efficient, but what to do if we have to evaluate a point whose order is not coprime to d and how can we find both an isogeny and its ideal?
To remedy the first point, we just have to find two isogenies of coprime degree. There are many methods to obtain this result that we can engulf as **DoublePath** algorithms.

DoublePath

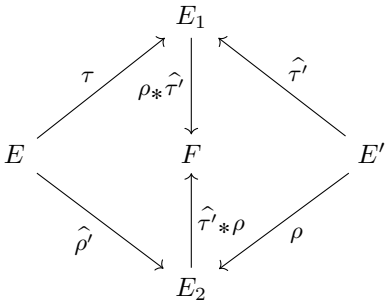
The goal of the a **DoublePath** algorithm is to use the knowledge of $\mathcal{O}_E$ the endomorphism ring structure of E to construct two coprime isogenies $\phi, \psi : E \to F$ together with their ideals. We give here a presentation of the **DoublePath** as presented in [DLRW23, Algorithm 1].[4] To do so, it uses both pushforwards and the factorization of isogenies, given by corollary 1.2.9.
Take $\theta \in \mathrm{End}(E)$ an endomorphism of order A^2B^2 with A and B both smooth and coprime number. Then, we can write θ as

$$\theta = \rho \circ \rho' \circ \tau' \circ \tau$$

with $\deg(\rho) = \deg(\rho') = B$ and $\deg(\tau) = \deg(\tau') = A$. Then, we consider the following commutative

[4] But it is not the only one, as other variants exists, that uses for example the **KLPT**.

diagram



and define $\phi = (\rho_*\widehat{\tau'}) \circ \tau$ and $\psi = (\widehat{\tau'}_*\rho) \circ \widehat{\rho'}$, two isogenies between E and F of respective degree A^2 and B^2. Furthermore, by factoring θ and by using **KernelToIdeal** we can evaluate all these functions and pushforwards.

2.2.2 Ideals representation

We now answer our second point and see that provided $\mathfrak{O}_E$ an evaluation basis of $\mathrm{End}(E)$, then we can easily find the representing ideal of $\phi : E \to F$, provided that its degree is smooth. This is done using the **KernelToIdeal** algorithm.

KernelToIdeal

We present here **KernelToIdeal** as given by [DLRW23, Algorithm 9]. The main idea is to construct an endomorphism $\gamma \in \mathrm{End}(E)$ such that γ factors through ϕ, meaning that $\gamma(K) = 0$, with $\langle K \rangle = \ker(\phi)$. To do so, we simply find a linear combination between the component of the evaluation basis $\{\beta_i\}_{i=1}^4$ such that it maps K to 0.

Algorithm 3 KernelToIdeal

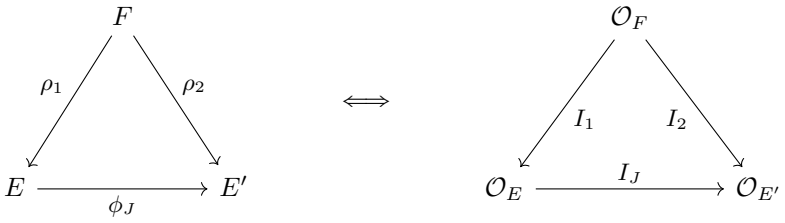
Input: $\mathfrak{O}_E$ an N-evaluation basis over E and K the kernel of an isogeny ϕ of smooth degree d coprime to N
Output: I_ϕ the ideal such that $E[I_\phi] = \langle P \rangle$

1: For each basis component β_i, compute $Q_i \leftarrow \delta(\beta_i)(P)$
2: Find $i \neq j$ such that $\langle Q_i, Q_j \rangle = E[d]$ ▷ use discrete log
3: Take $k \neq i, j$ and find a, b such that $Q_k = aQ_i + bQ_j$. ▷ also use discrete log
4: Define $\gamma = \beta_k - a\beta_i - b\beta_j$
5: **return** $\mathcal{O}_E\gamma + \mathcal{O}_E d$

Note that the smoothness is required to efficiently perform discrete logarithms. We thus retrieve the representing ideal for smooth degree isogenies. This can be used in conjunction with the kernel representation to evaluate the isogeny described by an ideal J without any smoothness requirement on $n(J)$ using the **EvalTorsion** algorithm.

EvalTorsion

We present here the **EvalTorsion** as given in [DLRW23, Algorithm 11]. Assume knowledge of $\mathfrak{O}_F$ an evaluation basis over F and two isogenies $\rho_1 : F \to E$ and $\rho_2 : F \to E'$ of norm d_1 and d_2 with efficient representations together with their respective ideals I_1 and I_2. Consider J an $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal of norm N coprime to d_1 and d_2. Finally, let P be any point in E. We are thus in the following diagram



We then have that $I_1 J \overline{I_2}$ describe $\gamma = \widehat{\rho_2} \circ \phi_J \circ \rho_1$ an endomorphism of F. We thus get the following equality

$$\phi_J(P) = [(d_1 d_2)^{-1}]\rho_2 \circ \gamma \circ \widehat{\rho_1}(P) \mod N$$

Algorithm 4 EvalTorsion

Input: $\mathfrak{O}_F$ an evaluation basis over F, $\rho_1 : F \to E$ of degree $d_1, \rho_2 : F \to E'$ of degree d_2, both with efficient representations. J an $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal of norm N coprime to d_1 and d_2. $P \in E$
Output: $\phi_J(P)$

1: Compute an efficient representation of $\widehat{\rho_1}$ ▷ Doable with all representation in this thesis.
2: Find γ such that $\mathcal{O}_F \gamma = I_1 J \overline{I_2}$
3: Compute $R = \rho_2 \circ \delta^{-1}(\gamma) \circ \widehat{\rho_1}(P)$
4: Compute $\mu = (d_1 d_2)^{-1} \mod N$
5: **return** $[\mu]R$.

Using **EvalTorsion**, we can finally define the ideal representation of an isogeny.

Definition 2.2.3: Ideal representation

Let $\phi : E \to E'$ be an isogeny of degree d. Its **ideal representation** consists in:

- I_ϕ the $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal corresponding to ϕ, with $\mathfrak{O}_F$ an evaluation basis of $\mathrm{End}(F)$, $\rho_1 : F \to E$ and $\rho_2 : F \to E'$ two isogenies with efficient representations of respective degree d_1, d_2 and with corresponding ideals I_1 and I_2.
- The **EvalTorsion** algorithm.

We have that an ideal representation is an $d_1 d_2$-efficient representation.

2.2.3 KLPT

To conclude this section, we will present several algorithms that are linked to the **KLPT** algorithm. Most of the following is taken from Leroux's thesis [Ler22]. To simplify, the **KLPT** is an algorithm that transforms an $(\mathcal{O}_E, \mathcal{O}_F)$-ideal I into another $(\mathcal{O}_E, \mathcal{O}_F)$-ideal J whose norm is smooth. The **KLPT** utilize the following algorithms.

FullRepresentInteger

Given in [Ler22, Algorithm 4], the **FullRepresentInteger** take as input a number $N > p$ and return $\gamma \in \mathcal{O}_{1728}$ an endomorphism of E_{1728} such that $\gamma\overline{\gamma} = N$. To do so, it uses a modification of the

Cornacchia algorithm[5], named the **CornacchiaExtended** [Ler22, Algorithm 1] that does not require knowledge of the factorization of N but at the cost of some bias over the distributions of its answers. The **FullRepresentInteger** has the following properties.

Lemma 2.2.4: [Ler22, Lemma 3.1.4] + [FLLW22, section 6]

- **FullRepresentInteger** runs in $\mathsf{poly}(\log(N))$.
- Due to **CornacchiaExtended**, we can only output $\Theta(1/\log(N))$ of all possible endomorphisms of norm N.
- Under plausible heuristic assumptions, the distribution of γ as an output of **FullRepresentInteger** is computationally indistinguishable from the uniform distribution among all endomorphism of E_{1728} of degree N.

FullRepresentInteger is extremely useful when combined with the **DoublePath** as they can be used to compute evaluation basis over random curves E.

Proposition 2.2.5: [DLRW23, section 5.2]

Under plausible heuristic assumptions and provided $A^2 \simeq B^2 \simeq p$, then the following distributions are computationally indistinguishable.

- F the codomain of ϕ and ψ outputted by **DoublePath**(γ, A, B) with γ given by **FullRepresentInteger**(A^2B^2).
- F a random curve sampled uniformly among all supersingular curves.

RandomEquivalentIdeal

Given in [Ler22, Algorithm 6], the **RandomEquivalentIdeal** takes as input an $(\mathcal{O}_E, \mathcal{O}_F)$-ideal I and returns J another $(\mathcal{O}_E, \mathcal{O}_F)$-ideal such that $n(J)$ is a "small" prime. It is done by applying the **LLL** algorithm ([LLL82]) over the relative norm $n_I(\alpha) = \frac{n(\alpha)}{n(I)}$ to find a Minkowski reduced basis $\{\beta_i\}_{i=1}^4$ of $\mathcal{O}_L(I)$. It then samples at random $\{c_i\}_{i=1}^4 \in [-b, b]$, check primality and return $I\overline{\frac{\sum c_i\beta_i}{n(I)}}$.
The **RandomEquivalentIdeal** has the following properties.

Lemma 2.2.6: [Ler22, Lemma 3.2.3 & 3.2.4]

- As c_i are sample randomly, the output of **RandomEquivalentIdeal** has an uniform distribution among all small linking ideals.
- **RandomEquivalentIdeal** runs in $\mathsf{poly}\,(n(I)pC)$ with C a bound depending on the basis of $\mathcal{O}_E$.
- Given J outputted by **RandomEquivalentIdeal**. Then, under probable heuristic assumptions and with probability greater that $1 - 2^{-\epsilon}$,

$$\sqrt{\frac{p}{\log p}}\epsilon^{-1} \leqslant n(J) \leqslant \sqrt{p\log(p)}\epsilon$$

[5]Defined in [Cor07], the **Cornacchia** algorithm solves efficiently equations of the form $x^2 + qy^2 = N$ with $x, y \in \mathbb{Z}$ provided that we know the factorisation of N.

meaning that **RandomEquivalentIdeal** outputs inside this range with negligible probability provided $\epsilon = \sqrt{\log(p)}$.

KLPT

By combining **RandomEquivalentIdeal** together with **FullRepresentInteger** and with the addition of a third algorithm named the **FullStrongApproximation**, get the KLPT. [Ler22, Algorithm 7]. The KLPT outputs $\mathcal{O}_{1728}$-left ideal J whose norm has the desired divisors. It can therefore be used to compute ideals with smooth norms. This choice of smoothness comes at the cost of having a big norm. In general, one gets that $n(J) = O(p^{7/2})$, but tradeoffs exists. For example, [PS18] shortened the length of the isogeny to $O(p^{5/2})$ at the cost of a greater running time, while [FLLW22] lifted the necessity to have a left $\mathcal{O}_{1728}$-ideal, at the cost of a norm in $O(p^{11/2})$.

2.3 High dimension representation

To put it simply, the idea behind the high dimensional representation lies in embedding isogenies between elliptic curves into higher dimension isogenies. The idea of higher dimensional representation was proposed in [Rob22b] and is an adaptation of [MMP+23, Rob22a] attacks on SIDH. HD representation is based on something named the Kani's Lemma.

2.3.1 Kani's Lemma

Before explaining Kani's Lemma, we need to give a bit of background on high dimensional abelian varieties. Properly explaining the inner working of abelian varieties would easily require another dedicated chapter, so we will just introduce the main differences between elliptic curves and abelian varieties that will be useful to us. For the interested, we recommend the reading of [Mil86].
Let V be a n-dimensional abelian variety defined over $\overline{\mathbb{F}_p}$. Note that definition 1.2.1 of isogeny holds. The same goes for Theorem 1.2.8 that can be generalized using quotient varieties. Then,

- $V[N] \cong \mathbb{Z}_N^{2n}$ for N coprime to p. (which is in line with elliptic curves as they are 1 dimensional varieties.) To remain consistent with isogenies over elliptic curves, we define the degree of a separable[6] isogeny ϕ as $\sqrt[n]{|\ker(\phi)|}$. Thus, $\deg([m]) = m^2$ for any dimension.

- Any isogeny $\phi : V \to W$ induces via the pullback ϕ^* a dual isogeny $\widehat{\phi} : W^\vee \to V^\vee$ with $V^\vee = \mathrm{Pic}^0(V)$ the **dual variety**. The separation between V and $V^\vee$ is important, as there are no equivalents to Abel-Jacobi map, meaning that V and $V^\vee$ are usually *not* isomorphic. One then define an isogeny $\lambda : V \to V^\vee$ as a **polarization** and we write such system (V, λ).

- If λ is an isomorphism, then we say that (V, λ) is a **principally polarized variety**. Given (V, λ) and (W, μ) two principally oriented varieties, we say that $\phi : V \to W$ is a **polarized isogeny** if

$$\phi^\vee \circ \mu \circ \phi = [\deg(\phi)]\lambda$$

[6] Separability over higher dimension isogeny $\phi : V \to W$ is linked with the notion of separability of the field extension K(V) over K(W). This definition agrees with our definition in dim 1 1.2.4 but is heavily rooted in algebraic geometry.

We are then in the following diagram

$$\begin{array}{ccc} V & \xrightarrow{\phi} & W \\ {\scriptstyle\lambda}\Vert & & \Vert{\scriptstyle\mu} \\ V^\vee & \xleftarrow{\phi^\vee} & W^\vee \end{array}$$

and therefore define the **polarized dual** of ϕ, denoted $\widetilde{\phi}$ as $\mu \circ \phi^\vee \circ \lambda^{-1}$. This definition is then in line with the definition 1.2.11 and holds similar properties as in proposition 1.2.12.

As we will from now on only work with principally polarized variety, we will omit the notation of polarization.

Lemma 2.3.1: [Kan97] Kani's Lemma

Let A, B, A', B' be principally polarized abelian varieties with f, g, f' and g' polarized separable isogenies such that the following diagram is commutative:

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle g'} \\ A' & \xrightarrow{f'} & B' \end{array}$$

Then, the following map

$$F : B \times A' \to A \times B'$$
$$\begin{pmatrix} \tilde{f} & -\tilde{g} \\ g' & f' \end{pmatrix}$$

is a polarised separable isogeny with $D = \deg(F) = \deg(f) + \deg(g) = D_1 + D_2$ and such that

$$\ker(F) = \left\{ \big(f(P), -g(P)\big) \,\middle|\, P \in A[D] \right\} = \left\{ \big(\widetilde{g'}(P), \widetilde{f'}(P)\big) \,\middle|\, P \in B'[D] \right\}$$

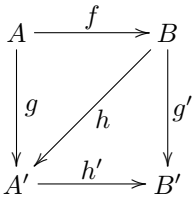
$$\ker(\widetilde{F}) = \left\{ \big(\tilde{f}(P), g'(P)\big) \,\middle|\, P \in B[D] \right\} = \left\{ \big(-\tilde{g}(P), f'(P)\big) \,\middle|\, P \in A'[D] \right\}$$

We recommend the proof in [Rob22a, section 3] that is both complete and easy to understand.
Although we presented here the Kani's Lemma under its canonical form, we can also use is as follows.

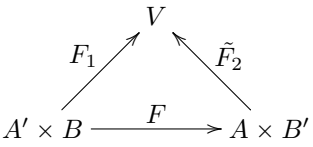
Corollary 2.3.2:

1. If $\ker(f) \cap \ker(g) = \emptyset$, then consider $h = g \circ \widetilde{f}$. We can also write $\ker(F)$ as

$$\left\{ \big([D_1](P), h(P)\big) \,\middle|\, P \in B[D] \right\} = \left\{ \big(\tilde{h}(P), [D_2](P)\big) \,\middle|\, P \in A'[D] \right\}$$



2. Given $D = d_1 d_2$, then we can write $F = F_2 \circ F_1$ with $\deg F_1 = d_1$ and $\deg F_2 = d_2$ such that



$$\ker(F_1) = \ker(F)[d_1] = \left\{ \left(f(P), g(P)\right) \middle| \ P \in A[d_1] \right\}$$
$$\ker(\widetilde{F_2}) = \ker(\widetilde{F})[d_2] = \left\{ \left(\tilde{f}(P), g'(P)\right) \middle| \ P \in B[d_2] \right\}$$

2.3.2 Computing with Kani's Lemma

Let us now explain how we can use Kani's Lemma to represent isogenies.

HDKernelToIsogeny

It is apparent that to efficiently uses Kani's isogeny, we need to use a kernel representation for high-dimensional isogenies. We will use the representation given by the evaluation algorithm [Rob10, Algorithm 7.2.4]. This algorithm is based on [Rob10, Algorithm 7.3.2], an analog of **KernelToIsogeny** for high-dimensional isogenies that we will therefore name **HDKernelToIsogeny**. We take this algorithm in a black box manner, as it relies on θ-functions. For the interested, see [Rob10, Part I] together with [Mum66].
Similarly to kernel representation, the **HDKernelToIsogeny** algorithm enables efficient representation of HD isogenies of *smooth* degree d. More specifically, if ϕ is an isogeny of dimension n that is B-smooth, then **HDKernelToIsogeny** would return an evaluation in $O\left(\log(d) B^n \log(B)\right)$. Likewise to [FJP11, section 4.2.2], we can also use optimized strategies to not perform wasteful multiplication. See [DLRW23, section F.1] for more details.

EvalKani

To evaluate isogenies using Kani's Lemma, we will construct **EvalKani**. It uses Kani together with the Zahrin's trick to evaluate an isogeny $\varphi : E_1 \to E_2$ of degree d over a point $R \in E$ when given $P, Q, \varphi(P)$ and $\varphi(Q)$ with $\langle P, Q \rangle = E_1[N]$ and N smooth, coprime with d and such that $N \geqslant \sqrt{d}$. Let N_1, N_2 be divisors of N. We will use isogenies in dimension 2, 4 or 8 depending on the value of $a = N_1 N_2 - d$, with N_1 and N_2 divisors of N. There are three cases:

1. If $a = a_1^2$, then we use Kani in dimension 2 over $[a_1]$ and φ.
2. If $a = a_1^2 + a_2^2$ (which occurs for $a = 3 \mod 4$ prime) then we use Kani in dimension 4 over the

following functions

$$\alpha_i := \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \in \mathrm{End}(E_i^2) \text{ with } \deg(\alpha) = \sum_{i=1}^{4} a_i^2; \Sigma := \mathrm{Diag}(\varphi, \varphi)$$

3. Otherwise $a = a_1^2 + a_2^2 + a_3^2 + a_4^2$ and we are force to use Kani in dimension 8 over the following functions

$$\alpha_i := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix} \in \mathrm{End}(E_i^4) \text{ with } \deg(\alpha) = \sum_{i=1}^{4} a_i^2; \Sigma := \mathrm{Diag}(\varphi, \varphi, \varphi, \varphi)$$

1)
$$\begin{array}{ccc} E_1 & \xrightarrow{\varphi} & E_2 \\ {\scriptstyle [a_1]}\downarrow & & \downarrow{\scriptstyle [a_1]} \\ E_1 & \xrightarrow{\varphi} & E_2 \end{array}$$

2)
$$\begin{array}{ccc} E_1^2 & \xrightarrow{\Sigma} & E_2^2 \\ {\scriptstyle \alpha_1}\downarrow & & \downarrow{\scriptstyle \alpha_2} \\ E_1^2 & \xrightarrow{\Sigma} & E_2^2 \end{array}$$

3)
$$\begin{array}{ccc} E_1^4 & \xrightarrow{\Sigma} & E_2^4 \\ {\scriptstyle \alpha_1}\downarrow & & \downarrow{\scriptstyle \alpha_2} \\ E_1^4 & \xrightarrow{\Sigma} & E_2^4 \end{array}$$

In each case, we construct F, the Kani's isogeny of degree N_1N_2. A good choice is of N_1 and N_2 is thus important as it may enable us to work in dimension 4 or in dimension 2, which significantly improve the efficiency of **EvalKani**.
To evaluate F, we split F into $F = F_2 \circ F_1$ with $\deg F_i = N_i$ using the second point of corollary 2.3.2. As we already know a basis of $E[N]$, together with its image by ϕ, we also have basis of $E[N_1]$ and $E[N_2]$ together their images through φ. We can thus construct a basis of $E_1[N]^k$denoted $\{P_{i,j}\}_{0 \leqslant i,j \leqslant 2,k}$ with j the position and i the choice of P or Q. Using corollary 2.3.2, we can compute B_1 and B_2, the respective basis of the kernel of F_1 and $\widetilde{F_2}$. Using **HDKernelToIsogeny**, we can therefore retrieve F_1 and $\widetilde{F_2}$.
We then have to compute F_2. There are 2 methods, depending on the order of R, the point we want to evaluate.

1. If R is not smooth, then we have to compute F_2. To do so, we simply compute $\ker(F_2) = \widetilde{F_2}\left(0^k \times E_2[N_2]^k\right)$. As we know a basis of $E_2[N_2]$ using $\phi(E[N])$, we can find B_3 a basis of $\ker(F_2)$ by evaluating $\widetilde{F_2}$ at $2k$ points. We then make a third call to **HDKernelToIsogeny** to retrieve F_2. It then suffices to write R in one of the components of F to retrieve $\varphi(R)$.

2. If R is smooth, then we can do without computing F_2. To do so, let r be the degree of R. We first use **CanonicalTorsionBasis** to find a basis of $E_1[r]$ and $E_2[r]$. We then construct B_0 a basis of $(E_1 \times E_2)^k[r]$ and evaluate that basis through both F_1 and $\widetilde{F_2}$ (meaning $4k$ evaluations). Using discrete logarithms, we compute the matrix $\mathbf{M} \in M_{2k}(\mathbb{Z}_r)$ such that

$$F_1(\mathsf{B}_0) = \mathbf{M}\widetilde{F_2}(\mathsf{B}_0)$$

knowing that matrix enables us to compute any values of R as for any vector $v \in (E_1 \times E_2)^k[r]$

$$F(v) = F_2 \circ F_1(v) = F_2 \circ \mathbf{M}\widetilde{F_2}(v) = \mathbf{M}\widetilde{F_2} \circ F_2(v) = [d_2]\mathbf{M}v$$

We then evaluate $\varphi(R)$ using F with the same method as in the first point. More details on this method are available in [DLRW23, section F.3].

We only write the pseudocode for dimension 8. It is easy to modify the algorithm for dimension 4 and 2, as the main difference is that instead of using the **EHR** algorithm[7], we use **Cornacchia**. We also here present the case where we compute 3 HD isogenies. Finally, we define a subroutine, called **ConstructKani** that construct the Kani's isogeny.
Note that **EvalKani** works perfectly if we set $N_2 = 1$. In that case, we will compute F is one go. Nevertheless, it is always a good idea to split the Kani's isogenies because computing F_1 and $\widetilde{F_2}$ can be parallelized, which significantly speed up the computations.

Algorithm 5 ConstructKani

Input: $\varphi : E_1 \to E_2$ an isogeny of degree d with N_1, N_2 divisors of N. $\langle P, Q \rangle = E_1[N]$ with $S = \varphi(P), T = \varphi(Q)$.
Output: F the Kani's isogeny

$k_1 \leftarrow N/N_1$, $k_2 \leftarrow N/N_2$
$a_1, a_2, a_3, a_4 \leftarrow$ **EHR**$(N_1N_2 - d)$ ▷ Use **Cornacchia**$(1, N_1N_2 - d)$ if dimension 4
Construct α and $\widetilde{\alpha}$
Compute $\{P_{i,j}\}_{0 \leqslant i,j \leqslant 2,4}$ a basis of $E_1^4[N]$ ▷ Using P, Q
$\mathsf{B}_1 \leftarrow \left\{\left([k_1]\Sigma(P_{i,j}), [-k_1]\alpha(P_{i,j})\right)\right\}_{0 \leqslant i,j \leqslant 2,4}$ ▷ $\Sigma(P_{i,j})$ computed using S, T
$\mathsf{B}_2 \leftarrow \left\{\left([-k_2]\widetilde{\alpha}(P_{i,j}), [k_2]\Sigma(P_{i,j})\right)\right\}_{0 \leqslant i,j \leqslant 2,4}$
$F_1 \leftarrow$ **HDKernelToIsogeny**(B_1)
$\widetilde{F_2} \leftarrow$ **HDKernelToIsogeny**(B_2)
$\mathsf{B}_3 \longleftarrow \left\{\widetilde{F}_2\left(0^4 \times [k_2]\Sigma(P_{i,j})\right)\right\}_{0 \leqslant i,j \leqslant 2,4}$
$F_2 \longleftarrow$ **HDKernelToIsogeny**(B_3)
return $F_2 \circ F_1$

Algorithm 6 EvalKani

Input: $\varphi : E_1 \to E_2$ an isogeny of degree d with N_1, N_2 divisors of N. $\langle P, Q \rangle = E_1[N]$ with $S = \varphi(P), T = \varphi(Q)$ and $R \in E_1$
Output: $\varphi(R)$

$F \longleftarrow$ **ConstructKani**$\left(d, N_1, N_2, P, Q, S, T\right)$
return $\Big(F(0, 0, 0, 0, R, 0, 0, 0)\Big)_5$

Using **EvalKani**, we can define the notion of high dimensional representation as follows.

Definition 2.3.3: High dimensional representation

Let $\phi : E \to E'$ be an isogeny of degree d, its **high dimensional representation** consists in:

- $\big(P, Q, \phi(P), \phi(Q)\big)$ with $\langle P, Q \rangle = E[N]$, N *smooth* and coprime with d such that $N \geqslant \sqrt{d}$.
- the **EvalKani** algorithm.

We see that, using HD representation, we can solve the supersingular isogeny problem with torsion point information, provided that N is smooth. This is essentially how the attacks of [MMP+23, Rob22a] works. More generally, HD representation enables new cryptosystems such as [BMP23, DLRW23, Mor23, ...] or

[7] Initially proposed in [RS86] and improved by [PT18] that efficiently solve Legendre 4 squares problem.

our new cryptographic protocols, SQIPrime and SILBE, the subject of the following chapters.

Chapter 3

SQIPrime: SQISignHD with highly two addic primes

We now present SQIPrime a post-quantum digital signature scheme based on the Deuring correspondence. Comparatively to its inspirations SQISign [FKL+20] and SQISignHD [DLRW23], SQIPrime further-expand the use of Kani's Lemma initially introduced in SQISignHD for verification to both key generation and commitment. Through these modifications, it gains the following properties:

- SQIPrime uses highly two addic base prime numbers.
- All isogenies used in SQIPrime have big prime degree.

This chapter is structured as such. Section 3.1 explains the main ideas and architecture behind both SQISign and SQISignHD. Section 3.2 details the new tools that we used in SQIPrime. Finally, section 3.3 and 3.4 give the detailed construction and security analysis of SQIPrime.

3.1 SQISign & SQISignHD

One of the main interests of isogeny based signature schemes is that they provide compact post-quantum signatures. This property, which comes at the cost of a greater computational cost, fuelled their research. Among the early propositions ([YAJ+17, BKV19, ...]) was GPS [GPS16] that relied on the Deuring correspondence. Its ideas were expended and improved in [FKL+20] to create the SQISign protocol. As of today, SQISign is the only isogeny based candidate at the NIST post-quantum cryptography standardization effort. In 2023, [DLRW23] proposed SQISignHD, a variant utilizing HD representation for verification.
Both SQISign and SQISignHD are in fact identification schemes, and more precisely Σ protocol based identification schemes. They are then transformed into signatures schemes using the Fiat-Shamir transform [FS86]. An identification protocol is defined as such.

Definition 3.1.1: Identification schemes

Let λ be a security parameter, an **identification scheme** is given by set of 3 $\mathsf{PPT}(\lambda)$ algorithms $\mathsf{KeyGen}, \mathsf{P}, \mathsf{V}$ together with a setup algorithm $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ the public parameters.

- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{pk})$ a secret/public key pair.
- P, V are an interactive protocol such that for all $(\mathsf{sk}, \mathsf{pk})$:

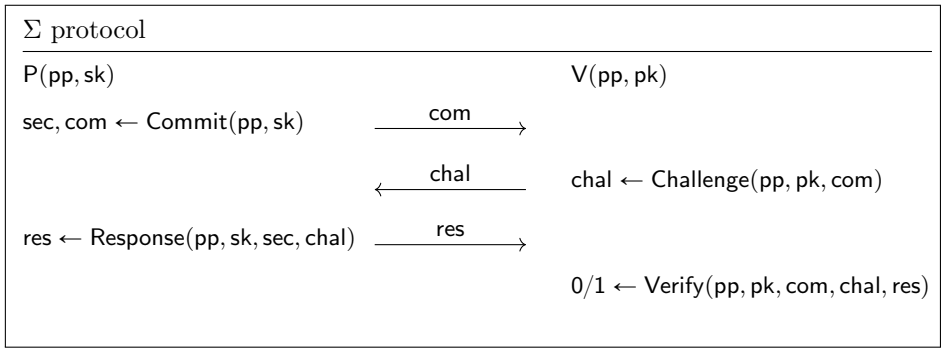
– *Correctness*:
$$\mathbb{P}\Big[\mathsf{Output}\big(\mathsf{P}(\mathsf{pp},\mathsf{sk}) \longleftrightarrow \mathsf{V}(\mathsf{pp},\mathsf{pk})\big) = 1\Big] = 1$$

– *Soundness*: For any $\overline{P}$, an interactive $\mathsf{PPT}(\lambda)$, then
$$\mathbb{P}\Big[\mathsf{Output}\big(\overline{\mathsf{P}}(\mathsf{pp},\mathsf{pk}) \longleftrightarrow \mathsf{V}(\mathsf{pp},\mathsf{pk})\big) = 1\Big] \leqslant \mathsf{negl}(\lambda)$$

Note that the definition of an identification scheme allows for the prover to just send the secret key to the verifier. If it is hard to retrieve sk from pk, then we would have a valid identification scheme. This is why it is often asked for identification schemes to be *zero-knowledge*, which intuitively means that the verifier V gains no information about sk when interacting with P. In this paradigm, the verifier can be also honest or dishonest. See [Gol09, chapter 4] for a proper definition of zero-knowledge.
For signature schemes, we often define P and V using a Σ protocol. This is because Σ protocols are easier to define while being honest verifier zero-knowledge interactive protocol, which is sufficient in terms of security to construct using [FS86] existentially unforgeable under chosen message attacks (EU-CCA) signature scheme in the Random Oracle Model (ROM). We define a Σ protocol that is adapted to identification schemes as such.

Definition 3.1.2: Σ-protocols

A Σ **protocol** is an interactive protocol composed of (P,V) that are decomposed in 4 sub-algorithms $\mathsf{Commit}, \mathsf{Challenge}, \mathsf{Response}, \mathsf{Verify}$:



This protocol must ensure

- *Special Soundness.* There exists $\mathcal{E}$ a $\mathsf{PPT}(\lambda)$ algorithm called the *extractor* such that for any pk, if $(\mathsf{pk},\mathsf{com},\mathsf{chal},\mathsf{res})$ and $(\mathsf{pk},\mathsf{com},\mathsf{chal}',\mathsf{res}')$ are two accepting views for V such that $\mathsf{chal} \neq \mathsf{chal}'$, then $\mathcal{E}(\mathsf{pk},\mathsf{com},\mathsf{chal},\mathsf{res},\mathsf{chal}',\mathsf{res}')$ yields sk, a valid secret key.
- *Special Honest-Verifier Zero-Knowledge (HVZK).* There exists $\mathcal{S}$ a $\mathsf{PPT}(\lambda)$ algorithm called the *simulator* such that for any $(\mathsf{sk},\mathsf{pk})$, the transcript $(\mathsf{com},\mathsf{chal},\mathsf{res})$ of the interaction $\mathsf{P}(\mathsf{pp},\mathsf{sk}) \leftrightarrow \mathsf{V}(\mathsf{pp},\mathsf{pk})$ conditioned to chal is computationally indistinguishable from $\mathsf{S}(\mathsf{pp},\mathsf{pk},\mathsf{chal})$.

As previously touched, SQISign and SQISignHD are Σ protocol based identification schemes build upon the Deuring correspondence (hence the acronym SQIS for Short Quaternion Identification Scheme). The main idea behind SQISign and SQISignHD is to prove the knowledge of the endomorphism ring $\mathrm{End}(E_A)$ with E_A a supersingular curve. To do so, the idea is to use the fact that knowing $\mathrm{End}(E_A)$ enables the

prover to find a connecting isogeny between E_A and any other curve E_2, provided that he also knows $\mathrm{End}(E_2)$. The idea is then to let E_2 be chosen as a challenge by the verifier in such a way that the prover can retrieve $\mathrm{End}(E_2)$ and respond the connecting isogeny that can be easily verified. The main difference between SQISign and SQISignHD consist in how this connecting isogeny is computed and represented. The respective architecture of SQISIgn and SQISignHD are given in figure 3.1.

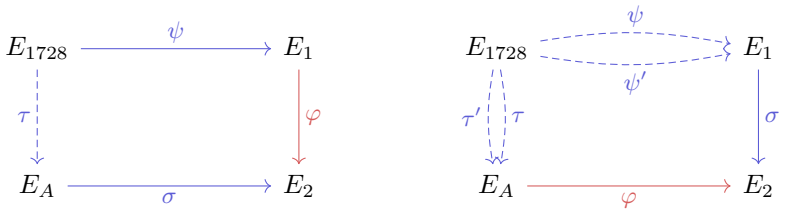


Figure 3.1: Diagrams of SQISign (left) and SQISignHD (right). The prover is in blue and the verifier is in red. Dashed isogenies are secrets.

SQISign: To construct σ the connecting isogeny, SQISign uses a variant of the **KLPT** named the **SigningKLPT** [FKL+20, Algo. 5]. The ideal I_σ is smooth, as its norm is a large power of 2 of size $O(p^3)$. To be efficiently computed, σ is represented as a composition of isogenies with rational kernel generator. Transcribing I_σ to these kernels is done efficiently using **IdealToIsogeny** [FLLW22, Algo. 7] by setting the prime p of SQISign to be such that $2^\ell T | p^2 - 1$ with $T^2 \simeq p^3$ and T smooth. Finding such primes is *difficult* and T often has prime factors in the order of 10^3. Those big factors significantly slow the signing procedure, as several T isogenies have to be computed throughout **IdealToIsogeny**. On the other hand, the verification of SQISign is very efficient, as it essentially consists in computing a sequence of isogenies of degree 2^ℓ from their kernels. SQISign is performed as such.

- KeyGen: Compute $\tau : E_{1728} \to E_A$ together with its corresponding ideal I_τ. E_A is the public key, while τ is the secret key. E_A is the domain of the response isogeny.
- Commit: The prover computes $\psi : E_{1728} \to E_1$ together with its corresponding ideal I_ψ. It gives φ to the verifier.
- Challenge: The verifier then computes a challenge isogeny $\varphi : E_1 \to E_2$ and sends it to the prover. E_2 is the codomain for the answer isogeny.
- Response: Using its knowledge of ψ, the prover uses **KernelToIdeal** to compute I_φ. Then, using the **SigningKLPT** and **IdealToIsogeny**, the prover constructs an isogeny $\sigma : E_2 \to E_1$ different from $\varphi \circ \psi \circ \widehat{\tau}$ and gives σ as a response to the verifier.
- Verify: The verifier then checks that the received isogeny is valid using **KernelToIsogeny**.

SQISignHD: On the other hand, SQISignHD uses the **RandomEquivalentIdeal** to compute σ. The response isogeny is therefore short $\widetilde{O}(\sqrt{p})$ but not smooth. It is then given to the verifier using the HD representation. This shift to HD isogenies considerably speeds up the signature part of SQISignHD but shifts most of the expensive computation to the verification that has to use **EvalKani**. To be efficient, SQISignHD uses "SIDH"-like prime, that are easy to find. SQISignHD is thus performed as such.

- KeyGen: Compute $\tau, \tau' : E_{1728} \to E_A$ together with its corresponding ideal I_τ. E_A is the public key, while τ is the secret key.

- Commit: The prover computes isogenies $\psi, \psi' : E_{1728} \to E_1$ with **DoublePath** together with its ideal I_ψ and shares E_1. This curve is the domain of the response.
- Challenge: The verifier computes a challenge isogeny $\varphi : E_A \to E_2$ and sends it to the prover. E_2 is the codomain for the answer isogeny.
- Response: Using **RandomEquivalentIdeal**, the prover constructs an isogeny $\sigma : E_1 \to E_2$ different from $\varphi \circ \tau \circ \hat{\psi}$, evaluate it using τ', ψ' and **EvalTorsion** and gives this evaluation of σ as a response to the verifier.
- Verify: The verifier then checks that the received isogeny is valid using **EvalKani**.

3.2 New tools

Before jumping into SQIPrime, we detail two new tools that we will use to construct our variant of SQISignHD.

1. The first tool is called **KaniDoublePath**, a variant of **DoublePath** that uses Kani's Lemma to sample supersingular curves at random such that we can compute their endomorphism ring using isogenies of big prime degree. This algorithm is a slight modification of the **RandIsogImages** [NO23, Algorithm 2], as it additionally computes the corresponding ideal of these isogenies.
2. The second is a method to compute, given K a generator of the kernel of an isogeny, the corresponding ideal even when its degree is not smooth. This method is an adaptation of the work of Leroux over verifiable random functions in [Ler23] to use big prime order isogenies as isogeny challenge.

3.2.1 KaniDoublePath

The main idea behind **KaniDoublePath** is likewise to **DoublePath** to construct two isogenies of coprime degree between E_{1728} and another supersingular curve E. The main interest of **KaniDoublePath** lies in the fact that those isogenies are not necessary smooth.
To do so, we first use the **FullRepresentInteger** to find $\gamma \in \mathrm{End}(E_{1728})$ an endomorphism such that $\deg(\gamma) = \ell(N - \ell)$ with ℓ and N coprime such that N is smooth. Using corollary 1.2.8, we can write $\gamma = \rho \circ \tau$ with $\deg \tau = \ell$, $\deg \rho = N - \ell$. Using Kani's Lemma and especially corollary 2.3.2 over the following diagram, we compute the Kani's isogeny F.

$$\begin{array}{ccc} E & \xrightarrow{\hat{\tau}} & E_{1728} \\ {\scriptstyle\rho}\downarrow & \swarrow{\scriptstyle\gamma} & \downarrow{\scriptstyle\hat{\tau}_*\rho} \\ E_{1728} & \xrightarrow[\rho_*\hat{\tau}]{} & E' \end{array}$$

$$F : E_{1728}^2 \to E \times E'$$

$$\ker(F) = \left\{ \left([\ell](P), \gamma(P) \right) \,\middle|\, P \in E_{1728}[N] \right\}$$

We can therefore evaluate both τ and $\hat{\rho}$ at any points of E_{1728}. Additionally to [NO23, Algorithm 2], we also retrieve I_τ and I_ρ the ideal corresponding to τ and ρ using the factorization of γ.

$$I_\tau = \mathcal{O}_{1728}\overline{\gamma} + \mathcal{O}_{1728}\ell \qquad I_\rho = \frac{\overline{I_\tau}\mathcal{O}_{1728}\gamma}{\ell}$$

Algorithm 7 KaniDoublePath

Input: $\mathfrak{O}_{1728}$ an evaluation basis of $\mathrm{End}(E_{1728})$ with $\langle P, Q\rangle$ a basis of $E_{1728}[N]$ and ℓ s.t. $\gcd(\ell, N) = 1$ and $\ell(N-\ell) > p$ with N smooth

Output: $\tau, \hat{\rho} : E_{1728} \to E$ isogenies of respective degree ℓ and $N-\ell$ given as dimension 2 isogenies, together with I_ρ and $I_{\hat{\rho}}$ their ideals.

1: $\gamma \leftarrow$ **FullRepresentInteger**$(\mathfrak{O}_{1728}, \ell(N-\ell))$
2: $\mathsf{B} \leftarrow \{([\ell]P, \gamma(P)), ([\ell]Q, \gamma(Q))\}$
3: $F \leftarrow$ **HDKernelToIsogeny**(B)
4: $I_\tau \leftarrow \mathcal{O}_{1728}\overline{\gamma} + \mathcal{O}_{1728}\ell$
5: $I_{\hat{\rho}} \leftarrow \frac{1}{\ell}\mathcal{O}_{1728}\overline{\gamma}I_\tau$
6: **return** $F, I_\tau, I_{\hat{\rho}}$ ▷ $\tau(P) = F(P, 0)_1$ and $\hat{\rho}(P) = -F(0, P)_1$

When ℓ is prime, contrary to **DoublePath**, we are not considering long paths over *one* supersingular graph $\mathcal{G}_p^\ell$ but are instead considering neighbors of E_{1728} over $\tilde{O}(\sqrt{p})$ distinct supersingular isogeny graphs. We base our analysis on the following assumption.

Assumption 3.2.1:

The following two distributions are computationally indistinguishable.

- E a curve sampled randomly among all supersingular curves.
- E the random neighbour of 1728 in $\mathcal{G}_p^\ell$ with ℓ a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$.

Following lemma 2.2.6, we have that with extremely high probability, any curve E is the neighbor of E_{1728} in some $\mathcal{G}_p^\ell$, with ℓ in the above interval. It is therefore sound to assume that the distribution is close from uniform. It nevertheless would require further studies, as we found non literature on that specific problem. Using this mathematical assumption, we can then justify the output distribution of **KaniDoublePath** as follows.

Corollary 3.2.2:

Under assumption 3.2.1 together with other heuristics and if $N \simeq p$, then the following distributions are computationally indistinguishable.

- E a curve sampled randomly among all supersingular curves.
- E the codomain of τ and $\hat{\rho}$, outputted by **KaniDoublePath**(N, P, Q, ℓ) with ℓ a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$.

Proof of Corollary 3.2.2:
Consider the following distributions:

1. E the codomain of τ and $\hat{\rho}$, outputted by **KaniDoublePath**(N, P, Q, ℓ) with ℓ a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$.
2. E the codomain of τ and $\hat{\rho}$, with $\rho \circ \tau$ an endomorphism of E_{1728} of degree $\ell(N-\ell)$, with ℓ a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$.
3. E the neighbour of 1728 in $\mathcal{G}_p^\ell$ with ℓ a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$.

4. E a curve sampled randomly among all supersingular curves.

We can go from 1. to 2. using lemma 2.2.4 as we have that the **FullRepresentInteger** algorithm outputs endomorphisms that are computationally indistinguishable from a uniform sample among all endomorphism of norm $\ell(N-\ell)$.
To go from 2. to 3., we use the same heuristic argument as [NO23] i.e. that there are at least $N-\ell+1 \simeq p$ isogenies of degree $N-\ell$ over any curve E. It is therefore very likely that we can find an isogeny with domain E and with codomain E_{1728}. If we do the heuristic assumption that isogenies of size $O(p)$ have a codomain close from random, then there is an isogeny of degree $N-\ell$ between E_{1728} and E with probability around $12(N-\ell)/p \simeq 1$.
Going from 3. to 4. is given by assumption 3.2.1.

□ 3.2.2

3.2.2 KernelToIdeal for generic degree isogenies

Going back to the details of **KernelToIdeal**, we see that it makes extensive usage of discrete logarithms over $E[d]$, with d being the degree of the isogeny. To be efficient, this method requires d to be smooth. We therefore need another method for generic degree. The idea proposed by Leroux in [Ler23] is to use the knowledge of the endomorphism ring of E to construct a *special basis* of $E[d]$.

Definition 3.2.3: Special basis

Let E be any supersingular curve. (P, Q, ι, I_P) is **special basis** of $E[d]$, with:

- $P, Q \in E$ such that $\langle P, Q\rangle = E[d]$.
- $\iota \in \mathrm{End}(E)$ such that $\iota(P) = Q$.
- I_P the ideal such that $E[I_P] = \langle P\rangle$.

Given $\mathfrak{O}_E$ an evaluation basis of $\mathrm{End}(E)$, we can construct a special basis using the following algorithm, proposed in [Ler23].

Algorithm 8 FindSpecialBasis

Input: $\mathfrak{O}_E = \left(\{b_i\}_{i=1}^4, \delta\right)$ an evaluation basis of E with d an integer
Output: (P, Q, ι, I_P) a special basis of $E[d]$.

1: Sample $R \in_\$ E[d]$
2: Sample $\alpha \in_\$ \mathcal{O}_E$ such that $\gcd\left(n(\alpha), d^2\right) = d$
3: **if** $\delta^{-1}(\alpha)(R) = 0$ **do** try with new R.
4: $P \leftarrow \delta^{-1}(\alpha)(R)$
5: $I_P \leftarrow \mathcal{O}_E\overline{\alpha} + \mathcal{O}_E d$
6: Sample $\iota \in_\$ \mathcal{O}_E$ such that $\gcd(n(\iota), d) = 1$
7: **if** $e_d(P, \delta^{-1}(\iota)(P)) = 1$ **do** sample new ι. ▷ Ensures they are not colinear
8: **return** $P, \delta^{-1}(\iota)(P), \delta^{-1}(\iota), I_P$

Using special basis, we can compute ideals from a kernel generator $K \in E[d]$ using the following lemma.

Lemma 3.2.4:

Let (P, Q, ι, I_P) be a special basis of $E[d]$ and let $K = [a]P + [b]Q$ be a point in $E[d]$. Then $\phi_K : E \to E/_{\langle K \rangle}$ has for representing ideal

$$I_K = [a + b\delta(\iota)]_* I_P$$

Proof of Lemma 3.2.4:

$$\begin{aligned}\langle K \rangle &= \langle [a]P + [b]Q \rangle \\ &= \langle [a]P + [b]\iota(P) \rangle \\ &= [a + b\iota]\langle P \rangle\end{aligned}$$

$$\text{i.e. } \phi_K = [a + b\delta(\iota)]_* \phi_P$$

using the Deuring correspondence, we get the desired result.

□ 3.2.4

We can therefore compute ideals of any degree but the method that we presented here requires knowing $\mathfrak{O}_E$. We now present a modification of lemma 3.2.4 that just requires knowing $\mathfrak{O}_{1728}$ and $\phi : E_{1728} \to E$ an isogeny of degree coprime to d.

Corollary 3.2.5:

Let (P, Q, ι, I_P) be a special basis of $E_{1728}[d]$ and let $\phi : E_{1728} \to E$ be an isogeny with corresponding ideal I_ϕ such that d and $\deg(\phi)$ are coprime. Let $S, T \in E$ be the respective images of P and Q by ϕ and let $K = [a]S + [b]T$ be a point in $E[d]$. Then,

$$I_K = \left[(a + b\delta(\iota))I_\phi\right]_* I_P$$

Proof of Corollary 3.2.5:
Similarly to lemma 3.2.4, we have that

$$\begin{aligned}\langle K \rangle &= [q]\langle K \rangle \\ &= \phi\hat{\phi}\langle [a]S + [b]T \rangle \\ &= \phi\langle [a]\hat{\phi}(S) + [b]\hat{\phi}(T) \rangle \\ &= \phi\langle [ad]P + [bd]Q \rangle \\ &= \phi\langle [a]P + [b]Q \rangle \\ &= \phi\langle [a]P + [b]\iota(P) \rangle \\ &= \phi \circ [a + b\iota]\langle P \rangle\end{aligned}$$

$$\text{i.e. } \phi_K = [\phi \circ (a + b\iota)]_* \phi_P \text{ and thus } I_K = [(a + b\delta(\iota))I_\phi]_* I_P$$

□ 3.2.5

Note that [Ler23] propose to use ϕ to directly compute a special basis over E. Indeed, if (P, Q, ι, I_P) is a special basis over $E_{1728}[d]$, then $\big(\phi(P), [\deg(\phi)]\phi(Q), \theta, [I_\phi(a + b\delta(\theta))]_* I_P\big)$ is a special basis of $E[d]$ with $\theta = \phi \circ \iota \circ \widehat{\phi}$. The main interest of corollary 3.2.5 comes from the fact that it only uses endomorphism over E_{1728} and not over E. This therefore lightens the computational cost, and is more suited to our usage in SQIPrime.

3.3 Construction

Now that we are familiar with the architecture behind SQISign and SQISignHD and have introduced and explained the new tools that it utilize, we can construct SQIPrime. As previously stated in the introduction, SQIPrime further-expend the use of the Kani's Lemma to both KeyGen and Commit. More precisely, SQIPrime is based on the architecture in Figure 3.2 and is performed as such

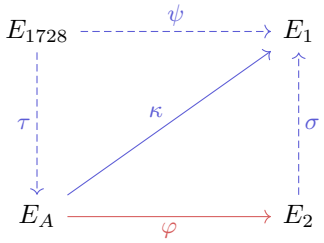


Figure 3.2: Diagram of SQIPrime, prover in blue and verifier in red. Dashed isogenies are secrets

- KeyGen: Compute $\tau : E_{1728} \to E_A$ together with its corresponding ideal I_τ using **KaniDoublePath**. Additionally, compute a matrix $\mathbf{M}$ and use it to mask the image through τ of a special basis of degree qN, with $q \simeq 2^\lambda$ prime. E_A and the masked basis is the public key, while τ and the matrix $\mathbf{M}$ is the secret key.
- Commit: The prover computes an isogeny $\psi : E_{1728} \to E_1$ with **KaniDoublePath** together with its ideal I_ψ and shares E_1. This curve is the domain of the response.
- Challenge: The verifier computes a challenge point $C_1 \in E_A[q]$ and send it to the prover.
- Response: Using the special basis over E_{1728} and its knowledge of I_τ, the prover retrieves I_φ, with $\ker(\varphi) = \langle C_1 \rangle$. It then computes $\sigma : E_2 \to E_1$ different from $\psi \circ \hat{\tau} \circ \hat{\varphi}$ using **RandomEquivalentIdeal** and construct $\kappa = \sigma \circ \varphi$, evaluate it using **EvalTorsion** and send this evaluation of κ as a response to the verifier.
- Verify: The verifier receives κ and checks using **EvalKani** that it is valid by seeing if $\kappa(C_1) = 0$.

The public parameters of SQIPrime are defined as such.

Algorithm 9 SQIPrime.Setup

Input: 1^λ
Output: $\mathsf{pp} = \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$

1: Take p a prime of the form $p = 2^{2\lambda} f - 1$ such that $p - 1 = 2Nq$ with $q \simeq 2^\lambda$ prime and N coprime to q.
2: $P_0, Q_0 \leftarrow$ **CanonicalTorsionBasis**$(E_{1728}, 2^{2\lambda})$
3: $(P, Q, \iota, I_P) \leftarrow$ **FindSpecialBasis**$(\mathfrak{O}_{1728}, qN)$
4: Compute $I_{[N]P} = I_P + \mathcal{O}_{1728} q$
5: $\beta \leftarrow \lceil \log_2(p)/2 + \log_2(q) + \log_2 \log_2(p) \rceil$
6: $\mathsf{pp} \leftarrow \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$
7: **return** pp

3.3.1 Key generation & commitment

Both key generation and commitment consist essentially in using **KaniDoublePath**. We take a random prime $\ell \in [\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ and use the **KaniDoublePath** with an endomorphism of norm $\ell(2^{2\lambda} - \ell)$ to retrieve τ in the case of **SQIPrime.KeyGen** and ψ in **SQIPrime.Commit**. The only significant differences between the secret key and the challenge generation is that during the key generation, we additionally compute a masked basis of $E_A[Nq]$. To do so, we compute the image of (P, Q) through the isogeny τ and use a random matrix $\mathbf{M} \in \mathrm{GL}_2(Nq)$ to mask the torsion points. Note that this masking is necessary as N could be smooth, in which case, we could retrieve τ_A using **EvalKani**.

Algorithm 10 SQIPrime.KeyGen

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$
Output: $\mathsf{sk} = \big(E_A, \mathsf{F_A}, I_\tau, \mathbf{M}\big)$, $\mathsf{pk} = \big(E_A, (R, S)\big)$ with $\mathsf{F_A}$ a HD-isogeny representing $\tau : E_{1728} \to E_A$ with corresponding ideal I_τ. $\mathbf{M} \in \mathrm{GL}_2(Nq)$ and R, S a basis of $E_A[Nq]$.

1: Sample ℓ_A a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ such that $\ell_A \neq q$.
2: $\mathsf{F_A}, I_\tau, * \leftarrow$ **KaniDoublePath**$(2^{2\lambda}, P_0, Q_0, \ell_A)$
3: Compute E_A.
4: Sample $\mathbf{M} \in_\$ \mathrm{GL}_2(Nq)$
5: $\binom{R}{S} \leftarrow M\binom{F(P,0)_1}{F(Q,0)_1}$
6: **return** $(\mathsf{F_A}, I_\tau, \mathbf{M})$, $\big(E_A, (R, S)\big)$

Algorithm 11 SQIPrime.Commit

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$
Output: $\mathsf{sec} = \big(E_1, \mathsf{F_1}, I_\psi\big)$, $\mathsf{pub} = (E_1)$ with $\mathsf{F_1}$ a HD-isogeny representing $\psi : E_{1728} \to E_1$ with corresponding ideal I_ψ.

1: Take ℓ_1 a random prime in $[\sqrt{p}\log(p)^{-1}, \sqrt{p}\log(p)]$ such that $\ell_1 \neq q$
2: $\mathsf{F_1}, I_\psi, * \leftarrow$ **KaniDoublePath**$(2^{2\lambda}, P_0, Q_0, \ell_1)$
3: Compute E_1
4: **return** $\big(\mathsf{F_1}, I_\psi\big)$, $\big(E_1\big)$

3.3.2 Challenge & response

Challenge

As touched earlier, our challenge is significantly different from the challenge of SQISign and SQISignHD, as the evaluation of the challenge isogeny has been moved from the verifier to the prover. This adjustment is necessary since the verifier lacks an efficient means to evaluate this isogeny, as it only has access to kernel representation of φ, whose degree is not smooth. In idea, the prover uses ideal representation to construct an HD representation of φ that is then sent to the verifier together with the HD representation of the answer isogeny σ. Thus, instead of providing an isogeny of smooth degree, the challenger simply sends a challenge point $C_1 \in E_A[q]$. This point is given as $a \in \mathbb{Z}_q$ such that $C_1 = [N](R + [a]S)$. with R, S given during **SQIPrime.KeyGen**. This point is the generator of the kernel of $\varphi : E_A \to E/\langle C_1 \rangle = E_2$. We have $q \simeq 2^\lambda$ possible challenge isogenies.

Response

In line with SQISignHD, our objective is to compute an isogeny $\sigma : E_2 \to E_1$ but the verifier lacks knowledge of E_2. An idea would be to provide him with an HD representation of φ as the verifier could check that the kernels match but the problem lies in the need for knowledge of a third map between E_{1728} and E_2, which is something that would be complex to construct.[1]
So instead of sending σ and φ separately, we send $\kappa = \sigma \circ \varphi$ and use the Kani's Lemma over κ to prove that φ factors through κ using the fact that $\ker(\kappa) \cap E_A[q] = \ker(\varphi)$. But first, we have to adapt corollary 3.2.5 to compute $I_{C_1} = I_\varphi$. Having received the challenge $\mathsf{Chal} = a$, the prover has to find $b, c \in \mathbb{Z}_q$ such that $C_1 = [N]\Big([c]\tau(P) + [d]\tau(Q)\Big)$. Those scalars are computable as $\binom{b}{c} = \mathbf{M}^{-1}\binom{1}{a}$. Having computed b and c, we can then compute I_{C_1} as

$$I_{C_1} = \big[\big(b + c\delta(\iota)\big)I_\tau\big]_* I_{[N]P}$$

We then compute the $(\mathcal{O}_2, \mathcal{O}_1)$-ideal $\overline{I_{C_1} I_\tau} I_\varphi$ and find another small $(\mathcal{O}_2, \mathcal{O}_1)$-ideal J using **RandomEquivalentIdeal**. Following lemma 2.2.4, we expect $n(J)$ to be smaller than $\sqrt{p}\log(p)$. J correspond to the isogeny $\sigma : E_2 \to E_1$ of prime degree d that closes our diagram in figure 3.2. Additionally, we want, to ensure the efficiency of the verification, that $2^\beta - qd = 1 \mod 4$ and is prime and thus use the Kani's Lemma in dimension 4. If d does not have this property, then we simply sample a new J using **RandomEquivalentIdeal**. Heuristically, this event should occur with probability $\mathcal{O}(1/\lambda)$.[2]
The response to our challenge is to give the isogeny $\kappa = \sigma \circ \varphi$ to the verifier as an HD representation together with d the degree of σ. We first call **CanonicalTorsionBasis** over E_A to find a basis[3] of $E_A[2^{2\lambda}]$ and then simply use **EvalTorsion**. Note that the probability that $d = \ell_1$ or that $d = \ell_A$ is negligible. Additionally, we also use **EvalTorsion** to compute and send the image of a third point $C_2 = [a]R - S$. This additional point is used to ensure the soundness of our verification. It is important to note that C_2 is such that $\langle C_1, [N]C_2 \rangle = E_A[q]$.

Algorithm 12 SQIPrime.Response

Input: $\mathsf{pp}, \mathsf{sk}, \mathsf{sec}, \mathsf{chal} = \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big), (\mathsf{F_A}, I_\tau, \mathbf{M}), (\mathsf{F_1}, I_\varphi), a$ with $a \in \mathbb{Z}_q$.
Output: $\mathsf{res} = (S, T, U, d)$ with $S, T \in E_1[2^{2\lambda}]$, $U \in E_1[Nq]$ and d the degree of φ.

1: $\binom{b}{c} \leftarrow \mathbf{M}^{-1}\binom{1}{a}$
2: $I_{C_1} \leftarrow [(b + c\iota)I_\tau]_* I_{[N]P}$
3: $J \leftarrow$ **RandomEquivalentIdeal**$(\overline{I_{C_1} I_\tau} I_\psi)$ $\quad$ $d \leftarrow n(J)$
4: **check if** $2^\beta - dq = 1 \mod 4$ and is prime. If not, go back to line 3.
5: $X, Y \leftarrow$ **CanonicalTorsionBasis**$(E_A, 2^{2\lambda})$
6: $C_2 \leftarrow [a]R - S$
7: Define $\tau = (F_A(-, 0))_1$ and $\psi = (F_1(-, 0))_1$.
8: $S, T, U \leftarrow$ **EvalTorsion**$\Big(\mathfrak{O}_{1728}, \tau, I_\tau, \psi, I_\psi, I_{C_1} J, qd, \{X, Y, C_2\}\Big)$
9: **return** $\mathsf{res} = (S, T, U, d)$

3.3.3 Verification

Upon receiving S, T, U, d we want to verify that the following statement hold:

[1] We could use the **KLPT** and then use the **IdealToKernel** algorithm but avoiding this algorithm was a reason behind the development of SQISignHD.
[2] This is the method used in [DLRW23]. They furthermore constructed in [DLRW23, E.2] a method to efficiently perform this random sampling.
[3] As they are given by **CanonicalTorsionBasis**, they are not sent to the verifier. We could also take any basis of $E_A[2^{2\lambda}]$ but we would need to specify in res.

- The torsion points we received define an HD representation of an isogeny $\kappa : E_A \to E_1$ of degree dq such that the isogeny φ factors through κ, meaning that $\ker(\kappa)[q] = \langle C_1 \rangle$.

To do this verification efficiently, we modify **EvalKani** so that we never have to compute the full isogeny F. We have the following diagram:

$$\begin{array}{ccc} E_A^2 & \xrightarrow{\Sigma} & E_1^2 \\ {\scriptstyle\gamma}\downarrow & & \downarrow{\scriptstyle\gamma} \\ E_A^2 & \xrightarrow{\Sigma} & E_1^2 \end{array}$$

$$\gamma := \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \text{ with } \deg(\gamma) = \sum_{i=1}^{2} a_i^2;\ \Sigma := \operatorname{diag}(\kappa, \kappa)$$

Using Kani's Lemma and corollary 2.3.2, we split the isogeny

$$F = \begin{pmatrix} \tilde{\Sigma} & -\tilde{\gamma} \\ \gamma & \Sigma \end{pmatrix}$$

in two isogenies F_1 and F_2 with $F = F_2 \circ F_1$ and $\deg F_i = d_i$ with

$$\ker F_1 = \left\{ \big(\Sigma(P), -\gamma(P)\big) \,\middle|\, P \in E_A^2[d_1] \right\} \qquad \ker \widetilde{F_2} = \left\{ \big(-\tilde{\gamma}(P), \Sigma(P)\big) \,\middle|\, P \in E_A^2[d_2] \right\}$$

We then use the following property:
Let $X \in E_A$ be a point of order coprime to $d_1 d_2$, Then we have the following equivalence.

$$F \begin{pmatrix} 0 \\ 0 \\ X \\ 0 \end{pmatrix} = \begin{pmatrix} [a_1]X \\ [-a_2]X \\ Y \\ 0 \end{pmatrix} \iff [d_2]F_1 \begin{pmatrix} 0 \\ 0 \\ X \\ 0 \end{pmatrix} = \widetilde{F_2} \begin{pmatrix} [a_1]X \\ [-a_2]X \\ Y \\ 0 \end{pmatrix}$$

We use this equivalence on the two points C_1 and C_2 of respective order q and Nq.

Algorithm 13 SQIPrime.Verify

Input: $\mathsf{pp}, \mathsf{pk}, \mathsf{com}, \mathsf{chal}, \mathsf{res} = \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big), (E_A, R, S), E_1, a, (S, T, U, d)$
Output: $b \in \{0, 1\}$

1: **if** one of the points S, T, U is not in E_1 **do return** 0
2: $d_1 \leftarrow 2^{\lfloor \frac{\beta}{2} \rfloor}, d_2 \leftarrow 2^{\lceil \frac{\beta}{2} \rceil}$, $k_1 \leftarrow 2^{2\lambda}/d_1$, $k_2 \leftarrow 2^{2\lambda}/d_2$
3: $(a_1, a_2) \leftarrow$ **Cornacchia**$(2^\beta - qd)$
4: Compute γ and $\tilde{\gamma}$
5: Compute $\{P_{i,j}\}_{0 \leqslant i,j \leqslant 2,2}$ a basis of $E_A^2[2^{2\lambda}]$ ▷ Using **CanonicalTorsionBasis**
6: $\mathsf{B}_1 \leftarrow \big\{\big([k_1]\Sigma(P_{i,j}), [-k_1]\gamma(P_{i,j})\big)\big\}_{0 \leqslant i,j \leqslant 2,2}$ ▷ $\Sigma(P_{i,j})$ computed using S, T
7: $\mathsf{B}_2 \leftarrow \big\{\big([-k_2]\tilde{\gamma}(P_{i,j}), [k_2]\Sigma(P_{i,j})\big)\big\}_{0 \leqslant i,j \leqslant 2,2}$
8: $F_1 \leftarrow$ **HDKernelToIsogeny**(B_1)
9: $\tilde{F}_2 \leftarrow$ **HDKernelToIsogeny**(B_2)
10: **if** codomain$(F_1) \neq$ codomain$(\tilde{F}_2)$ **do return** 0 ▷ Do like [DLRW23, section F.3]
11: $C_1 \leftarrow [N](R + [a]S)$, $C_2 \leftarrow ([a]R - S)$
12: $b_1 \leftarrow [d_2]F_1(0, 0, C_1, 0) \stackrel{?}{=} \tilde{F}_2([a_1]C_1, [-a_2]C_1, 0, 0)$
13: $b_2 \leftarrow [d_2]F_1(0, 0, C_2, 0) \stackrel{?}{=} \tilde{F}_2([a_1]C_2, [-a_2]C_2, U, 0)$ and $[N]U \neq 0$.
14: **return** $b_1 \wedge b_2$

Proposition 3.3.1: Correctness of SQIPrime

Let $\mathsf{pp}, \mathsf{pk}, \mathsf{chal}$ be a valid public key, commitment, and challenge of SQIPrime and let P, Q be the canonical basis of $E_A[2^{2\lambda}]$. Let $\overline{\mathsf{Res}}$ be any possible response. Then

$$\textbf{SQIPrime.Verify}(\mathsf{pp}, \mathsf{pk}, \mathsf{pub}, \mathsf{chal}, \overline{\mathsf{Res}}) = 1 \iff \overline{\mathsf{Res}} = (\overline{S}, \overline{T}, \overline{U}, \overline{d}) \text{ is such that:}$$

- $(P, Q, \overline{S}, \overline{T})$ is an HD representation of an isogeny $\kappa : E_A \to E_1$ of degree $q\overline{d}$.
- $\ker(\kappa) \cap E[q] = \langle C_1 \rangle$.

Proof of Proposition 3.3.1:
Our proof takes inspiration from [DLRW23, section E.5]. Assume that **SQIPrime.Verify**$(\mathsf{pp}, \mathsf{pk}, \mathsf{pub}, \mathsf{chal}, \overline{\mathsf{Res}}) = 1$. Then, this means that $\overline{S}, \overline{T}, \overline{U}$ are in E_1, that $[N]\overline{U} \neq 0$, that $\overline{F_1}$ and $\overline{F_2}$ are well-defined, have the same codomain and that the following equalities hold.

$$[d_2]\overline{F_1}(0, 0, C_1, 0) = \widetilde{\overline{F_2}}(a_1 C_1, -a_2 C_2, 0, 0) \text{ and } [d_2]\overline{F_1}(0, 0, C_2, 0) = \widetilde{\overline{F_2}}(a_1 C_1, -a_2 C_2, \overline{U}, 0)$$

$$\text{Therefore } \overline{F}(0, 0, C_1, 0) = (a_1 C_1, -a_2 C_2, 0, 0) \text{ and } \overline{F}(0, 0, C_2, 0) = (a_1 C_1, -a_2 C_2, \overline{U}, 0)$$

From isogeny $\overline{F}$,using ι_i and ρ_j the standard injections/restrictions of product spaces, we can construct 16 elliptic curve isogenies $\overline{F}_{i,j} = \rho_i \circ \overline{F} \circ \iota_j$ with $1 \leqslant i, j \leqslant 4$ such that for all $j = 1, \cdots, 4$:

$$\sum_{i=1}^{4} \deg(\overline{F}_{i,j}) = \deg(\overline{F}) = 2^\beta$$

We interest ourselves at $j = 3$. We want to show that for $i = 1, 2$ and 4, $F_{i,3} = [b_i]$ with $b_i = a_1, -a_2$ and 0. To do so, we use corollary 1.3.9. Indeed, using the triangular inequality.

$$\text{for } i = 1, 2, 4; \ \deg(\overline{F}_{i,3} - [b_i]) \leqslant 4 \cdot 2^\beta \simeq 2^{2\lambda + 2\log(\lambda) + 2}$$

But we know that $\overline{F}_{i,3} = [b_i]$ for all points generated by $\langle C_1, C_2 \rangle$, i.e, for $Nq^2 \simeq 2^{3\lambda}$ points. Thus, $\overline{F}_{1,3} = [a_1]$, $\overline{F}_{2,3} = [-a_2]$ and $\overline{F}_{4,3} = 0$, meaning using the previous equality that $\overline{F}_{3,3}$ is an isogeny of degree qd between E_A and E_1. Furthermore, we have that $\overline{F}_{3,3}(C_1) = 0$ and that $\overline{F}_{3,3}([N]C_2) \neq 0$, meaning that $\ker \overline{F}_{3,3} \cap E_A[q] = \langle C_1 \rangle$, proving our point.

□ 3.3.1

Table 3.1 gives us a comparative between SQISign, SQISignHD and SQIPrime.

	SQISign	**SQISignHD**	**SQIPrime**
prime	$2^f T \mid (p^2-1)$ and $T = DT'$	$p = 2^\lambda 3^{\lambda'} f - 1$	$p = 2^{2\lambda} f - 1$ and $p - 1 = 2Nq$
Key gen	$l^\bullet$ isogenies	2^λ isogenies	$(2,2)$-isogenies
Commitment	T' isogenies	2^λ isogenies	$(2,2)$-isogenies
Challenge	D isogenies	$3^{\lambda'}$ isogenies	$C \in E_A[q]$
Response	kernel representation	HD representation	HD representation
Verification	$l^\bullet$ isogenies	$(2,2)$-isogenies	$(2,2)$-isogenies

Table 3.1: Comparative of the SQISign Family

3.4 Security analysis

To construct the SQIPrime digital signature scheme by applying the Fiat-Shamir transform [FS86], we still have to prove that SQIPrime is an identification scheme, i.e. that SQIPrime is a Σ protocol. We also discuss how to find good prime numbers for SQIPrime.

3.4.1 SQIPrime is a Σ protocol

To prove that SQIPrime is a Σ protocol, we have to prove special soundness and HVZK. The extractor is constructed as follows.

Proposition 3.4.1: SQIPrime Extractor $\mathcal{E}$

Let $(E_1, a_1, S_1, T_1, U_1, d_1)$ and $(E_1, a_2, S_2, T_2, U_2, d_2)$ be 2 transcripts with identical commitment E_1 and different challenges points a_1 and a_2. There exists an extractor $\mathcal{E}$ that, given both transcript, can efficiently solve the endomorphism problem over E_A.

Proof of Proposition 3.4.1:
Our proof is very similar to [DLRW23, section 5.1]. The only meaningful difference comes from the fact that the probability that any two of q, d_1 and d_2 are not coprime is negligible.
We can use S_1, T_1 to compute an HD representation of $\kappa_1 = \sigma_1 \circ \varphi_1$ and S_2, T_2 to compute an HD representation of $\widehat{\kappa_2} = \widehat{\sigma_2 \circ \varphi_2}$. Then, $\alpha = \widehat{\kappa_2} \circ \kappa_1 \in \mathrm{End}(E_A)$ is non-scalar, as otherwise, we have that $\alpha = [\chi]$ such that $\chi^2 = q^2 d_1 d_2$ and thus $\chi = q\chi'$. Therefore $[d_2]\kappa_1 = [\chi']\kappa_2$ and as d_2, d_1 and χ' are coprime, this induces that $\varphi_1 = \varphi_2$ i.e., that $a_1 = a_2$, which is a contradiction.

□ 3.4.1

The extractor ensures us that SQIPrime has special soundness. Similarly to [DLRW23, section 5.2], we construct the simulator under the assumption that we have access to the following oracle.

Definition 3.4.2: RUCGIO

The **Random Uniformly Constrained Good Isogeny Oracle** (RUCGIO) is an oracle that take as input E any supersingular curve together with $P \in E[q]$ and that return an efficient representation of $\kappa : E \to E'$ of degree $q\ell$ with ℓ prime such that:

- E' is uniformly distributed over all supersingular curves.
- κ is uniformly distributed among all isogenies between E and E' such that $P \in \ker(\kappa)$ and such that $2^\beta - q\ell$ is a prime equal to $1 \mod 4$.

Proposition 3.4.3: SQIPrime simulator $\mathcal{S}$

Given pp, pk and chal, there exists a $\mathsf{PPT}(\lambda)$ simulator $\mathcal{S}$ with access to a RUCGIO that simulates transcripts with a distribution that is computationally indistinguishable from the distribution of transcripts of SQIPrime, conditioned to chal.

Proof of Proposition 3.4.3:
Given $a \in \mathbb{Z}_q$, we compute $C_1 = [N](R + [a]S)$. Calling RUCGIO over E_A and C_1, we retrieve an efficient representation of $\kappa : E_A \to E_1$ and use this representation to compute the points $X = \kappa(A), Y = \kappa(B)$, and $Z = \kappa([b]R - [a]S)$ with A, B the canonical basis over $E_A[2^{2\lambda}]$.
We then simply return the following transcript

$$(E_1, a, X, Y, Z, \deg(\kappa)/q)$$

This transcript is computationally indistinguishable from a genuine transcript, as:

- Following corollary 3.2.2, we have that a genuine E_1 or one given by RUDGIO are computationally indistinguishable.
- Following lemma 2.2.6, a genuine κ or one given by RUDGIO are computationally indistinguishable, and so does $X, Y, Z, \frac{\deg(\kappa)}{q}$.

□ 3.4.3

We now make the following assumption.

Assumption 3.4.4:

The endomorphism problem remains hard even when given access to RUCGIO.

By definition, RUCGIO, when given an input C, generates a random isogeny that factors ϕ_C and that are of good degree. If C is of smooth order, then RUCGIO is in fact equivalent to the RUGDIO oracle [DLRW23, Definition 5.2.1]. Thus, the arguments of [DLRW23, section 5.3] also applies to RUCGIO. it is therefore reasonable to assume that RUCGIO does not help to break the endomorphism problem.
Thus, using the Fiat-Shamir transform [FS86], we construct a digital signature scheme that is EU-CCA in the ROM.

3.4.2 Finding "SQIPrime-friendly" primes

As touched on in **SQIPrime.Setup**, public parameters and especially the base prime numbers p are different from primes in [FKL+20] and [DLRW23]. They can in fact be seen as a mix between the 2, as

they are similar to the "SIDH" primes of SQISignHD but also require conditions on both $p+1$ and $p-1$ like SQISign. Nevertheless, in SQIPrime, the condition is just that $p-1$ has a factor of size $O(\sqrt{p})$. It is thus easier to find "SQIPrime-friendly" primes than to find 'SQISign-friendly" primes. They can in fact be found by brute-force over f. Indeed, if we take $p = 2^{2\lambda}f - 1$ prime, then the probability that a random prime q divides $(p-1)$ is $1/q$. As there are about $2^\lambda(2^t-1)/\lambda$ distinct primes in $[2^\lambda, 2^{\lambda+t}]$, we have that the probability that there is a prime q in $[2^\lambda, 2^{\lambda+t}]$ that divides p is heuristically

$$\begin{aligned}\mathbb{P}\left[\exists q \in \left[2^\lambda, 2^{\lambda+t}\right] \text{ such that } q\middle|(p-1)\right] &\simeq \sum_{q\geqslant 2^\lambda}^{2^{\lambda+t}} \mathbb{P}\big[q|(p-1)\big] \simeq \sum_{q\geqslant 2^\lambda}^{2^{\lambda+t}} \frac{1}{q} \geqslant \sum_{i=1}^{t} \sum_{q\geqslant 2^{\lambda+i-1}}^{2^{\lambda+i}} \frac{1}{2^{\lambda+i}} \\ &\simeq \sum_{i=1}^{t} \frac{2^{\lambda+i}}{(\lambda+i)} \frac{1}{2^{\lambda+i}} \simeq \sum_{i=1}^{t} \frac{1}{\lambda+i} \geqslant \frac{t}{\lambda+t}\end{aligned}$$

We give here a few examples of good candidates we found using this brute force method for $\lambda = 128$:

$$\begin{aligned}p+1 &= 2^{2\cdot 128} \cdot 11 \cdot 13 \simeq 2^{263.15} \\ p-1 &= 2 \cdot 3^2 \cdot 127 \cdot 2797 \cdot 112170853 \cdot 772493863 \cdot 2770313983597 \cdot q \\ q &= 147639672482289482282790769905784189787 3 \simeq 2^{130.11}\end{aligned}$$

$$\begin{aligned}p+1 &= 2^{2\cdot 120} \cdot 39405 \simeq 2^{255.93} \\ p-1 &= 2 \cdot 3 \cdot 149 \cdot 2745335386200139 \cdot 122125102148171050639 \cdot q \\ q &= 369237590624773543866334185733060208813 \simeq 2^{128.11}\end{aligned}$$

$$\begin{aligned}p+1 &= 2^{2\cdot 120} \cdot 167 \cdot 397 \simeq 2^{256.01} \\ p-1 &= 2 \cdot 3 \cdot 7 \cdot 11 \cdot 41 \cdot 5683514583831199 \cdot 500402127095125861 \cdot q \\ q &= 217442272953827514442892286 3792468335219 \simeq 2^{130.67}\end{aligned}$$

The first prime is in line with our definition, while the latter two are constructed to be very close to 2^{256}. In each case, we have enough 2 torsion points to compute all HD isogenies.

Efficiency of SQIPrime

The next step with SQIPrime is to write an efficient implementation. This is something that we were unable to do in the scope of this thesis, as that would have required a proper implementation of HD isogenies. Nevertheless, we can make a reasonable assumption on the efficiency of SQIPrime based on the implementations in SAGE of [DLRW23, section 6.2] and [NO23, section 5.3]. We expect to have similar speed for **SQIPrime.KeyGen** and **SQIPrime.Commit** as QFESTA's KeyGen. We also expect **SQIPrime.Verify** to be two times slower than its counterpart in SQISignHD due to the fact that we will use Kani's Lemma over isogenies that are two time longer. This is therefore encouraging, especially with the later speed-up in computing $(2,2)$ isogenies in [DMPR23].

Chapter 4

SILBE: an UPKE on lollipop attacks

In this chapter, we present a new Public Key Encryption scheme (PKE) named Supersingular Isogeny Lollipop Based Encryption or SILBE ("syllable" in german). As its name entails, SILBE's inner workings are rooted in lollipop attacks, particularly leveraging the generalized lollipop attack [CV23] on M-SIDH [FMP23]. Thanks to its architecture, we can easily make of SILBE an Updatable Public Key Encryption scheme (UPKE). This makes of SILBE the first[1] isogeny-based UPKE not based on group actions as are [LR22] and [EJKM20, section 6].
This chapter will be structured as such: Section 4.1 explains the definitions of UPKE, M-SIDH and of lollipop attacks. Section 4.2 details how we construct the PKE SILBE. Finally, section 4.3 explains how we can make of SILBE an UPKE together with a discussion on parameter selection.

4.1 Generalities

4.1.1 UPKE

First and foremost, we need to properly define the notion of Updatable Public Key Encryption (UPKE). This notion was initially introduced in [BLMR15] as a relaxation of Forward Secure Public Key Encryption (FSPKE), given the inherent complexity of constructing FSPKE systems and the shared advantageous properties between the two. In addition to functioning as a PKE, UPKE allows for secure asynchronous key updates. several UPKE schemes have been proposed based on discrete logarithm, LWE or DCR. [DKW22, AHLP22]. The definition of UPKE provided below is taken from [EJKM20].

Definition 4.1.1: Updatable Public Key Encryption

Given λ a security parameter, an **UPKE** scheme is given by a set of 6 $\mathsf{PPT}(\lambda)$ together with a setup algorithm $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ the public parameters.

- $\mathsf{KG}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{sk}, \mathsf{pk})$
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}) \xrightarrow{\$} \mathsf{ct}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \longrightarrow \mathsf{m}$
- $\mathsf{UG}(\mathsf{pp}) \xrightarrow{\$} \mu$
- $\mathsf{Upk}(\mathsf{pk}, \mu) \longrightarrow \mathsf{pk}'$
- $\mathsf{Usk}(\mathsf{sk}, \mu) \longrightarrow \mathsf{sk}'$

Likewise to PKE, they also must ensure *correctness*

[1] To our knowledge, the only other proposed architecture was [EJKM20, section 5], that was based on the extended-SIDH and was named by its authors an "online UPKE" as it was not fully asynchronous.

$$\mathbb{P}\left[\mathsf{Dec}\big(\mathsf{sk_i}, \mathsf{Enc}(\mathsf{pk_i}, \mathsf{m})\big) = \mathsf{m} \;\middle|\; \begin{array}{c} (\mathsf{sk_0}, \mathsf{pk_0}) \xleftarrow{\$} \mathsf{KG}(1^\lambda), \\ \mu_\mathsf{i} \xleftarrow{\$} \mathsf{UG}(1^\lambda), \\ \big(\mathsf{sk_i}, \mathsf{pk_i}\big) \xleftarrow{\$} \Big(\mathsf{Usk}(\mathsf{sk_{i-1}}, \mu_\mathsf{i}), \mathsf{Upk}(\mathsf{pk_{i-1}}, \mu_\mathsf{i})\Big) \end{array}\right] = 1$$

We make a slight abuse of notation, as all algorithms know of pp, but this choice is made to clarify already heavy notations. The idea behind the security of an UPKE is to be a secure PKE with a key update mechanism that ensures both *Forward Security* and *Post-Compromise Security*. The first notion means that if the adversary learns about sk_i, then it can not use this information to retrieve sk_j for $j < i$ without knowing the update values μ_j. Similarly, the second notion induces that the adversary is not able to retrieve sk_j for $j > i$ without knowing the update values μ_j.
To ensure that those security notions are respected and to enable the adversary to adaptively choose updates, we use the following oracles and lists.

- Upd_list and Cor_list are two lists that respectively store the updates made by the adversary and what keys are corrupted.
- Fresh_Upd: The *Fresh-Update oracle* samples a random update μ_i, computes the updated keys $(\mathsf{sk_{i+1}}, \mathsf{pk_{i+1}})$ and return $\mathsf{pk_{i+1}}$.
- Given_Upd: The *Given-Update oracle* computes the keys $(\mathsf{sk_{i+1}}, \mathsf{pk_{i+1}})$ corresponding to a given update μ_i and return $\mathsf{pk_{i+1}}$. The update $(i, i+1)$ is added to Upd_list.
- Corrupt: The *Corruption oracle* that receive an index j and return sk_j. It marks j as corrupted together with all others keys of index i such that there is no fresh update in-between.

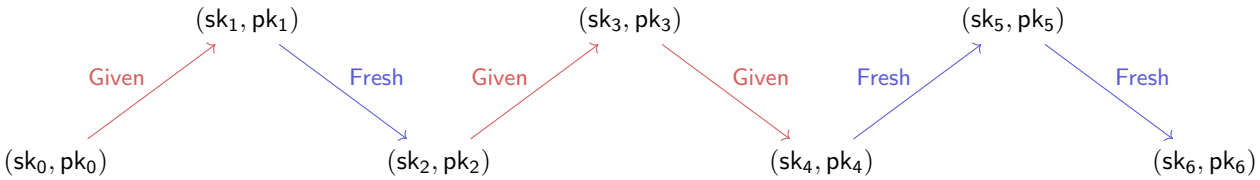


Figure 4.1: Representation of the updated keys

The first two security notions we will consider are the *INDistinguishability under quantum Chosen Plaintext Attack with Updatability* (IND-qCPA-U) and *INDistinguishability under quantum Chosen Ciphertext Attack with Updatability* (IND-qCCA-U). In IND-qCCA-U, adversaries have access to an additional *decryption oracle* $\mathcal{O}_{Dec}$, that decrypt the ciphertext ct given by the adversary.
We have given just below a modular description of two games with the IND-qCCA-U additions in red.

Game 4.1.2: IND-qCPA/qCCA-U

$\mathcal{G}_b^{\text{IND-qCPA/qCCA-U}}(\mathcal{A}_1, \mathcal{A}_2)$

1 : $i = 0$
2 : $\mathsf{Upd\_list} = \mathsf{Cor\_list} = \varnothing$
3 : $\mathsf{sk}_0, \mathsf{pk}_0 \xleftarrow{\$} \mathsf{KG}(1^\lambda)$
4 : $\mathsf{m}_0, \mathsf{m}_1, \mathsf{j}, \mathsf{st} \longleftarrow \mathcal{A}_1^{\mathsf{Fresh\_Upd}, \mathsf{Given\_Upd}, \mathsf{Corrupt}, \mathcal{O}_{Dec}}(\mathsf{pk}_0)$
5 : **if** $\mathsf{m}_0 = \mathsf{m}_1$ **return** $\perp$
6 : $ct_b \longleftarrow \mathsf{Enc}(\mathsf{pk_j}, \mathsf{m_b})$
7 : $d \longleftarrow \mathcal{A}_2^{\mathsf{Given\_Upd}, \mathsf{Fresh\_Upd}, \mathsf{Corrupt}, \mathcal{O}_{Dec}}(\mathsf{ct_b}, \mathsf{st})$
8 : **if** $\mathsf{IsFresh}(j)$ **do** :
9 : **return** $b = d$
10 : **return** $\perp$

$\mathsf{Fresh\_Upd}() \rightarrow \mathsf{pk_i}$

1 : $i = i + 1$ and $\mu \xleftarrow{\$} \mathsf{UG}(1^\lambda)$
2 : $(\mathsf{sk}_{i+1}, \mathsf{pk}_{i+1}) \xleftarrow{\$} (\mathsf{Usk}(\mathsf{sk_i}, \mu), \mathsf{Upk}(\mathsf{pk_i}, \mu))$
3 : **return** pk_{i+1}

$\mathsf{Given\_Upd}(\mu) \rightarrow \mathsf{pk_i}$

1 : $i = i + 1$
2 : $(\mathsf{sk}_{i+1}, \mathsf{pk}_{i+1}) \xleftarrow{\$} (\mathsf{Usk}(\mathsf{sk_i}, \mu), \mathsf{Upk}(\mathsf{pk_i}, \mu))$
3 : $\mathsf{Upd\_list} \longleftarrow \mathsf{Upd\_list} \cup \{(i, i+1)\}$
4 : **return** pk_{i+1}

$\mathcal{O}_{Dec}(k, \mathsf{c}) \rightarrow m$

1 : **if** $k = \mathsf{j}$ and $\mathsf{c} = \mathsf{ct}_b$, **return** $\perp$
2 : **return** $\mathsf{Dec}(\mathsf{sk_k}, \mathsf{c})$

$\mathsf{Corrupt}(j) \rightarrow \mathsf{sk_j}$

1 : $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{j\}$
2 : $i, k \leftarrow j$
3 : **while** $(i-1, i) \in \mathsf{Upd\_list}$ **do** :
4 : $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{i-1\}$ and $i = i - 1$
5 : **while** $(k, k+1) \in \mathsf{Upd\_list}$ **do** :
6 : $\mathsf{Cor\_list} = \mathsf{Cor\_list} \cup \{k+1\}$ and $k = k + 1$
7 : **return** $\mathsf{sk_j}$

$\mathsf{IsFresh}(j)$

1 : **return not** $j \stackrel{?}{\in} C$

Definition 4.1.3: IND-qCPA/qCCA-U **Secure**

An UPKE is **IND-qCPA/qCCA-U secure** if for any given $(\mathcal{A}_1, \mathcal{A}_2)$ quantum $\mathsf{poly}(\lambda)$ adversaries such that

$$\mathsf{Adv}^{\text{IND-CPA/qCCA-U}}(\mathcal{A}_1, \mathcal{A}_2) = \left| \mathbb{P}\left[\mathcal{G}_1^{\text{IND-CPA/qCCA-U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right] - \mathbb{P}\left[\mathcal{G}_0^{\text{IND-CPA/qCCA-U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right] \right| \leqslant \mathsf{negl}(\lambda)$$

We also work with a third security notion, *One-Wayness under quantum Plaintext Checking Attack with Updatability* (OW-qPCA-U). Here, instead of distinguishing between the ciphers of two chosen messages, the adversaries have to decrypt a challenge ciphertext. Additionally, adversaries in this game have access to $\mathcal{O}_{PCA}$ a *plaintext checking oracle* that receives a plaintext and a ciphertext and returns if the ciphertext is a valid encryption of the plaintext.

Game 4.1.4: OW-qPCA-U

$\mathcal{G}^{\mathsf{OW\text{-}qPCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2)$

1 : $i = 0, \quad \mathsf{Upd\_list} = \mathsf{Cor\_list} = \varnothing$
2 : $\mathsf{sk}_0, \mathsf{pk}_0 \xleftarrow{\$} \mathsf{KG}(1^\lambda)$
3 : $\mathsf{j}, \mathsf{st} \longleftarrow \mathcal{A}_1^{\mathsf{Fresh\_Upd}, \mathsf{Given\_Upd}, \mathsf{Corrupt}, \mathcal{O}_{PCO}}(\mathsf{pk}_0)$
4 : $\mathsf{m} \xleftarrow{\$} \mathcal{M}$
5 : $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk_j}, \mathsf{m})$
6 : $\mathsf{n} \longleftarrow \mathcal{A}_2^{\mathsf{Given\_Upd}, \mathsf{Fresh\_Upd}, \mathsf{Corrupt}, \mathcal{O}_{PCO}}(\mathsf{ct}, \mathsf{st})$
7 : **if** $\mathsf{IsFresh}(j)$ **do** :
8 : **return** $m \stackrel{?}{=} n$
9 : **return** $\perp$

$\mathcal{O}_{PCO}(\mathsf{m}, \mathsf{c}, \mathsf{pk_i}) \to b$

1 : **if** $\mathsf{m} \notin \mathcal{M}$ **do**
2 : **return** $\perp$
3 : **else do**
4 : **return** $\mathsf{m} \stackrel{?}{=} \mathsf{Dec}(\mathsf{sk_i}, \mathsf{c})$

Definition 4.1.5: OW-qPCA-U **Secure**

An UPKE is **OW-qPCA-U secure** if for any given $(\mathcal{A}_1, \mathcal{A}_2)$ quantum $\mathsf{poly}(\lambda)$ adversaries such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}qPCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = \mathbb{P}\left[\mathcal{G}^{\mathsf{OW\text{-}qPCA\text{-}U}}(\mathcal{A}_1, \mathcal{A}_2) = 1\right] \leqslant \mathsf{negl}(\lambda)$$

4.1.2 M-SIDH

Going back to isogenies and more precisely to SIDH, we have seen in section 2.3 how **EvalKani** could be used to solve the supersingular isogeny problem with torsion point information. We now present some countermeasures. To make the SIDH resistant to Kani's Lemma, the idea of [Fou22, FMP23] is to mask the torsion points, hence its name of Masked-SIDH (M-SIDH). The central idea comes from the following equality. let φ be an isogeny of degree coprime to m. Then

$$\Big\langle [a]\varphi(P) + [b]\varphi(Q) \Big\rangle = \Big\langle [a]([m]\varphi(P)) + [b]([m]\varphi(Q)) \Big\rangle$$

meaning that inside the SIDH, if instead of sending $\varphi_A(P), \varphi_A(Q)$, we send $[m]\varphi_A(P), [m]\varphi_A(Q)$ with a secret number m coprime to the degree of φ_A, then we could still compute the required pushforwards. Sadly, this is not so simple, as using Weil pairing, we have that $e_A\big(\phi(P), \phi(Q)\big) = e_A(P, Q)^{\deg \phi}$. If A is smooth, using discrete logarithm, we can recover $\deg \phi \mod A$. Applied to $[m]\phi$, this entails that we can recover $m^2 \deg(\phi) \mod A$ and thus $m^2 \mod A$. Finding the mask m is therefore equivalent to finding the right square root of m^2 in $\mathbb{Z}_A$. Thus, to be secure, we need to have a A such that $\mathbb{Z}_A$ has many roots of the unity, which means that $A = \prod_{i=1}^{n} p_i$ with n large and p_i distinct odd primes. This is the general idea behind the M-SIDH that we now describe as presented in [FMP23]. Let the M-SIDH public parameter be as follows:

- $p = ABf - 1$ a prime number such that with $A = \prod_{i=1}^{n_A} p_i$ and $B = \prod_{j=1}^{n_B} q_j$ coprime such that $A \simeq B$ and $n_A \simeq n_B$.
- E a supersingular curve defined over $\mathbb{F}_{p^2}$.
- $\langle P_A, Q_A \rangle$ a basis of $E[A]$.

- $\langle P_B, Q_B \rangle$ a basis of $E[B]$.

with both Alice and Bob that can efficiently sample at random over $\mu_2(A) = \{x \in \mathbb{Z}_A | x^2 = 1\}$ and $\mu_2(B)$.

M-SIDH

Alice(pp)		**Bob**(pp)
$s_A \leftarrow_\$ \mathbb{Z}_A, \alpha \leftarrow_\$ \mu_2(B)$		$s_B \leftarrow_\$ \mathbb{Z}_B, \beta \leftarrow_\$ \mu_2(A)$
$R_A \leftarrow P_A + [s_A]Q_A$		$R_B \leftarrow P_B + [s_B]Q_B$
$\phi_A, E_A \leftarrow$ **KernelToIsogeny**(E, R_A, A)		$\phi_B, E_B \leftarrow$ **KernelToIsogeny**(E, R_B, B)
$S_A \leftarrow [\alpha]\phi_A(P_B), T_A \leftarrow [\alpha]\phi_A(Q_B)$		$S_B \leftarrow [\beta]\phi_B(P_A), T_B \leftarrow [\beta]\phi_B(Q_A)$
	$E_A, S_A, T_A \rightarrow$	
	$\leftarrow E_B, S_B, T_B$	
$U_A \leftarrow S_B + [s_A]T_B$		$U_B \leftarrow S_A + [s_B]T_A$
$\psi_A, E_K \leftarrow$ **KernelToIsogeny**(E_B, U_A, A)		$\psi_B, E_K \leftarrow$ **KernelToIsogeny**(E_A, U_B, B)
$K \leftarrow \mathsf{KDF}(j(E_K))$		$K \leftarrow \mathsf{KDF}(j(E_K))$

It was proven in [FMP23] that the key security of M-SIDH reduces to the following problem with adequate N and d.

Problem 4.1.6: Supersingular isogeny problem with masked torsion point information

Let $\phi : E \to E'$ be an isogeny of degree d, let $\langle P, Q \rangle$ be a basis of $E[N]$ with $N = \prod_{i=1}^{n} p_i$ coprime to d and let $m \in \mu_2(N)$ be a random element.
Given $P, Q, [m]\phi(P), [m]\phi(Q)$, compute ϕ.

The rationale behind why masking provides protection against **EvalKani** comes from the fact that the torsion points we receive describe the isogeny $[m]\phi$ whose degree is greater than N. Intuitively, we would think that it suffice for n_A and n_B to be around λ each to ensure that we have $|\mu_2(A)| = 2^\lambda$, but this is not sufficient. This is because instead of needing to find $m \mod N$ it suffice to find $m \mod N_t$, with $N_t = \prod_{i=t}^{n} p_i$ such that $N_t \geqslant \sqrt{d}$. This is because we have enough torsion points on N_t to use **EvalKani** efficiently and thus retrieve ϕ. Then, as $m \in \mu_2(N)$, we have that $m \mod N_t \in \mu_2(N_t)$ with $|\mu_2(N_t)| = 2^{n-t}$, meaning that we have significantly diminished the numbers of possible masks. Taking N_t with only the biggest factors of N is the optimal solution to increase N_t while minimizing $|\mu_2(N_t)|$. Using this method, we get the following theorem.

Theorem 4.1.7: [FMP23, Theorem 7] Attack by using less torsion point

Let $\phi : E \to E'$ be an isogeny of degree d, $\langle P, Q \rangle = E[N]$ with $N = \prod_{i=1}^{n} p_i$ and define $N_t = \prod_{i=t}^{n} p_i$ with t minimal such that $N_t \geqslant \sqrt{d}$. Then, there exists an algorithm that solve the supersingular isogeny problem with masked torsion point information in $\tilde{O}(2^{n-t+1})$.

Theorem 4.1.7 induces that to ensure the security of M-SIDH, we need for A and B to be such that for all $A_t = \prod_{i=t}^{n_A} p_i$, we have that $A_t \geqslant \sqrt{B} \Rightarrow n_A - t \geqslant \lambda$ and similarly for B. As shown in [FMP23, section 7.3], the number of needed distinct prime divisors of p is around 4.5λ. M-SIDH is therefore significantly slower than SIDH as further supported in [LLC+23, section 5]. It was also shown in [FMP23, section

4.2] that M-SIDH was also insecure if the starting curve has a non-small endomorphism or a known endomorphism ring. Finally, it was shown in [CV23] that M-SIDH was insecure if the starting curve was defined over $\mathbb{F}_p$, as we can perform a lollipop attack.

4.1.3 Generalised lollipop

We briefly mentioned *lollipop attacks* when we talked about SIDH in section 2.1.2. They were originally introduced in [Pet17], improved in [dQKL+20] and used in [FP21] as attacks over the SIDH and supersingular isogeny problem with torsion point information. The original lollipop attack used the knowledge of the endomorphism ring over the starting curve of a secret isogeny φ. In the case of SIDH, this curve often was E_{1728}.
To retrieve φ, we choose a non-trivial endomorphism $\theta \in \mathrm{End}(E_{1728})$ and find $\mathbf{M}$ the matrix that represent the action of θ over the basis $\langle P, Q\rangle$ of $E_{1728}[N]$. We choose θ specifically such that the *lollipop* $\Sigma \in \mathrm{End}(E)$ defined as $\Sigma = \varphi \circ \theta \circ \widehat{\varphi} + [m]$ is of degree N. Seeing $E[N]$ over the basis $\langle \varphi(P), \varphi(Q)\rangle$, we have that the action of Σ is given by the matrix $[d]\mathbf{M} + [m]\mathbf{Id_2}$, meaning that we know $\ker(\Sigma)$. We then evaluate Σ over any points with **KernelToIsogeny**. By subtracting $[m]$, we can therefore evaluate $\varphi \circ \theta \circ \widehat{\varphi}$ over $E[d]$ and thus retrieve information on $\ker(\widehat{\varphi})$.

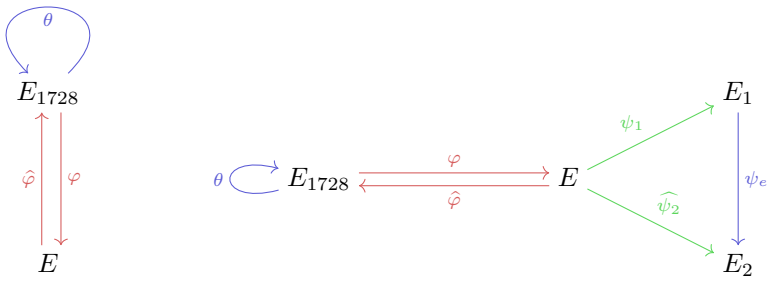
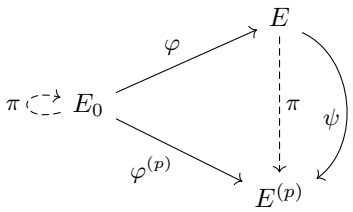


Figure 4.2: Examples of lollipop attacks. The left one is the original attack in [Pet17] while the second is taken from [FP21]. Red isogenies are secrets, while blue isogenies are chosen by the attacker.

We now present the *generalized lollipop attack*, an attack that works over M-SIDH. It is detailed in [CV23]. Its general idea is to use the fact that the domain of the mask isogeny φ is defined over $\mathbb{F}_p$ to construct a new unmasked isogeny ψ, and use **EvalKani** over ψ to retrieve $\ker(\psi)$ and extract $\ker(\varphi)$ from $\ker(\psi)$.
To be more specific, let $\varphi : E_0 \to E$ be an isogeny of degree d, with E_0 defined over $\mathbb{F}_p$. We set $\langle P, Q\rangle$ to be a basis of $E_0[N]$ and S, T to be the masked image of those points, i.e. $\binom{S}{T} = \mathbf{A}\varphi\binom{P}{Q}$ with $\mathbf{A} = \begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix}$.
We then consider the following diagram, where we denote as $\phi^{(p)}$ the map $\pi_*\phi$. Because E_0 is defined over $\mathbb{F}_p$, we have that $\pi \in \mathrm{End}(E_0)$ and its pushforward is well-defined.



We set $\psi = \varphi^{(p)} \circ \widehat{\varphi}$. We will then use the following lemma.

Lemma 4.1.8: [CV23, Lemma 3]

Using the above notation, assume that the matrix $\mathbf{M}_{\hat{\pi}}$ is such that

$$\hat{\pi}\begin{pmatrix} P \\ Q \end{pmatrix} = \mathbf{M}_{\hat{\pi}}\begin{pmatrix} P \\ Q \end{pmatrix}$$

Then, we have that

$$\psi\begin{pmatrix} S \\ T \end{pmatrix} = dp^{-1}\mathbf{M}_{\hat{\pi}}\pi\begin{pmatrix} S \\ T \end{pmatrix} \mod N$$

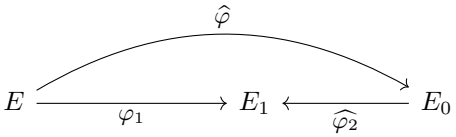
meaning that we can compute $\psi(E[N])$.

Proof of Lemma 4.1.8:
As $\varphi^{(p)} \circ \pi = \pi \circ \varphi$, we have that $[p]\varphi^{(p)} = \pi \circ \varphi \circ \hat{\pi}$. Thus using $\mathbf{M}_{\hat{\pi}} = \hat{\pi}|_{E_0[N]}$, we have that

$$\begin{aligned}
[p]\psi\begin{pmatrix} S \\ T \end{pmatrix} &= [p]\varphi^{(p)} \circ \widehat{\varphi}\begin{pmatrix} S \\ T \end{pmatrix} = [d]\mathbf{A}\left([p]\varphi^{(p)}\begin{pmatrix} P \\ Q \end{pmatrix}\right) \\
&= [d]\mathbf{A}\left(\pi \circ \varphi \circ \hat{\pi}\begin{pmatrix} P \\ Q \end{pmatrix}\right) = [d]\mathbf{A}\mathbf{M}_{\hat{\pi}}\left(\pi \circ \varphi\begin{pmatrix} P \\ Q \end{pmatrix}\right) \\
&= [d]\mathbf{A}\mathbf{M}_{\hat{\pi}}\mathbf{A}^{-1}\pi\begin{pmatrix} S \\ T \end{pmatrix} = [d]\mathbf{M}_{\hat{\pi}}\pi\begin{pmatrix} S \\ T \end{pmatrix} \\
\text{i.e. } \psi\begin{pmatrix} S \\ T \end{pmatrix} &= (dp^{-1} \bmod N)\mathbf{M}_{\hat{\pi}}\pi\begin{pmatrix} S \\ T \end{pmatrix} = d\mathbf{M}_{\pi}^{-\mathbf{1}}\begin{pmatrix} S \\ T \end{pmatrix}
\end{aligned}$$

□ 4.1.8

As we can evaluate ψ over $E[N]$ and we have that $\deg(\psi) = d^2 \leqslant N^2$, we can use **EvalKani** over ψ to evaluate ψ over any points and in particular over $E[d]$. By definition, we have that $\ker(\widehat{\varphi}) \subseteq \ker(\psi)[d] = \ker(\psi) \cap E[d]$, but we do not necessarily have an equality, meaning that we have not yet found $\ker(\widehat{\varphi})$. Nevertheless, we have gained a substantial among of information. Indeed, consider d' the biggest divisor of d such that $\ker(\psi)[d'] = E[d']$. We have that $[d']\big(\ker(\psi)[d]\big)$ is a cyclic group of size d/d', meaning that using **KernelToIsogeny** over a generator of $[d']\big(\ker(\psi)[d]\big)$ will give us $\varphi_1 : E \to E_1$ a component of degree d/d' of $\widehat{\varphi}$



We have retrieved φ_1, it therefore remains to retrieve φ_2, of degree d'. The fact that $\ker(\psi)[d'] = E[d']$ induces that

$$\ker(\varphi)[d'] = \ker(\varphi^{(p)})[d'] = \pi\Big(\ker(\varphi)[d']\Big)$$

i.e. $\ker(\varphi)[d']$ is an eigenspace of $\hat{\pi}$ over $E[d']$. This information is especially useful as this tremendously reduces the possibilities for φ_2. We will only detail here the case $d' = q^\ell$ with q a prime number. The generalized case is detailed in [CV23, section 3.2]. We have three distinct scenarios that depend on how q behave in $\mathbb{Z}[\hat{\pi}] \cong \mathbb{Z}[\sqrt{-p}]$. See [Sam13] for more details in the reason behind this partition.

1. q is *inert* in $\mathbb{Z}[\sqrt{-p}]$, meaning that $\left(\frac{-p}{q}\right) = -1$. In that case, we have that p remains prime and thus that $\hat{\pi}$ is not diagonalizable over $E[d']$, meaning that $d' = 1$ and that $\ker(\psi)[d] = \ker(\hat{\varphi})$.

2. q is *decomposed* in $\mathbb{Z}[\sqrt{-p}]$, meaning that $\left(\frac{-p}{q}\right) = 1$. We have that $\widehat{\pi}$ has two distinct eigenvalues over $E[d']$, i.e. two distinct eigenspaces. We therefore have restricted the possibilities to two eigenvectors and can construct the corresponding isogenies φ_2^1 and φ_2^2. The valid φ_2 is the one with the same codomain as φ_1.

3. q is *ramified* in $\mathbb{Z}[\sqrt{-p}]$, meaning that $\left(\frac{-p}{q}\right) = 0$. In this case, π has one eigenvalue with an eigenspace of dimension two, and we gain no information about isogeny φ_2. This scenario thankfully does not occur when φ is separable.

Using this method, we get the following result.

Theorem 4.1.9: [CV23, section 4] M-SIDH Generalized Lollipop Attacks

Let $\varphi : E_0 \to E$ be an isogeny of degree d with E_0 defined over $\mathbb{F}_p$ and let P, Q be a basis of $E_0[N]$, with N smooth such that $N \geqslant d$. Then, there exists an algorithm named **GeneralisedLollipop** that efficiently solve the supersingular isogeny problem with masked torsion point information over these parameters.

The implications of generalized lollipops are not restricted to M-SIDH and can also be used over other cryptosystems such as FESTA [BMP23] and CSIDH [CLM+18]. Nevertheless, this attack does not significantly lower the assumption that the supersingular isogeny problem with masked torsion point information is hard over random curves, as the probability that a random supersingular curve is defined in $\mathbb{F}_p$ is in $\Theta(p^{-1/2})$, which is negligible.

4.2 PKE from M-SIDH attacks

The core concept behind SILBE is to leverage the generalized lollipop attack and the **GeneralisedLollipop** algorithm as a deciphering mechanism, akin to how the original lollipop attack was employed in designing SETA [FdSGF+19]. This endeavor will make usage of all the different isogeny representations that we detailed in chapter 2. SILBE is in fact related to [CV23, section 4.3] and the idea of M-SIDH with trapdoor curves, although there are substantial changes that we now detail.
The underlying architecture behind the PKE part of SILBE is given in figure 4.3. It works as follows:

- KG: Alice computes a long isogeny between E_{1728} and E_A. Using **EvalKani**, it retrieves the representing ideal I and use **RandomEquivalentIdeal** to find a short connecting isogeny $\phi_A : E_{1728} \to E_A$. E_A is then used as the public key while ϕ_A is the secret key.
- Enc: Bob computes $\phi_B : E_A \to E_B$ an isogeny. It then sends the masked image by ϕ_B of a basis $E_A[N]$, with the mask is the message m.
- Dec: Using its knowledge of ϕ_A, Alice uses **GeneralisedLollipop** over $\phi_B \circ \phi_A$ to retrieve $\ker(\widehat{\phi_B})$ and using the discrete logarithm, it retrieves m.

The public parameters of SILBE are constructed as such.

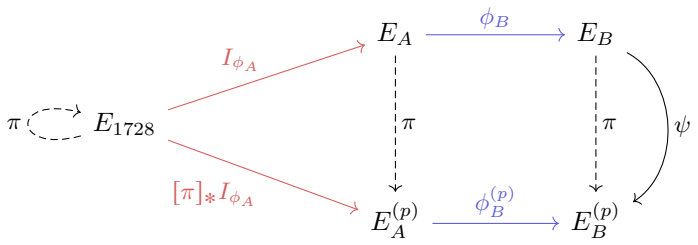


Figure 4.3: Diagram of the PKE part of SILBE, Alice in red and Bob in blue

Algorithm 14 SILBE.Setup

Input: 1^λ

Output: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$ with p a prime, $\langle P_0, Q_0\rangle = E_{1728}[N]$, $\langle U_0, V_0\rangle = E_{1728}[3^\beta]$, $\mathbf{M}_\pi \in \mathrm{GL}_2(N)$ and t an integer.

1: Take p a prime of the form $p = 3^\beta N f + 1$ such that $p = 3 \mod 4$ and $N = \prod_{i=1}^n p_i$ with p_i distinct odd small prime numbers such that $N \geqslant 3^\beta p^{1/2} \log(p)^2$, N is coprime to 3 and n big enough such that for all $N_k = \prod_{i=k}^n p_i$, we have that $N_k \geqslant \sqrt{3^\beta} \Rightarrow n - k \geqslant \lambda$.
2: $P_0, Q_0 \leftarrow$ **CanonicalTorsionBasis**(E_{1728}, N)
3: $U_0, V_0 \leftarrow$ **CanonicalTorsionBasis**$(E_{1728}, 3^\beta)$
4: $\mathbf{M}_\pi \leftarrow$ **EvalImageMatrix**$(E_{1728}, P_0, Q_0, \pi(P_0), \pi(Q_0))$.
5: $t \leftarrow \left\lceil \frac{2\log_2(p)}{\beta \log_2(3)} \right\rceil$
6: $\mathsf{pp} \leftarrow \big(p, N, P_0, Q_0, U_0, V_0, \mathbf{M}_\pi, t\big)$.
7: **return** pp

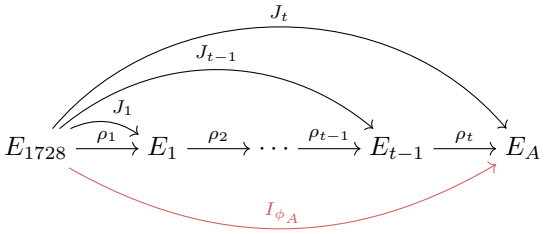
4.2.1 Key generation

As touched earlier, the key generation of SILBE constructs a long isogeny walk with starting curves E_{1728}. This is done to use the following proposition.

Proposition 4.2.1: [DLRW23, Proposition B.2.1]

Let $\phi : E \to E'$ be an ℓ^h-isogeny obtained from a non-backtracking random ℓ-isogeny walk over $\mathcal{G}_p^\ell$. Then, for all $\epsilon \in]0, 2]$, the distribution of E' has statistical distance $O(p^{-\epsilon/2})$ to the uniform distribution in the supersingular isogeny graph, provided that $h \geqslant (1+\epsilon)\log_\ell(p)$.

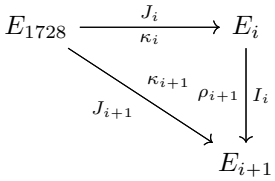
By constructing a path of length t of 3^β-isogenies $\rho_1, \cdots, \rho_t$, we get that the degree of their composition is $O(p^2)$ and the end curve distribution will be $O(p^{-1/2})$ statistically close from the uniform distribution, meaning that it will be computationally indistinguishable from an uniform random sampling. We call the end curve E_A.
We then want to compute $I_1, \cdots I_t$ the ideals corresponding to $\rho_1, \cdots, \rho_t$. This is done using the following recursive mechanism:



1. Assume knowledge of $\kappa_i : E_{1728} \to E_i$ together with its representative ideal J_i such that $n(J_i)$ is prime and coprime to 3. Furthermore, assume knowledge of $\mathfrak{O}_{E_i}$ a T-evaluation basis over E_i with $T \neq 3$ prime. Finally, assume knowledge of I_j with $1 \leqslant j \leqslant i$.

2. Using **KernelToIsogeny**, we can construct ρ_{i+1} and find E_{i+1} and using $\mathfrak{O}_{E_i}$ we can find I_{i+1} with the **KernelToIdeal**.

3. Then, we have that $J_i I_{i+1}$ is a $(\mathcal{O}_{1728}, \mathcal{O}_{E_{i+1}})$-ideal, using **RandomEquivalentIdeal**, we find an ideal J_{i+1} such that $n(J_i) \neq n(J_{i+1})$ and $n(J_{i+1}) \in [\sqrt{p}\log(p), \sqrt{p}\log(p)]$ is prime. Furthermore, to speed-up computations, we consider $\widetilde{N} = \prod_{i=1}^{x} p_i$ with x minimal such that $\widetilde{N} \geqslant p^{1/4}\log(p)^{1/2}$ and ask for $\widetilde{N}^2 - n(J_i)$ to be prime and equal to $1 \mod 4$.



4. Now, using **EvalTorsion** over the above triangle, we evaluate $\kappa_{i+1} = \phi_{J_{i+1}}$ over $\langle P_0, Q_0 \rangle = E_{1728}[N]$. We then have constructed a HD representation of κ_{i+1}.

5. Using **ConstructKani** over $(P_0, Q_0, \kappa_{i+1}(P_0), \kappa_{i+1}(Q_0))$ in dimension 4 thanks to $\tilde{N}$, we get a Kani's isogeny F_{i+1} and can therefore evaluate κ_{i+1} over any points. This is then used to apply the **PushEndRing** over κ_{i+1} and J_{i+1} to retrieve $\mathfrak{O}_{E_{i+1}}$ a $n(J_{i+1})$-evaluation basis over E_{i+1}.

Using this mechanism, we compute I_i for $i = 1, \cdots, t$. Additionally, we also compute $\mathfrak{O}_{E_A}$ a $n(J_t)$-evaluation basis of $\mathrm{End}(E_A)$. To speed up the decryption part of SILBE, we use **RandomEquivalentIdeal** over J_t to find another $(\mathcal{O}_{1728}, \mathcal{O}_{E_A})$-ideal I_{ϕ_A} such that $N' - n(I_{\phi_A})^2 3^{2\beta} = 1 \mod 4$ and is a prime number, with $N' = p_1 \cdot \prod_{i=2}^{n} p_i^2$. This ensures that the **EvalKani** in **GeneralisedLollipop** is performed in dimension 4. The reason behind the choice of N' and not N^2 comes from the fact $N^2 - n(I_{\phi_A})^2 3^{2\beta} = (N - n(I_{\phi_A})3^{\beta})(N + n(I_{\phi_A})3^{\beta})$ and can therefore never be prime.
Once found, we use **EvalTorsion** over $\rho_t \circ \cdots \circ \rho_1$ and $I_1 \cdots I_t$ to evaluate $\phi_A\binom{P_0}{Q_0}$ and, using a small subroutine based on Weil's pairing named **EvalImageMatrix**, we compute the matrix $\mathbf{M}_{\phi_\mathbf{A}}$ such that $\phi_A\binom{P_0}{Q_0} = \mathbf{M}_{\phi_\mathbf{A}}\binom{P_A}{Q_A}$.
We then set E_A as the public key and $\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}$ as the secret key. We construct ρ_i in such a way that our walk cannot be backwards. To do so, we use U_i, V_i a basis of $E_i[3^\beta]$ such that $\rho_i(E_{i-1}[3^\beta]) = \langle V_i \rangle$. As we set $\ker(\rho_{i+1}) = \langle U_i + [\eta_{i+1}]V_i \rangle$, we have that it can be any cyclic isogeny of degree 3^β except $\widehat{\rho_i}$.

Algorithm 15 SILBE.KG

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$
Output: pk, sk a public/secret key pair.

1: $E_0 \leftarrow E_{1728} \qquad J_0 \leftarrow \mathcal{O}_{1728} \qquad \mathfrak{O}_0 \leftarrow \mathfrak{O}_{1728}$
2: **for** $1 \leqslant i \leqslant t$ **do**
3: Sample $\eta_i \in_\$ \mathbb{Z}_{3^\beta}$.
4: $E_i, \rho_i \longleftarrow \textbf{KernelToIsogeny}\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1}), 3^\beta\big)$ ▷ Already in pp if $i = 1$.
5: $I_i \longleftarrow \textbf{KernelToIdeal}\big(\mathfrak{O}_{E_{i-1}}, (U_{i-1} + [\eta_i]V_{i-1})\big)$
6: Deterministically compute U_i, V_i a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
7: $J_i \longleftarrow \textbf{RandomEquivalentIdeal}(J_{i-1}I_i)$
8: **if** $n(J_i) = n(J_{i-1})$ **or** and $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ **or** is not prime **do** go back to line 7.
9: $S_i, T_i \longleftarrow \textbf{EvalTorsion}(\mathfrak{O}_{1728}, \rho_i \circ \kappa_{i-1}, J_{i-1}I_i, id, J_i, \{P_0, Q_0\})$
10: $F_i \longleftarrow \textbf{ConstructKani}\big(n(J_i), \widetilde{N}, \widetilde{N}, (P_0, Q_0, S_i, T_i)\big)$
11: $\mathfrak{O}_{E_i} \longleftarrow \textbf{PushEndRing}(\mathfrak{O}_{1728}, \kappa_i, J_i)$ ▷ $\kappa_i \leftarrow F_i(0, 0, -, 0)_3$
12: $I_{\phi_A} \longleftarrow \textbf{RandomEquivalentIdeal}\big(J_t\big)$
13: **if** $N' - n(I_{\phi_A})^2 3^{2\beta} \neq 1 \mod 4$ **or** is not prime **do** go back to line 12.
14: $K, L \longleftarrow \textbf{EvalTorsion}(\mathcal{O}_{1728}, \rho_t \circ \cdots \circ \rho_1, I_1 \cdots I_t, 1, I_{\phi_A}, P_0, Q_0)$
15: $\mathbf{M}_{\phi_\mathbf{A}} \longleftarrow \textbf{EvalImageMatrix}(E_t, N, P_t, Q_t, K, L)$
16: $\mathsf{pk} \longleftarrow \big(E_t = E_A\big)$
17: $\mathsf{sk} \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}\big)$
18: **return** pk, sk.

Algorithm 16 EvalImageMatrix

Input: (E, N, P, Q, X, Y) with E a curve, N smooth integer, $\langle P, Q \rangle = E[N]$ and $X, Y \in E[N]$.
Output: $\mathbf{M}$ such that $\binom{X}{Y} = \mathbf{M}\binom{P}{Q}$.

1: $w_0 \leftarrow e_N(P, Q)$
2: $w_1 \leftarrow e_N(P, X)$
3: $w_2 \leftarrow e_N(P, Y)$
4: $w_3 \leftarrow e_N(X, Q)$
5: $w_4 \leftarrow e_N(Y, Q)$
6: $v_{1,1} \leftarrow \textbf{discretelog}(w_0, w_3, N)$
7: $v_{1,2} \leftarrow \textbf{discretelog}(w_0, w_1, N)$
8: $v_{2,1} \leftarrow \textbf{discretelog}(w_0, w_3, N)$
9: $v_{2,2} \leftarrow \textbf{discretelog}(w_0, w_2, N)$
10: **return** $\begin{pmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{pmatrix}$

4.2.2 Encryption & decryption

Encryption

As explained, the message space of SILBE is $\mu_2(N) = \big\{x \in \mathbb{Z}_N \big| x^2 = 1\big\}$. As $N = \prod_{i=1}^n p_i$, we have that $|\mu_2(N)| = 2^n$ and we can furthermore construct an efficient mapping between $\{0,1\}^n$ and $\mu_2(N)$ using the Chinese remainder theorem. To encrypt m, Bob starts to compute a random isogeny $\phi_B : E_A \to E_B$ of degree 3^β. Then, similarly to M-SIDH, we compute the image of the N torsion points through this isogeny and mask those points using the message m. The ciphertext is therefore $E_B, [m]\phi_B(P), [m]\phi_B(Q)$.

Algorithm 17 SILBE.Enc

Input: $\mathsf{pp}, \mathsf{pk}, \mathsf{m} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), E_A$ with $\mathsf{m} \in \mu_2(N)$
Output: $\mathsf{ct} = (E_B, R_1, R_2)$ with $R_1, R_2 \in E_B[N]$.

1: $P_A, Q_A \longleftarrow$ **CanonicalTorsionBasis**(E_A, N)
2: $U_A, V_A \longleftarrow$ **CanonicalTorsionBasis**$(E_A, 3^\beta)$
3: Sample $r_B \in_\$ \mathbb{Z}_{3^\beta}$
4: $E_B, \phi_B \longleftarrow$ **KernelToIsogeny**$\big(E_A, (U_A + [r_B]V_A), N\big)$
5: $\binom{R_1}{R_2} \longleftarrow [\mathsf{m}]\phi_B\binom{P_A}{Q_A}$
6: $\mathsf{ct} \longleftarrow (E_B, R_1, R_2)$
7: **return** ct

Decryption

As previously stated, we use the **GeneralisedLollipop** over $\phi_B \circ \phi_A$ to decipher our message. Indeed, using the torsion points in ct, we can define $\binom{S}{T} = [\mathsf{m}]\phi_B \circ \phi_A\binom{P_0}{Q_0}$. Theses points are easily computable using sk as

$$[\mathsf{m}]\phi_B \circ \phi_A\begin{pmatrix}P_0\\Q_0\end{pmatrix} = [\mathsf{m}]\mathbf{M}_{\phi_\mathbf{A}}\phi_B\begin{pmatrix}P_A\\Q_A\end{pmatrix} = \mathbf{M}_{\phi_\mathbf{A}}\begin{pmatrix}R_1\\R_2\end{pmatrix}$$

We modify the **GeneralisedLollipop** algorithm of [CV23] such that it just computes $\ker(\widehat{\phi_B})$ and not the whole $\ker(\widehat{\phi_B \circ \phi_A})$. This speed up the decryption.
We consider the following isogeny

$$\psi : E_B \longrightarrow E_B^{(p)}$$

$$\psi = (\phi_B \circ \phi_A)^{(p)} \circ \phi_A \circ \phi_B = \phi_B^{(p)} \circ \phi_A^{(p)} \circ \phi_A \circ \phi_B$$

Using lemma 4.1.8, we can evaluate ψ over $E_B[N]$ as

$$\psi\begin{pmatrix}S\\T\end{pmatrix} = n(I_{\phi_A})3^\beta {\mathbf{M}_\pi}^{-1}\pi\begin{pmatrix}S\\T\end{pmatrix} = n(I_{\phi_A})3^\beta {\mathbf{M}_\pi}^{-1}\mathsf{M}_{\phi_\mathsf{A}}\pi\begin{pmatrix}R_1\\R_2\end{pmatrix}$$

We then use **EvalKani** over ψ to evaluate $\widehat{\psi}$ over $E_B^{(p)}[3^\beta]$. Due to the nature of $N' - n(I_{\phi_A})^2 3^{2\beta}$, this is done in dimension 4. Note that it is more efficient to compute two HD-isogenies instead of three, using the method we detailed in section 2.3.2.
We then have that $\widehat{\psi}(E_B^{(p)}[3^\beta]) = \ker(\psi)[3^\beta] = \ker(\widehat{\phi_B})$. The reason comes from our good choice of public parameters, as $p - 1 = 0 \mod 3$ and thus $\left(\frac{-p}{3}\right) = -1$, meaning that 3 is inert inside $\mathbb{Z}[\sqrt{-p}]$ and in $\mathbb{Z}[\sqrt{\chi}]$ with $\chi \in \mathrm{End}(E)$ a lollipop endomorphism defined as

$$\chi = \widehat{\pi} \circ \phi_A^{(p)} \circ \widehat{\phi_A} = [-1]\phi_A \circ \pi \circ \widehat{\phi_A} \text{ such that } \chi^2 = [-p(\deg \phi_A)^2]$$

We thus know $\ker(\widehat{\phi_B})$, so we can thus use **KernelToIsogeny** to compute $\widehat{\phi_B}(R_1) = [\mathsf{m}3^\beta]P_A$ and retrieve m using the discrete logarithm over $E[N]$.

Algorithm 18 SILBE.Dec

Input: $\mathsf{pp}, \mathsf{sk}, \mathsf{ct} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), (\mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}), (E_B, \mathsf{R}_1, \mathsf{R}_2)$
Output: m

$P_A, Q_A \longleftarrow$ **CanonicalTorsionBasis**(E_A, N)
$U_B, V_B \longleftarrow$ **CanonicalTorsionBasis**$(E_B^{(p)}, 3^\beta)$
$\binom{S}{T} \longleftarrow \mathbf{M}_{\phi_\mathbf{A}} \binom{R_1}{R_2}$
$\binom{K}{L} \longleftarrow [n(I_{\phi_A})3^\beta]\mathbf{M}_\pi^{-\mathbf{1}} \pi \binom{S}{T}$
$G, H \longleftarrow$ **EvalKani**$\big(n(I_{\phi_A})^2 3^{2\beta}, N, N/p_1, (S, T, K, L), U_B, V_B\big)$ ▷ $\widehat{\psi}(P) = F(P, 0, 0, 0)_1$
$\widehat{\phi_B} \longleftarrow$ **KernelToIsogeny**$(E_B, G + H, 3^\beta)$ ▷ if $G = H$, take G
return $(3^\beta)^{-1} \cdot \big(\textbf{discretelog}(P_A, \widehat{\phi_B}(R_1), N)\big) \mod N$

4.2.3 Security analysis

First and foremost, we see that SILBE is not IND-CPA secure. Indeed, To distinguish between two known message m_0 and m_1, we simply have to multiply R_1 and R_2 by m_0 and use **EvalKani** in dimension 8. If we are able to retrieve ϕ_B, then this means that the encrypted message was m_0, as that would induces that $[m_0]R_1 = [m_0^2]\phi_B(P) = \phi_B(P)$. Otherwise, this means that the encrypted message was m_1 with overwelming probability. That mechanism can be used to know if a ciphertext ct is the encryption of a plaintext m or not. This induces that any adversary of SILBE can simulate the oracle $\mathcal{O}_{PCO}$. This will be useful in the following proposition.

Proposition 4.2.2:

The security of SILBE as an OW-qPCA PKE reduces to the supersingular isogeny problem with masked torsion point information over random curves.

Proof of Proposition 4.2.2:
Using the previously explained method to simulate $\mathcal{O}_{PCO}$, we have that

$$\text{SILBE is OW-qPCA secure} \iff \text{SILBE is OW-qCPA secure}$$

Following proposition 4.2.1, we have that the distribution of the public key E_A is $O(p^{-1/2})$ close from the uniform distribution over supersingular curves, meaning that it is computationally indistinguishable. Let $\mathcal{A}^{\mathsf{OW-qCPA}}$ be any adversary for SILBE. We can then construct an algorithm $\mathcal{B}$ that solve the supersingular isogeny problem with masked torsion point information (SSIPMTI) over random curves with the same advantage. $\mathcal{B}$ is defined as such:

1. $\mathcal{B}$ receives as input (P, Q, S, T) with P, Q the canonical basis of $E[N]$ and $\binom{S}{T} = [\mathsf{m}]\varphi\binom{P}{Q}$ with $\varphi : E \to E'$ an isogeny of degree 3^β.
2. It then calls $\mathcal{A}^{\mathsf{OW\text{-}qCPA}}\big(E, (E', S, T)\big)$ and receive $\mathsf{n} \in \mu_2(N)$.
3. It then compute $[\mathsf{n}]S, [\mathsf{n}]T$ and use **EvalKani** in dimension 8 over theses points to retrieve $\ker(\varphi)$. As 3^β is smooth, using **KernelToIsogeny**, it can compute φ.

We see that if $\mathcal{A}^{\mathsf{OW-qCPA}}$ succeeds, then so does $\mathcal{B}$, meaning that

$$\mathbb{P}[\mathcal{B} \text{ solve the SSIPMTI}] \geqslant \mathsf{Adv}^{\mathsf{OW\text{-}qCPA}}(\mathcal{A}^{\mathsf{OW\text{-}qCPA}})$$

Thus, under the assumption that the supersingular isogeny problem with masked torsion point information over random curves is hard, then SILBE is OW-qPCA secure. To make it IND-qCCA in the ROM, we can use the $U^{\not\perp}$ variant of the Fujisaki-Okamoto transform, as detailed in [JZC+17, section 4.2].

4.3 Updatability

SILBE is thus OW-qPCA secure and can be made IND-qCCA. We can thus construct PKE from the generalised lollipop attack. We now make of SILBE an UPKE. The idea behind SILBE key update mechanism comes from the fact that our key generation mechanism has two excellent properties, namely that it can be adapted to start over any curve E, provided that we know an isogeny $\phi : E_{1728} \to E$ and that finding the public key can be done by just using **KernelToIsogeny**, without knowledge of $\phi : E_{1728} \to E$.

4.3.1 Design

Our update mechanism is therefore an adaptation of the key generation and is done as such:

- UG: Generate a seed $\mu \in \{0,1\}^{4\log(p)}$.
- Upk: Use a hash function over μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to create kernels of an isogeny walk starting at the public key E_A. Thanks to **KernelToIsogeny**, we compute the end curve of that walk, defined as E'_A, the updated public key.
- Usk: Use a hash function over μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to create kernels of an isogeny walk starting at the public key E_A. Thanks to **KernelToIsogeny**, we compute the end curve of that walk, defined as E'_A. Using the knowledge of $\phi_A : E_{1728} \to E_A$, we construct, using **EvalKani** and **RandomEquivalentIdeal** an isogeny $\phi'_A : E_{1728} \to E'_A$, the updated secret key.

The underlying architecture of the key update mechanism of part of SILBE is given in figure 4.4.

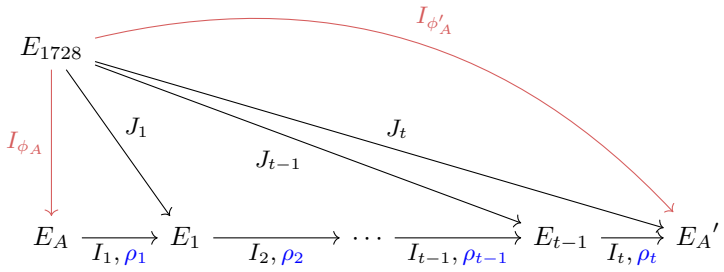


Figure 4.4: Diagram of the key update mechanism of SILBE, Alice in red and Bob in blue. Black isogenies are used for the construction of **SILBE.Usk**.

Algorithm 19 SILBE.UG

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big)$
Output: μ an update.

1: Sample $\mu \in_\$ \{0,1\}^{4\log(p)}$
2: **return** μ

Algorithm 20 SILBE.Upk

Input: $\mathsf{pp}, \mathsf{pk}, \mu = \big(p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t\big), E_A$.
Output: pk' the updated public key.

1: $E_0 \leftarrow E_A \qquad U_0, V_0 \longleftarrow$ **CanonicalTorsionBasis**$(E_A, 3^\beta)$
2: $(\eta_1, \cdots, \eta_t) \leftarrow H(\mu)$ ▷ $\eta_i \in \mathbb{Z}_{3^\beta}$
3: **for** $1 \leqslant i \leqslant t$ **do**
4: $\quad E_i, \rho_i \longleftarrow$ **KernelToIsogeny**$\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1}), 3^\beta\big)$
5: $\quad$ Deterministically compute U_i, V_i a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
6: $\mathsf{pk}' \leftarrow E_t = E'_A$
7: **return** pk'

Algorithm 21 SILBE.Usk

Input: $\mathsf{pp}, \mathsf{sk}\ , \mu = (p, (P_0, Q_0), (V_0, U_0), \mathbf{M}_\pi, t), \mathfrak{O}_{E_A}, I_{\phi_A}, \mathbf{M}_{\phi_\mathbf{A}}$
Output: sk' the updated secret key.

1: $E_0 \leftarrow E_A \qquad J_0 \leftarrow I_\phi \qquad U_0, V_0 \leftarrow$ **CanonicalTorsionBasis**$(E_A, 3^\beta)$
2: $(\eta_1, \cdots, \eta_t) \leftarrow H(\mu)$. ▷ $\eta_i \in \mathbb{Z}_{3^\beta}$
3: **for** $1 \leqslant i \leqslant t$ **do**
4: $\quad E_i, \rho_i \longleftarrow$ **KernelToIsogeny**$\big(E_{i-1}, (U_{i-1} + [\eta_i]V_{i-1}), 3^\beta\big)$
5: $\quad I_i \longleftarrow$ **KernelToIdeal**$\big(\mathfrak{O}_{E_{i-1}}, (U_i + [\eta_i]V_i)\big)$
6: $\quad$ Deterministically compute U_i, V_i a basis of $E_i[3^\beta]$ with $\langle V_i \rangle = \rho_i(E_{i-1}[3^\beta])$.
7: $\quad J_i \longleftarrow$ **RandomEquivalentIdeal**$(J_{i-1}I_i)$
8: $\quad$ **if** $n(J_i) = n(J_{i-1})$ **or** and $\widetilde{N}^2 - n(J_i) \neq 1 \mod 4$ **or** is not prime **do** go back to line 7.
9: $\quad S_i, T_i \longleftarrow$ **EvalTorsion**$(\mathfrak{O}_{1728}, \rho_i \circ \kappa_{i-1}, J_{i-1}I_i, id, J_i, P_0, Q_0)$ ▷ Use $\mathbf{M}_\phi$ if $i = 1$
10: $\quad F_i \longleftarrow$ **ConstructKani**$(n(J_i), \widetilde{N}, \widetilde{N}, (P_0, Q_0, S_i, T_i))$
11: $\quad \mathfrak{O}_{E_i} \longleftarrow$ **PushEndRing**$(\mathfrak{O}_{1728}, \kappa_i, J_i)$ ▷ $\kappa_i = F(0,0,-,0)_3$
12: $I_{\phi'} \longleftarrow$ **RandomEquivalentIdeal**$\big(J_t\big)$
13: **if** $N' - n(I_{\phi'_A})^2 3^{2\beta} \neq 1 \mod 4$ **or** is not prime **do** go back to line 12.
14: $K, L \longleftarrow$ **EvalTorsion**$(\mathfrak{O}_{1728}, \kappa_t, J_t, id, I_{\phi'}, P_0, Q_0)$
15: $\mathbf{M}_{\phi'} \longleftarrow$ **EvalImageMatrix**(E_t, N, P_t, Q_t, K, L)
16: $\mathsf{sk}' \longleftarrow \big(\mathfrak{O}_{E_t}, I_{\phi'}, \mathbf{M}_{\phi'}\big)$
17: **return** sk'.

To summarize SILBE:

- **SILBE.Setup**: We find the adequate β and N to construct a base prime $p = 3^\beta N f + 1$ such that $p = 3 \mod 4$ and $N = \prod_{i=1}^n p_i$ with n big enough such that it resists theorem 4.1.7. We also compute P_0, Q_0 a basis of $E[N]$ and U_0, V_0 a basis of $E_{1728}[3^\beta]$. We compute a matrix $\mathbf{M}_\pi$ that represent the action of π over P_0, Q_0 a basis of $E_{1728}[N]$.
- **SILBE.KG**: Sample a uniformly random isogeny $\phi : E_{1728} \to E_A$ of degree $3^{\beta t} \simeq p^2$. Recover the

endomorphism ring $\mathcal{O}_{E_A}$ of E_A. Compute a short ideal I_{ϕ_A} connecting $\mathcal{O}_{1728}$ and $\mathcal{O}_{E_A}$. Let ϕ_A be the isogeny corresponding to I_{ϕ_A}. The secret key is ϕ_A while the public key is E_A.

- **SILBE.Enc**: Construct an isogeny $\phi_B : E_A \to E_B$ of degree 3^β. Evaluate ϕ_B over P_A, Q_A the canonical basis of $E_A[N]$ and mask the image with the message $\mathsf{m} \in \mu_2(N)$. The ciphertext is the curve E_B together with the masked image of P_A and Q_A through ϕ_B.
- **SILBE.Dec**: Using the knowledge of ϕ_A, we apply the **GeneralisedLollipop** over $\phi_B \circ \phi_A$ to retrieve $\ker(\widehat{\phi_B})$. We then evaluate $\widehat{\phi_B}$ over a masked image of ϕ_A and retrieve the message m using discrete logarithms.
- **SILBE.UG**: Generate a random seed $\mu \in \{0,1\}^{4\log(p)}$.
- **SILBE.Upk**: Hash μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to construct an isogeny $\rho : E_A \to E'_A$ of degree $3^{t\beta} \simeq p^2$. E'_A is the updated public key.
- **SILBE.Usk**: Hash μ to generate a sequence of elements in $\mathbb{Z}_{3^\beta}$. Use this sequence to construct an isogeny $\rho : E_A \to E'_A$ of degree $3^{t\beta} \simeq p^2$. Recover the endomorphism ring $\mathcal{O}_{E'_A}$ of E'_A. Compute a short ideal I'_A connecting $\mathcal{O}_{1728}$ and $\mathcal{O}_{E'_A}$. Let ϕ'_A be the isogeny corresponding to I_A. The updated secret key is ϕ'_A.

4.3.2 Security analysis

The reason behind the fact that SILBE remains secure as an UPKE comes from the fact that, in the ROM, we have that **SILBE.Upk** is a one way mechanism such that the distribution of the updated public key E'_A is statistically close from the uniform distribution and thus from the public key distribution E_A given by **SILBE.KG**. Therefore, any adversaries capable of breaking SILBE in the OW-qPCA-U scenario are also inherently capable of breaking a fresh instance of SILBE in a OW-qPCA scenario. This leads us to the following proposition.

Proposition 4.3.1

In the ROM,

$$\text{SILBE is OW-qPCA secure} \iff \text{SILBE is OW-qPCA-U secure}$$

Therefore, under the assumption that the supersingular isogeny problem with masked torsion point information is hard over random curves, we have that SILBE is a OW-qPCA-U secure UPKE. To make of SILBE an IND-CCA-U UPKE, we use the transformation in [AW23, section 4]. It transforms a OW-qCPA-U UPKE into an IND-CCA-U UPKE in the ROM.[2] To do so, we need to show that SILBE is λ-spread [AW23, definition 7] but this is a direct consequence of proposition 4.2.1 and of the fact that $3^\beta \gg 2^\lambda$, as we will now show.

4.3.3 Parameters & Efficiency

Finding "SILBE-friendly" primes

As we previously explained in **SILBE.Setup**, our public parameters and especially the cross relation between β and N forces N to have many prime factors. To find good N and β, we do as follows:

- If $N \leqslant 3^\beta \sqrt{p}\log(p)^2 \simeq 3^{3\beta/2} N^{1/2}\big(\log(N) + \beta\log(3)\big)$, we increase the size of N.

[2] Using their security definition, we indeed have that SILBE is an OW-CR-CPA.

- If $N_t \geqslant 3^{\beta/2}$ and $n - t < \lambda$, we increase the size of β.

Once we have found N and β, we find a good cofactor such that $p = 3^\beta N f + 1$ is prime. Using this method, we found the following parameters:

- For $\lambda = 128$:
$$\beta = 2043 \qquad N = 5 \times 7 \times 11 \times \cdots \times 6863 \qquad f = 1298$$
Here, $N = \prod_{i=1}^{n} p_i$ with $n = 881$ and $p = BNf + 1$ is 13013 bit long.
- For $\lambda = 192$:
$$\beta = 3229 \qquad N = 5 \times 7 \times 11 \times \cdots \times 10789 \qquad f = 1790$$
$n = 1312$ and p is 20538 bit long.
- For $\lambda = 256$:
$$\beta = 4461 \qquad N = 5 \times 7 \times 11 \times \cdots \times 14879 \qquad f = 16706$$
$n = 1741$ and p is 28346 bit long.

We see that in SILBE, we need N to have slightly less than 7λ distinct prime divisors.

Efficiency of SILBE

The main issue with SILBE is its efficiency. This essentially comes from the size of the parameters, together with performing Kani in dimension 4 with relatively large primes. For example, the number of field operations needed to perform the **HDKernelToIsogeny** in **SILBE.Dec** is in the order of $7^5\lambda^5\log(\lambda)^4$, which is, for $\lambda = 128$, around 2^{60}. Nevertheless, we can improve the efficiency of **SILBE.Usk** and **SILBE.KG** as follows:

- We could adapt the **RigorousDoublePath** [DLRW23, Algorithm 12] and replace Kani's Lemma by the **KLPT** for key generation and update mechanism. This would nevertheless require a change of prime p, as we would need for p to be of the form $p = 3^\beta F + 1$ such that $N|p^2 - 1$ with $N \geqslant p^{3/2}$. This is very similar to the primes used in SQISign [FKL+20]. Finding such primes would be difficult, which is the reason we choose to present **SILBE.Upk** using HD isogenies.
- We could also speed up the key generation by reusing **KaniDoublePath** to directly construct an isogeny $\phi_A : E_{1728} \to E_A$, this would nevertheless require additional assumption to ensure that the distribution is computationally indistinguishable from uniform.

Additionally, due to the size of p, we see that we can shorten the length of our path such that our distribution is not $O(p^{-1/2})$ -statistically close from uniform, but just $O(2^{-\lambda})$, which would be sufficient. Finally, to finish this chapter, we would like to highlight the fact that our update mechanism could be slightly changed to not require a hash function. This comes from the fact that our update mechanism is very similar to the CGL hash function [CGL06]. We can thus adapt [CGL06, section 5] and get that the problem of finding μ such that $\textbf{SILBE.Upk}(E, \mu) = E'$ reduces to isogeny walk problem and thus that our key update mechanism is one-way. Nevertheless, we would require some modifications of the public key as we would have to add $V_t \in E_A[3^\beta]$ such that $\langle V_t \rangle = \rho_t(E_{t-1}[3^\beta])$ to ensure that the update long isogeny is not backtracking. To keep the same security level, we would also need to compute a slightly longer isogeny.

Future directions

With the completion of chapters 3 and 4, the presentation of SQIPrime and SILBE marks the conclusion of this thesis. These two mechanisms, being distinct in nature, lead to different avenues for further exploration.
For SQIPrime, the logical next step involves a practical implementation of the scheme. Implementing SQIPrime would not only validate its theoretical underpinnings but also provide a platform to assess its efficiency. Additionally, an aspect deserving further scrutiny is the thorough exploration and refinements of assumption 3.2.1.
In the case of SILBE, an intriguing avenue for research lies in investigating whether its underlying principles can be extended to other cryptographic protocols vulnerable to generalized lollipop attacks, such as FESTA. Exploring this application could potentially enhance SILBE's efficiency.
On a more general note, a pivotal question for exploration is the refinement of **HDKernelToIsogeny**. While its results over a prime ℓ currently align with Vélu's formulas, there certainly exists opportunity for improvement. An avenue to explore is the construction of a higher-dimensional analog akin to $\sqrt{\text{élu}}$ [BFLS20]. This exploration could shed light on novel possibilities to use HD-isogenies in Isogeny Based Cryptography.

Acknowledgement

I would like to give the biggest gratitude possible (by assuming that this form an inductive poset and applying Zorn lemma) to Dr. Tako Boris Fouotsa whose unwavering support, swift problem-solving, and exceptional patience made him the invaluable oracle I could have ever wished for throughout this thesis. My sincere thanks to Prof. Serge Vaudenay for granting me the opportunity to delve into this subject and for managing all the administrative aspects of this thesis. I also appreciate the insights of Prof. Jacques Duparc, whose advices on LaTeX significantly contributed to the appearance of this thesis.

Bibliography

[AAM18] Gora Adj, Omran Ahmadi, and Alfred Menezes, *On isogeny graphs of supersingular elliptic curves over finite fields*, Cryptology ePrint Archive, Paper 2018/132, 2018, https://eprint.iacr.org/2018/132.

[AHLP22] Calvin Abou Haidar, Benoit Libert, and Alain Passelègue, *Updatable public key encryption from dcr: Efficient constructions with stronger security*, Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 11–22.

[AW23] Kyoichi Asano and Yohei Watanabe, *Updatable public key encryption with strong cca security: Security analysis and efficient generic construction*, Cryptology ePrint Archive, Paper 2023/976, 2023, https://eprint.iacr.org/2023/976.

[BF23] Andrea Basso and Tako Boris Fouotsa, *New sidh countermeasures for a more efficient key exchange*, Cryptology ePrint Archive, Paper 2023/791, 2023, https://eprint.iacr.org/2023/791.

[BFLS20] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith, *Faster computation of isogenies of large prime degree*, Cryptology ePrint Archive, Paper 2020/341, 2020, https://eprint.iacr.org/2020/341.

[Bie53] Jules Bienaymé, *Remarques sur les différences qui distinguent l'interpolation de m. cauchy de la méthode des moindres carrés, et qui assurent la supériorité de cette méthode*, Journal de Mathématiques Pures et Appliquées **18** (1853), 299–308.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren, *Csi-fish: Efficient isogeny based signatures through class group computations*, Cryptology ePrint Archive, Paper 2019/498, 2019, https://eprint.iacr.org/2019/498.

[BLMR15] Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan, *Key homomorphic prfs and their applications*, Cryptology ePrint Archive, Paper 2015/220, 2015, https://eprint.iacr.org/2015/220.

[BMP23] Andrea Basso, Luciano Maino, and Giacomo Pope, *Festa: Fast encryption from supersingular torsion attacks*, Cryptology ePrint Archive, Paper 2023/660, 2023, https://eprint.iacr.org/2023/660.

[CD23] Wouter Castryck and Thomas Decru, *An efficient key recovery attack on sidh*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 423–447.

[CGL06] Denis Charles, Eyal Goren, and Kristin Lauter, *Cryptographic hash functions from expander graphs*, Cryptology ePrint Archive, Paper 2006/021, 2006, https://eprint.iacr.org/2006/021.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes, *Csidh: An efficient post-quantum commutative group action*, Cryptology ePrint Archive, Paper 2018/383, 2018, https://eprint.iacr.org/2018/383.

[Cor07] Giuseppe Cornacchia, *Su di un metodo per la risoluzione in numeri interi dell'equazione...*, 1907.

[Cou06] Jean-Marc Couveignes, *Hard homogeneous spaces*, Cryptology ePrint Archive, Paper 2006/291, 2006, https://eprint.iacr.org/2006/291.

[CV23] Wouter Castryck and Frederik Vercauteren, *A polynomial-time attack on instances of m-sidh and festa*, Cryptology ePrint Archive, Paper 2023/1433, 2023, https://eprint.iacr.org/2023/1433.

[Deu41] Max Deuring, *Die typen der multiplikatorenringe elliptischer funktionenkörper*, Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 14, Springer Berlin/Heidelberg, 1941, pp. 197–272.

[DF17] Luca De Feo, *Mathematics of isogeny based cryptography*, arXiv preprint arXiv:1711.04062 (2017).

[DFHPS16] Luca De Feo, Cyril Hugounenq, Jérôme Plût, and Éric Schost, *Explicit isogenies in quadratic time in any characteristic*, LMS Journal of Computation and Mathematics **19** (2016), no. A, 267–282.

[DKW22] Yevgeniy Dodis, Harish Karthikeyan, and Daniel Wichs, *Updatable public key encryption in the standard model*, Cryptology ePrint Archive, Paper 2022/068, 2022, https://eprint.iacr.org/2022/068.

[DLRW23] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski, *Sqisignhd: New dimensions in cryptography*, Cryptology ePrint Archive, Paper 2023/436, 2023, https://eprint.iacr.org/2023/436.

[DMPR23] Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert, *An algorithmic approach to $(2,2)$-isogenies in the theta model and applications to isogeny-based cryptography*, Cryptology ePrint Archive, Paper 2023/1747, 2023, https://eprint.iacr.org/2023/1747.

[dQKL+20] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange, *Improved torsion-point attacks on sidh variants*, Cryptology ePrint Archive, Paper 2020/633, 2020, https://eprint.iacr.org/2020/633.

[EJKM20] Edward Eaton, David Jao, Chelsea Komlo, and Youcef Mokrani, *Towards post-quantum updatable public-key encryption via supersingular isogenies*, Cryptology ePrint Archive, Paper 2020/1593, 2020, https://eprint.iacr.org/2020/1593.

[FdSGF+19] Luca De Feo, Cyprien Delpech de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski, *SÉta: Supersingular encryption from torsion attacks*, Cryptology ePrint Archive, Paper 2019/1291, 2019, https://eprint.iacr.org/2019/1291.

[FJP11] Luca De Feo, David Jao, and Jérôme Plût, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Cryptology ePrint Archive, Paper 2011/506, 2011, https://eprint.iacr.org/2011/506.

[FKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *Sqisign: compact post-quantum signatures from quaternions and isogenies*, Cryptology ePrint Archive, Paper 2020/1240, 2020, https://eprint.iacr.org/2020/1240.

[FLLW22] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski, *New algorithms for the deuring correspondence: Towards practical and secure sqisign signatures*, Cryptology ePrint Archive, Paper 2022/234, 2022, https://eprint.iacr.org/2022/234.

[FMP23] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit, *M-sidh and md-sidh: countering sidh attacks by masking information*, Cryptology ePrint Archive, Paper 2023/013, 2023, https://eprint.iacr.org/2023/013.

[Fou22] Tako Boris Fouotsa, *Sidh with masked torsion point images*, Cryptology ePrint Archive, Paper 2022/1054, 2022, https://eprint.iacr.org/2022/1054.

[FP21] Tako Boris Fouotsa and Christophe Petit, *A new adaptive attack on sidh*, Cryptology ePrint Archive, Paper 2021/1322, 2021, https://eprint.iacr.org/2021/1322.

[FS86] Amos Fiat and Adi Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Conference on the theory and application of cryptographic techniques, Springer, 1986, pp. 186–194.

[Gol09] Oded Goldreich, *Foundations of cryptography: volume 2, basic applications*, Cambridge university press, 2009.

[GPS16] Steven D. Galbraith, Christophe Petit, and Javier Silva, *Identification protocols and signature schemes based on supersingular isogeny problems*, Cryptology ePrint Archive, Paper 2016/1154, 2016, https://eprint.iacr.org/2016/1154.

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti, *On the security of supersingular isogeny cryptosystems*, Cryptology ePrint Archive, Paper 2016/859, 2016, https://eprint.iacr.org/2016/859.

[HM21] Darrel Hankerson and Alfred Menezes, *Elliptic curve cryptography*, Encyclopedia of Cryptography, Security and Privacy, Springer, 2021, pp. 1–2.

[JZC+17] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma, *Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited*, Cryptology ePrint Archive, Paper 2017/1096, 2017, https://eprint.iacr.org/2017/1096.

[Kan97] Ernst Kani, *The number of curves of genus two with elliptic differentials.*

[Ler22] Antonin Leroux, *Quaternion algebra and isogeny-based cryptography*, Ph.D. thesis, Ecole doctorale de l?Institut Polytechnique de Paris, 2022.

[Ler23] Antonin Leroux, *Verifiable random function from the deuring correspondence and higher dimensional isogenies*, Cryptology ePrint Archive, Paper 2023/1251, 2023, https://eprint.iacr.org/2023/1251.

[LLC+23] Kaizhan Lin, Jianming Lin, Shiping Cai, Weize Wang, and Chang-An Zhao, *Public-key compression in m-sidh*, Cryptology ePrint Archive, Paper 2023/136, 2023, https://eprint.iacr.org/2023/136.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász, *Factoring polynomials with rational coefficients*, Mathematische annalen **261** (1982), no. ARTICLE, 515–534.

[LR22] Antonin Leroux and Maxime Roméas, *Updatable encryption from group actions*, Cryptology ePrint Archive (2022).

[Mil86] James S Milne, *Abelian varieties*, Arithmetic geometry (1986), 103–150.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski, *A direct key recovery attack on sidh*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 448–471.

[MMRV09] J Miret, Ramiro Moreno, Anna Rio, and Magda Valls, *Computing the -power torsion of an elliptic curve over a finite field*, Mathematics of computation **78** (2009), no. 267, 1767–1786.

[Mor23] Tomoki Moriya, *Is-cube: An isogeny-based compact kem using a boxed sidh diagram*, Cryptology ePrint Archive, Paper 2023/1506, 2023, https://eprint.iacr.org/2023/1506.

[Mum66] David Bryant Mumford, *On the equations defining abelian varieties. i*, Inventiones mathematicae (1966).

[MVO91] Alfred Menezes, Scott Vanstone, and Tatsuaki Okamoto, *Reducing elliptic curve logarithms to logarithms in a finite field*, Proceedings of the twenty-third annual ACM symposium on Theory of computing, 1991, pp. 80–89.

[NO23] Kohei Nakagawa and Hiroshi Onuki, *Qfesta: Efficient algorithms and parameters for festa using quaternion algebras*, Cryptology ePrint Archive, Paper 2023/1468, 2023, https://eprint.iacr.org/2023/1468.

[OM21] Hiroshi Onuki and Tomoki Moriya, *Radical isogenies on montgomery curves*, Cryptology ePrint Archive, Paper 2021/699, 2021, https://eprint.iacr.org/2021/699.

[Pet17] Christophe Petit, *Faster algorithms for isogeny problems using torsion point images*, 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.

[Piz90] Arnold K Pizer, *Ramanujan graphs and hecke operators*, Bulletin of the American Mathematical Society **23** (1990), no. 1, 127–137.

[PS18] Christophe Petit and Spike Smith, *An improvement to the quaternion analogue of the l-isogeny path problem*, Proceedings of MATHCRYPT (2018).

[PT18] Paul Pollack and Enrique Treviño, *Finding the four squares in lagrange's theorem.*, Integers **18** (2018), no. A15, 7–17.

[PW23] Aurel Page and Benjamin Wesolowski, *The supersingular endomorphism ring and one endomorphism problems are equivalent*, Cryptology ePrint Archive, Paper 2023/1399, 2023, https://eprint.iacr.org/2023/1399.

[Rob10] Damien Robert, *Fonctions thêta et applications à la cryptographie*, Ph.D. thesis, Université Henri Poincaré-Nancy I, 2010.

[Rob22a] Damien Robert, *Breaking sidh in polynomial time*, Cryptology ePrint Archive, Paper 2022/1038, 2022, https://eprint.iacr.org/2022/1038.

[Rob22b] Damien Robert, *Evaluating isogenies in polylogarithmic time.*

[RS86] Michael O Rabin and Jeffery O Shallit, *Randomized algorithms in number theory*, Communications on Pure and Applied Mathematics **39** (1986), no. S1, S239–S256.

[RS06] Alexander Rostovtsev and Anton Stolbunov, *Public-key cryptosystem based on isogenies*, Cryptology ePrint Archive, Paper 2006/145, 2006, https://eprint.iacr.org/2006/145.

[Sam13] Pierre Samuel, *Algebraic theory of numbers: Translated from the french by allan j. silberger*, Courier Corporation, 2013.

[Sha16] Igor R Shafarevich, *Basic algebraic geometry 1*, Springer, 2016.

[Sho94] P.W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134.

[Sil94] Joseph H Silverman, *Advanced topics in the arithmetic of elliptic curves*, vol. 151, Springer Science & Business Media, 1994.

[Sil09] ———, *The arithmetic of elliptic curves*, vol. 106, Springer, 2009.

[Sut15] Andrew Sutherland, *18.783 elliptic curves lecture*, 2015.

[Vél71] Jacques Vélu, *Isogénies entre courbes elliptiques*, Comptes-Rendus de l'Académie des Sciences **273** (1971), 238–241.

[Voi21] John Voight, *Quaternion algebras*, Springer Nature, 2021.

[Was08] Lawrence C Washington, *Elliptic curves: number theory and cryptography*, CRC press, 2008.

[Wes22] Benjamin Wesolowski, *The supersingular isogeny path and endomorphism ring problems are equivalent*, 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2022, pp. 1100–1111.

[YAJ+17] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev, *A post-quantum digital signature scheme based on supersingular isogenies*, Cryptology ePrint Archive, Paper 2017/186, 2017, https://eprint.iacr.org/2017/186.

[ZJP+17] Gustavo H. M. Zanon, Marcos A. Simplicio Jr, Geovandro C. C. F. Pereira, Javad Doliskani, and Paulo S. L. M. Barreto, *Faster key compression for isogeny-based cryptosystems*, Cryptology ePrint Archive, Paper 2017/1143, 2017, https://eprint.iacr.org/2017/1143.