A New Isogeny Representation and Applications to Cryptography

Antonin Leroux

[1] DGA
[2] LIX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris
[3] INRIA
antonin.leroux@polytechnique.org

Abstract. This paper focuses on isogeny representations, defined as witnesses of membership to the language of isogenous supersingular curves (the set of triples D, E_1, E_2 with a cyclic isogeny of degree D between E_1 and E_2). This language and its proofs of membership are known to have several fundamental cryptographic applications such as the construction of digital signatures and validation of encryption keys.
The first part of our article is dedicated to formalizing known results about isogenies to the framework of languages and proofs, culminating in a proof that the language of isogenous supersingular curves is in NP with the isogeny representation derived naturally from the Deuring Correspondence.
Our main contribution is the design of the suborder representation, a new isogeny representation targetted at the case of (big) prime degree. The core of our new method is the revelation of endomorphisms of smooth norm inside a well-chosen suborder of the codomain's endomorphism ring. These new membership witnesses appear to be opening interesting prospects for isogeny-based cryptography under the hardness of a new computational problem: the SubOrder to Ideal Problem (SOIP). As an application, we introduce pSIDH, a new NIKE based on the suborder representation.
In the process, we also develop several heuristic algorithmic tools to solve norm equations inside a new family of quaternion orders. These new algorithms may be of independent interest.

1 Introduction

Isogeny-based cryptography has been receiving an increasing amount of interest over the last few years due to its presumed resistance to quantum computers. As the variety of primitive achievable from isogenies is expanding, new problems are arising. The problem of proving the knowledge of an isogeny between two elliptic curves is appearing in various contexts such as SIDH [JDF11] key validation [GPST16], digital signatures [YAJ+17,DFG19,BKV19,JS14], VDFs [DFMPS19,CSHT21], delay encryption [BDF21] and oblivious PRF [BKW20].

Intuitively, proving a statement requires an efficient way to represent and manipulate the objects involved in that statement. In the case of isogenies, the standard representation is obtained from the Vélu formulas [Vél71] that give a way to compute and evaluate an isogeny from its kernel. The best generic algorithm to compute these formulas requires $\tilde{O}(\sqrt{D'})$ operations over the field of definition of the isogeny's kernel where D' is the biggest factor of the degree (see [BFLS20]). Thus, the computation is only efficient when the degree is smooth and the kernel points are defined over a small field extension. In full generality, this only happens when the degree is powersmooth but there are ways to make it work for smooth degrees as well. All the schemes we mentioned so far are subject to these computational limitations and use smooth degrees. However, the recent trend of works studying the Deuring Correspondence and its applications to isogeny-based cryptography has provided us the means to represent and manipulate efficiently isogenies of arbitrary degrees.

Everything started with the so-called KLPT algorithm from Kohel et al. [KLPT14] to solve the quaternion analog of the isogeny path problem. In [EHL+18], Eisentrager et al. heuristically showed that quaternion ideals can be used as an *efficient representation* of isogenies, with the *"effiency"* stemming from KLPT and other heuristic polynomial-time algorithms. Wesolowski presented provable variants of these algorithms in his recent article [Wes22].

The tools of the Deuring Correspondence and the efficient algorithms from [KLPT14,EHL+18] were originally introduced for cryptanalytic purposes and have only recently been used constructively. The main building blocks of the signature scheme from Galbraith, Petit and Silva [GPS17] and the later generalization of SQISign [DFKL+20] are variants of the KLPT algorithm from Kohel et al. The key generation of the encryption scheme SETA [DFFdSG+21] is also based on the same techniques. The first complete implementation of all these algorithmic blocks was another contribution of the authors of SQISign. Additionally, this protocol is the first example of a scheme that is explicitly making use of isogenies of big prime degree. In [DFKL+20, Section 8.3], the authors argue that using a secret key of prime degree provide better efficiency for the same level of security. While providing us with powerful tools, the representation of isogenies as quaternion ideals also seem to have some limitations when considering cryptographic applications as we argue in Section 3.3. The motivation of our paper is to fill that gap with a new way to represent isogenies.

A first small contribution of this work is to translate some of the notations and results from the isogeny literature into the formalism of languages and proofs. The results from [EHL+18,Wes22] proves that the language $\mathcal{L}_{\mathsf{isog}}$ of isogenous curves (see Definition 1) is in NP. We define *isogeny representations* as membership witnesses for $\mathcal{L}_{\mathsf{isog}}$. Our hope is to provide a precise terminology to state formal results about isogeny-based cryptography and proofs of isogeny knowledge.

Our main contribution is a new generic isogeny representation that we call a *suborder witness* or *suborder representation*. This representation is constituted of several endomorphisms of the isogeny's codomain. We present polynomial-time algorithms to compute and verify suborder witnesses when the degree D is prime. The case of composite D is more complicated and does not seem to be more interesting for cryptography, we treat it in appendix for completeness. The *suborder representation* is not equivalent to the *ideal representation* under the hardness of a new computational problem: the Suborder to Ideal Problem (SOIP), or its equivalent reformulation: the Suborder to Endomorphism Ring Problem (SOERP). The assumed hardness of the SOERP contradicts the common belief that the knowledge of a suborder of rank 4 is enough to derive the full endomorphism ring of a supersingular curve. We include in Section 4.5, a discussion about the hardness of those new problems.

Our new efficient algorithms requires to solve norm equations inside a new family of quaternion orders and we develop the necessary tools for that task. This contribution may be of independent interest as solving norm equations inside different types of order have proven to be useful in various situations such as [DFKL+20,DFFdSG+21].

Finally, we illustrate the cryptographic interest of our new isogeny representation by building pSIDH, a NIKE based on a generalization of SIDH to the prime degree setting. The key recovery problem is the SOIP and the key exchange is secure under the hardness of a decisional variant of the SOIP. We introduce this primitive not as a potential replacement for SIDH (efficiency will likely be too poor for that) but rather as a first step toward more involved applications as we discuss in Section 6.2.

The rest of this paper is organized as follows: Section 2 is dedicated to the background materials. In Section 3, we give the definition for $\mathcal{L}_{\mathsf{isog}}$, the language of isogenous curves, and show that it is in NP using the *ideal representation* of isogenies. In Section 4, we introduce a new isogeny representation: *suborder witnesses*. We provide some algorithms to compute and verify them, and analyze how they differ from *ideal witnesses*. The algorithmic gaps left in Section 4 are filled in Section 5 where we introduce new algorithms to solve norm equations inside a new family of quaternion orders. Finally, we discuss cryptographic applications of the suborder representation in Section 6 where we introduce a new isogeny-based NIKE scheme and discuss prospects for other constructions.

2 Background material

The set of prime numbers is denoted $\mathbb{P}$.

We call *negligible* a function $f : \mathbb{Z}_{>0} \to \mathbb{R}_{>0}$ if it is asymptotically dominated by $O(x^{-n})$ for all $n > 0$. In the analysis of a probabilistic algorithm, we say that an event happens with *overwhelming probability* if its probability of failure is a negligible function of the length of the input.

2.1 Languages and proofs

A relation is a map $R : L \times W \to \{0, 1\}$. Any relation implicitly defines a language as $\mathcal{L}_R = \{x \in L \mid \exists\ w \in W, R(x, w) = 1\}$. For $x \in \mathcal{L}_R$, we call a *membership witness* (sometimes simply *witness*) or *proof*, any $w \in W$ such that $R(x, w) = 1$.

Conversely, for any language $\mathcal{L} \subset L$, we call a verification algorithm any function $R_{\mathcal{L}} : L \times W \to \{0,1\}$ such that there exists a witness $w \in W$ with $R_{\mathcal{L}}(x,w) = 1$ if and only if $x \in \mathcal{L}$.

The class NP contains all languages that can be verified in polynomial time. More precisely, a language $\mathcal{L}$ is in NP if there exists a polynomial-time verification algorithm $R_{\mathcal{L}}$ and there exists a witness w with $|w| = \mathsf{poly}(|x|)$ for any $x \in \mathcal{L}$.

Note that the computation of the witness need not be efficient, only the verification. This is the main difference between P and NP.

2.2 Elliptic curves, quaternion algebras and the Deuring correspondence

Supersingular elliptic curves and isogenies. An *isogeny* $\varphi : E_1 \to E_2$ is a non-constant morphism sending the identity of E_1 to that of E_2. The degree of an isogeny is its degree as a rational map (see [HS09] for more details). When the degree $\deg(\varphi) = d$ is coprime to p, the isogeny is necessarily *separable* and $d = \# \ker \varphi$. An isogeny is said to be cyclic when its kernel is a cyclic group. The Vélu formulas [Vél71] can be used to compute any cyclic isogeny from its kernel. For any $\varphi : E_1 \to E_2$, there exists a unique dual isogeny $\hat{\varphi} : E_2 \to E_1$, satisfying $\varphi \circ \hat{\varphi} = [\deg(\varphi)]$.

Endomorphism ring. An isogeny from a curve E to itself is an *endomorphism*. The set $\mathrm{End}(E)$ of all endomorphisms of E forms a ring under addition and composition. For elliptic curves defined over a finite field $\mathbb{F}_q$, $\mathrm{End}(E)$ is isomorphic either to an order of a quadratic imaginary field or a maximal order in a quaternion algebra. In the first case, the curve is said to be *ordinary* and otherwise *supersingular*. We focus on the supersingular case in this article. Every supersingular elliptic curve defined over a field of characteristic p admits an isomorphic model over $\mathbb{F}_{p^2}$. It implies that there only a finite number of isomorphism class of supersingular elliptic curves. The Frobenius over $\mathbb{F}_p$ is the only inseparable isogeny between supersingular curves and it has degree p. We write $\pi : E \to E^p$. For any supersingular curve E, the property $\mathrm{End}(E) \cong \mathrm{End}(E^p)$ is satisfied but we have $E \cong E^p$ if and only if E has an isomorphic model over $\mathbb{F}_p$.

Quaternion algebras. For $a, b \in \mathbb{Q}^\star$ we denote by $H(a,b) = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + k\mathbb{Q}$ the quaternion algebra over $\mathbb{Q}$ with basis $1, i, j, k$ such that $i^2 = a$, $j^2 = b$ and $k = ij = -ji$. Every quaternion algebra has a canonical involution that sends an element $\alpha = a_1 + a_2 i + a_3 j + a_4 k$ to its conjugate $\overline{\alpha} = a_1 - a_2 i - a_3 j - a_4 k$. We define the *reduced trace* and the *reduced norm* by $tr(\alpha) = \alpha + \overline{\alpha}$ and $n(\alpha) = \alpha\overline{\alpha}$.

Orders and ideals. A *fractional ideal* I of a quaternion algebra $\mathcal{B}$ is a $\mathbb{Z}$-lattice of rank four contained in $\mathcal{B}$. We denote by $n(I)$ the *norm* of I, defined as the $\mathbb{Z}$-module generated by the reduced norms of the elements of I.

An order $\mathcal{O}$ is a subring of $\mathcal{B}$ that is also a fractional ideal. Elements of an order $\mathcal{O}$ have reduced norm and trace in $\mathbb{Z}$. An order is called *maximal* when it is not contained in any other larger order. A suborder $\mathfrak{O}$ of $\mathcal{O}$ is an order of rank 4 contained in $\mathcal{O}$.

The left order of a fractional ideal is defined as $\mathcal{O}_L(I) = \{\alpha \in \mathcal{B}_{p,\infty} \mid \alpha I \subset I\}$ and similarly for the right order $\mathcal{O}_R(I)$. A fractional ideal is *integral* if it is contained in its left order, or equivalently in its right order. An integral ideal is *primitive* if it is not the scalar multiple of another integral ideal. We refer to integral primitive ideals hereafter as ideals.

The product IJ of ideals I and J satisfying $\mathcal{O}_R(I) = \mathcal{O}_L(J)$ is the ideal generated by the products of pairs in $I \times J$. It follows that IJ is also an (integral) ideal and $\mathcal{O}_L(IJ) = \mathcal{O}_L(I)$ and $\mathcal{O}_R(IJ) = \mathcal{O}_R(J)$. The ideal norm is multiplicative with respect to ideal products. An ideal I is invertible if there exists another ideal I^{-1} verifying $II^{-1} = \mathcal{O}_L(I) = \mathcal{O}_R(I^{-1})$ and $I^{-1}I = \mathcal{O}_R(I) = \mathcal{O}_L(I^{-1})$. The conjugate of an ideal $\overline{I}$ is the set of conjugates of elements of I, which is an ideal satisfying $I\overline{I} = n(I)\mathcal{O}_L(I)$ and $\overline{I}I = n(I)\mathcal{O}_R(I)$.

We define an equivalence on orders by conjugacy and on left $\mathcal{O}$-ideals by right scalar multiplication. Two orders $\mathcal{O}_1$ and $\mathcal{O}_2$ are equivalent if there is an element $\beta \in \mathcal{B}^\star$ such that $\beta\mathcal{O}_1 = \mathcal{O}_2\beta$. Two left $\mathcal{O}$-ideals I and J are equivalent if there exists $\beta \in \mathcal{B}^\star$, such that $I = J\beta$. If the latter holds, then it follows that $\mathcal{O}_R(I)$ and $\mathcal{O}_R(J)$ are equivalent since $\beta\mathcal{O}_R(I) = \mathcal{O}_R(J)\beta$. For a given $\mathcal{O}$, this defines equivalences classes of left $\mathcal{O}$-ideals, and we denote the set of such classes by $\mathrm{Cl}(\mathcal{O})$.

Similarly to quadratic orders, quaternion admit what we call a *Gorenstein decomposition*. Any quaternion order $\mathcal{O}$ can be expressed as $\mathbb{Z} + f\mathcal{O}_0$, where f is the *Brandt Invariant* or *Gorenstein Conductor* and $\mathcal{O}_0$ is the *Gorenstein Closure*. As the name indicates, the Gorenstein Closure is a Gorenstein order (i.e orders whose Brandt invariant is 1). A *Bass* order, is an order for which all suborders are gorenstein. Equivalent definitions and further properties of Gorenstein and Bass orders can be found in [Voi18]. Eichler orders are Bass order that can be written as the intersection of two maximal orders. A study of Eichler orders and their interpretation under the Deuring Correspondence can be found in [DFKL+20, Section 4].

The Deuring correspondence is an equivalence of categories between isogenies of supersingular elliptic curves and the left ideals over maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$, the unique quaternion algebra ramified at p and ∞, inducing a bijection between conjugacy classes of supersingular j-invariants and maximal orders (up to equivalence) [Koh96]. Moreover, this bijection is explicitly constructed as $E \to \mathrm{End}(E)$. Hence, given a supersingular curve E_0 with endomorphism ring $\mathcal{O}_0$, the pair (E_1, φ), where E_1 is another supersingular elliptic curve and $\varphi : E_0 \to E_1$ is an isogeny, is sent to a left integral $\mathcal{O}_0$-ideal. The right order of this ideal is isomorphic to $\mathrm{End}(E_1)$. One way of realizing this correspondence is obtained through the kernel ideals defined in [Wat69]. Given an integral left-$\mathcal{O}_0$-ideal I, we define the kernel of I as the subgroup

$$E_0[I] = \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0 \text{ for all } \alpha \in I\}.$$

To I, we associate the isogeny

$$\varphi_I : E_0 \to E_0/E_0[I].$$

Conversely, given an isogeny φ, the corresponding *kernel ideal* is

$$I_\varphi = \{\alpha \in \mathcal{O}_0 \ : \ \alpha(P) = 0 \text{ for all } P \in \ker(\varphi)\}.$$

In Table 1, we recall the main features of the Deuring correspondence.

Supersingular j-invariants over $\mathbb{F}_{p^2}$ $j(E)$ (up to Galois conjugacy)	Maximal orders in $B_{p,\infty}$ $\mathcal{O} \cong \mathrm{End}(E)$ (up to isomorpshim)
(E_1, φ) with $\varphi : E \to E_1$	I_φ integral left $\mathcal{O}$-ideal and right $\mathcal{O}_1$-ideal
$\theta \in \mathrm{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$n(I_\varphi)$
$\hat{\varphi}$	$\overline{I_\varphi}$
$\varphi : E \to E_1, \psi : E \to E_1$	Equivalent Ideals $I_\varphi \sim I_\psi$
Supersingular j-invariants over $\mathbb{F}_{p^2}$	$\mathrm{Cl}(\mathcal{O})$
$\tau \circ \rho : E \to E_1 \to E_2$	$I_{\tau\circ\rho} = I_\rho \cdot I_\tau$

Table 1. The Deuring correspondence, a summary from [DFKL$^+$20].

3 The language of isogenous supersingular curves is in NP

Let us fix a prime p. Note that most of what follows is not targetted at any specific prime p, even though an efficient instantiation might require a careful choice of p.

We will study $\mathcal{L}_{\mathsf{isog}}$, the language of isogenous supersingular curves in characteristic p. The purpose of this section is to show in Theorem 1 that $\mathcal{L}_{\mathsf{isog}} \in \mathsf{NP}$.

We write $\mathcal{S}_p$ as the set isomorphism classes of supersingular elliptic curves in characteristic p, and Isog_D the set (up to pre and post-composition with isomorphims) of cyclic D-isogenies between curves of $\mathcal{S}_p$.

Definition 1. *The language of isogenous supersingular curves is*

$$\mathcal{L}_{\mathsf{isog}} = \{(D, E_1, E_2) \in \mathbb{N} \times \mathcal{S}_p^2 \ \mid \ \exists \ \varphi : E_1 \to E_2 \in \mathsf{Isog}_D\}.$$

We call an ***isogeny representation*** *any membership witness to $x \in \mathcal{L}_{\mathsf{isog}}$.*

In the rest of this paper, we implicitly assume that any isogeny representation for D, E_1, E_2 is associated to a concrete isogeny $\varphi : E_1 \to E_2$ of degree D. We will write it φ_I for ideal witnesses from Section 3.2, φ_π for the suborder witnesses introduced in Section 4.2 and when it is clear from the context, we might just write φ.

In the rest of this section, we fix an element $x = (D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$ and show that we can build an isogeny representation that has polynomial size in $|x|$ from the Deuring Correspondence. We recall the relevant results and algorithms from the literature in Section 3.1

3.1 Polynomial-time algorithms of the Deuring Correspondence

We give below a list of algorithms taken from the literature. Throughout this paper, we are going to use the provable version of these algorithms, most of which were introduced by Wesolowski in [Wes22]. For a concrete instantiation of any of them, one will rather want to use the efficient heuristic version (see [DFKL+20] for instance). The KLPT algorithms depend on some special extremal order $\mathcal{O}_0$ that we consider as a fixed parameter.

- ConnectingIdeal: takes two maximal orders $\mathcal{O}_1, \mathcal{O}_2 \subset B_{p,\infty}$ and outputs an ideal I with $\mathcal{O}_L(I) = \mathcal{O}_1$ and $\mathcal{O}_R(I) = \mathcal{O}_2$.
- $\mathsf{KLPT}_{\ell^\bullet}$: takes an ideal I and output $J \sim I$ of norm ℓ^e.
- $\mathsf{KLPT}_{\mathsf{PS}}$: takes a left $\mathcal{O}_0$-ideal I and output $J \sim I$ of powersmooth norm.
- $\mathsf{IdealToIsogeny}_T$: takes a left $\mathcal{O}$-ideal I of norm T and compute φ_I.
- $\mathsf{IsogenyToIdeal}_T$: takes an isogeny $\varphi : E \to E'$ of degree T, a maximal order $\mathcal{O} \cong \mathrm{End}(E)$ and compute I_φ.

We reformulate below in Proposition 1 to Proposition 5, some of the results proven in [Wes22].

Proposition 1. *ConnectingIdeal terminates in $O(\mathsf{poly}(\log(p)+C))$ when the coefficients of the bases of the two maximal orders can be represented with C bits.*

Proposition 2. *Assuming GRH, $\mathsf{KLPT}_{\ell^\bullet}$ terminates in expected $O(\mathsf{poly}(\log(pD))$ where D is the norm of the input and outputs an ideal of norm e where $e = O(\mathsf{poly}(\log(p))$.*

Proposition 3. *Assuming GRH, $\mathsf{KLPT}_{\mathsf{PS}}$ terminates in expected $O(\mathsf{poly}(\log(pD))$ where D is the norm of the input and outputs an ideal of norm in $O(\mathsf{poly}(p))$ with smoothness bound in $O(\mathsf{poly}(\log(p)))$.*

Proposition 4. *For any number $T = O(\mathsf{poly}(p))$ with smoothness bound in $O(\mathsf{poly}(\log(p)))$, $\mathsf{IsogenyToIdeal}_T$ terminates in expected $O(\mathsf{poly}(\log(p)))$ and the output has size $O(\mathsf{poly}(\log(p)))$.*

Proposition 5. *For any number $T = O(\mathsf{poly}(p))$ with smoothness bound in $O(\mathsf{poly}(\log(p)))$, $\mathsf{IdealToIsogeny}_T$ terminates in expected $O(\mathsf{poly}(\log(p)+C))$ and the output has size $O(\mathsf{poly}(\log(p)))$ when the coefficients of the basis of $\mathcal{O}$ can be represented with C bits.*

3.2 Ideal witnesses: membership proofs to $\mathcal{L}_{\mathsf{isog}}$ from the Deuring Correspondence

The membership witnesses for $\mathcal{L}_{\mathsf{isog}}$ that we propose to use are the following: if φ is an isogeny of degree D between E_1 and E_2, the witness to $x = (D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$ is the corresponding ideal I_φ. Henceforth, we will call such an ideal I an *ideal withness* to $x \in \mathcal{L}_{\mathsf{isog}}$.

Lemma 1. *Any ideal of norm D admits a representation of size $O(\log(D) + \log(p))$.*

Proof. It was shown in [EHL+18] that any maximal order admits a basis whose coefficients have size $O(\log(p))$ in the basis $\langle 1, i, j, k\rangle$ of $B_{p,\infty}$. Since $D\mathcal{O} \subset I$ for any cyclic $\mathcal{O}$-ideal of norm D we see that we can choose coefficients to represent any elements of I inside the basis of $\mathcal{O}$ with coefficients of size $O(\log(D))$. Thus, there exists a representation of a basis of I in $\langle 1, i, j, k\rangle$ whose coefficients have size $O(\log(p) + \log(D))$.

When the prover is unbounded, it is clear that he can compute the compact representation of an ideal I whose corresponding isogeny is connecting E_1 and E_2. Indeed, since there is a finite number of maximal orders and ideals of a given norm inside $B_{p,\infty}$, the prover can simply enumerate through all of them until a fitting one is found.

We now present $\mathsf{VerifIdealProof}$, a verification algorithm that takes a triple $x = (D, E_1, E_2)$ and an ideal I and decides if $x \in \mathcal{L}_{\mathsf{isog}}$. The idea is to use the following procedure on ideals connecting a special order $\mathcal{O}_0$ with $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$: use KLPT to get an equivalent ideal of smooth norm and compute the corresponding isogeny with $\mathsf{IdealToIsogeny}$. Since, these isogenies have smooth norm, they can be efficiently computed and after that it is just matter of checking that their codomain is correct.

Lemma 2. *Let D be any integer in $\mathbb{N}$ coprime with p. If $\varphi : E_1 \to E_2$ has degree D, then $\mathsf{VerifIdealProof}((D, E_1, E_2), I_\varphi) = 1$.*

Conversely, for $(D, E_1, E_2) \in \mathbb{N} \times \mathcal{S}_p^2$, if there exists an ideal I such that $\mathsf{VerifIdealProof}((D, E_1, E_2), I) = 1$ *then $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$.*

Proof. Let us take $\varphi : E_1 \to E_2$ of degree D. By definition of I_φ, we have $n(I_\varphi) = D$ and $I_\varphi \subset \mathcal{O}_L(I)$ so the first check passes. Then, the codomain of the two φ_{I_i} have endomorphism ring isomorphic to $\mathcal{O}_R(I_i)$ so they might be either both E_i or both E_i^p (since $I_2 = I_1 I$, it cannot be E_1, E_2^p or E_1^p, E_2). In both cases, the final output is 1.

If there exists an ideal I such that $\mathsf{VerifIdealProof}((D, E_1, E_2), I) = 1$, then $n(I) = D$ and I is integral (this is from the first verification). Since $I = \overline{I_1} \cdot I_2/n(I_1) \sim \overline{J_1} \cdot J_2$ is an integral ideal of degree D, there exists an isogeny of degree D between E_1', E_2'. Since the final output is 1, the two curves E_1', E_2' are equal to either E_1, E_2 or E_1^p, E_2^p. Since $\varphi : E_1^p \to E_2^p$ of degree D imply the existence of $\varphi^p : E_1 \to E_2$ of degree D, in both cases we have that $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$.

Algorithm 1 $\mathsf{VerifIdealProof}(x, I)$

Require: $x \in \mathbb{N} \times \mathcal{S}_p^2$ and I an ideal of $B_{p,\infty}$.
Ensure: A bit indicating if $x \in \mathcal{L}_{\mathsf{isog}}$.

```
Parse x as D, E_1, E_2 and take ℓ a small prime.
Compute n(I) and O_L(I), O_R(I).
if n(I) ≠ D or I ⊄ O_L(I) then
    Return 0.
end if
Take a curve E_0 defined over F_p with End(E_0) ≅ O_0 and compute I_1 =
    ConnectingIdeal(O_0, O_L(I)), I_2 = I_1 · I.
for i ∈ [1, 2] do
    Compute J_i = KLPT_{ℓ•}(I_i) and φ_i : E_0 → E'_i = IdealToIsogeny_{ℓ•}(E_0, J_i).
end for
if j(E'_1), j(E'_2) ∉ {(j(E_1), j(E_2)), (j(E_1)^p, j(E_2)^p)} then
    Return 0.
end if
return 1.
```

Proposition 6. *Under GRH,* $\mathsf{VerifIdealProof}$ *terminates in expected* $O(\mathsf{poly}(\log(pD)))$.

Proof. The basis elements of left and right orders of an ideal of norm D can be written in $O(\log(pD))$ bits. Then, the results follows from Propositions 1, 2 and 5.

We are now ready to state our important result. Theorem 1 below is a consequence of Proposition 6 and Lemmas 1 and 2.

Theorem 1. *Assuming GRH,* $\mathcal{L}_{\mathsf{isog}} \in \mathsf{NP}$.

Proof. Lemma 1 ensures that the proof has polynomial size in x, Proposition 6 shows that the verification is polynomial-time under GRH and Lemma 2 shows that there exists a witness and that it passes verification if and only if $x \in \mathcal{L}_{\mathsf{isog}}$. Combining those three properties together proves the result.

3.3 Advantages and limitations of the ideal witness

In the beginning of this section, we highlight some of the powerful operations achievable with ideal witnesses. In the last paragraph, we will explain why these algorithms imply that ideal witnesses have a limited interest for some cryptographic applications.

The alternate path problem. Two curves E_1, E_2 are connected by an infinite number of isogenies. The problem of finding an ideal witness for (N, E_1, E_2) from an ideal witness for (D, E_1, E_2) with $N \neq D$, was first introduced and solved in [KLPT14]. The efficient solution KLPT presented in this article for $N = \ell^\bullet$, unlocked all the subsequent results and algorithms from [EHL+18,Wes22]. The verification process that we described in Algorithm 1 is heavily relying on KLPT to find equivalent ideals of powersmooth norms. The $\mathsf{IdealEvaluation}$ algorithm described below is also making use of that mechanism.

Isogeny Evaluation. Next, we show how to evaluate the isogeny φ_I on any point of order coprime with $n(I)$ from I. For simplicity, we assume that I is an $\mathcal{O}_0$-ideal where $\mathcal{O}_0 \cong \mathrm{End}(E_0)$ and E_0 is a curve for which evaluating endomorphisms can be done easily (the curve of j-invariant 1728 is an example of such a curve). A generic algorithm of complexity $O(\mathsf{poly}(\log(pD))$ exists but it is more complicated and we do not really need it here. An algorithm very similar to IdealEvaluation can be found in [FKM21]. The main idea is to apply KLPT and IdealToIsogeny to find an equivalent isogeny of powersmooth degree and making use of it to perform the computation.

Algorithm 2 IdealEvaluation(I, P)

Require: I an $\mathcal{O}_0$-ideal of $B_{p,\infty}$ and $P \in E_0(\overline{\mathbb{F}_p})$ of order coprime with $D = n(I)$.
Ensure: $\varphi_I(P)$.
1: Take a small prime number ℓ.
2: Compute $J = \mathsf{KLPT}_{\ell^\bullet}(I)$ and set $K = I \cdot \overline{J}$. We write $\alpha \in \mathrm{End}(E)$ for the endomorphism φ_K.
3: Compute $\alpha(P)$.
4: Compute $\varphi_J = \mathsf{IdealToIsogeny}_{\ell^\bullet}(J)$ and compute $Q = \varphi_J(\alpha(P))$.
5: Compute $\mu = n(J)^{-1} \mod n(I)$.
6: **return** $[\mu]Q$.

Proposition 7. *Under GRH,* IdealEvaluation *is correct and terminates in probabilistic $O(\mathsf{poly}(\log(p) + \log(D)))$ operations over the field of definition of P.*

Proof. We have $\varphi_K = \hat{\varphi}_J \circ \varphi_I$ and so $\mu\varphi_J(\alpha(P)) = \varphi_I(P)$. The division by μ makes sens mod $n(I)$ since the order of P is coprime with $n(I)$. The correctness of IdealEvaluation follows from the correctness of the sub-algorithms KLPT,IdealToIsogeny (see Propositions 2 and 5). Step 3 can be executed because of our assumption on E_0. If we assume that $\ell = O(1)$, termination is a consequence of Propositions 1, 2 and 5 and that the computation of $\varphi_J(P)$ can be done in $O(\mathsf{poly}(\log(p)))$ operations over the field of definition of P since $\deg\varphi = O(\mathsf{poly}(p))$ and have smoothness bound in $O(\mathsf{poly}(\log(p)))$.

Limitations of the ideal representation for cryptographic applications. The previous paragraphs were dedicated to illustrate the algorithmic benefit of the ideal representation. However, the existence of those efficient algorithms is not necessarily a good thing in the context of cryptography. Indeed, the bottom line is that I reveals pretty much everything there is to know about the two curves E_1, E_2 and the isogenies connecting them. Thus, there is not much hope to use ideal witness as anything else than secret keys.

Even as secret knowledge, ideal witness have their limitation. For instance, zero-knowledge proofs of ideal witness knowledge appears hard to obtain. Theorem 1 implies the existence of zero-knowledge proof systems for $\mathcal{L}_{\mathsf{isog}}$ under standard cryptographic assumptions such as the existence of one way-functions.

While this result is nice in theory, it is not really helpful to build a practical zero-knowledge proof-system for $\mathcal{L}_{\mathsf{isog}}$.

The goal of our new suborder representation is to address the shortcomings of the ideal representation. In particular, under the hardness of the new SOI problem (see Problem 1), it seems plausible to use suborder witnesses in a public manner. This idea is the basis of pSIDH, the new NIKE scheme that we introduce in Section 6.1. More generally, the gap between ideal and suborder witnesses open interesting cryptographical prospects as we discuss in Section 6.2.

4 A new isogeny representation

In this section, we propose a new way to prove the existence of a D-isogeny between two curves when D is a prime number. We call it the *suborder representation/witness*. Composite degrees require more care and we will argue at the end of Section 4.5 that they do not appear more interesting. We briefly explain how to extend the suborder representation to composite degrees in Appendix A. From now on, unless stated otherwise, D can be assumed to be prime. The suborder representation has also another small limitation: the proof only shows that either E_1, E_2 are D-isogenies or E_1, E_2^p are D-isogenous and works only when $\mathrm{End}(E_1) \not\cong \mathrm{End}(E_2)$. Thus, we consider the alternate language $\mathcal{L}_{\mathsf{p-isog}}$ defined as follows:

$$\mathcal{L}_{\mathsf{p-isog}} = \{(D, E_1, E_2) \in \mathbb{P} \times \mathcal{S}_p^2 | E_1 \neq E_2, E_2^p \text{ and } (D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}} \text{ or } (D, E_1, E_2^p) \in \mathcal{L}_{\mathsf{isog}}\}$$

In Section 4.1, we introduce the mathematical results underlying our new method. The method to extract the new witness from the ideal witness is the goal of Section 4.2. Then, in Section 4.3, we explain how to perform a polynomial-time verification of this new witness. We start with a brief summary of the important ideas in the next paragraph.

A brief overview. Our starting point is Proposition 8 which implies that the quaternion sub-order $\mathbb{Z} + D\mathrm{End}(E_1)$ is embedded inside $\mathrm{End}(E_2)$, if and only if either $\mathrm{End}(E_2) \cong \mathrm{End}(E_1)$ or $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$. Thus, our new witness will be constituted of a maximal order $\mathcal{O} \cong \mathrm{End}(E_1)$ and a concrete embedding of $\mathbb{Z} + D\mathcal{O}$ inside $\mathrm{End}(E_2)$ and this is what we concretely call a *suborder witness*. We highlight that $\mathcal{O}$ is simply given as an order inside $B_{p,\infty}$ (through a basis of 4 quaternion elements), whereas the embedding of $\mathbb{Z} + D\mathcal{O}$ is made of isogenies of smooth degree from E_2 to E_2. The suborder witness can be verified by computing the traces of the endomorphisms revealed in this manner.

4.1 Brandt Invariant and relation with isogenies

The goal of this section is to prove Proposition 8 that links the *Brandt invariant* of an order with isogenies through the Deuring Correspondence.

Proposition 8. *Let $D \neq p$ be a prime number and E_1, E_2 be two supersingular curves, $\mathcal{O} \subset B_{p,\infty}$ is a maximal order isomorphic to* $\mathrm{End}(E_1)$*. The order $\mathbb{Z}+D\mathcal{O}$ is embedded inside* $\mathrm{End}(E_2)$ *if and only if either $j(E_2) \in \{j(E_1), j(E_1)^p\}$ or $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{p-isog}}$.*

The backward direction is obtained by considering the map $\alpha_0 \mapsto [d]+\varphi\circ\alpha_0\circ\hat{\varphi}$ between $\mathrm{End}(E_1)$ and $\mathrm{End}(E_2)$ when there exists $\varphi : E_1 \to E_2$ of degree D. In fact, this map is at the heart of the attacks [Pet17,KMP+20] on the SIDH key exchange and underlies the decryption process of the encryption scheme from [DFFdSG+21].

The forward direction is more subtle and we use the preliminary Lemma 3 over ideals and quaternion orders before using the Deuring Correspondence to translate it over isogenies.

Lemma 3. *Let D be prime number different from p. When $\mathfrak{O} = \mathbb{Z} + D\mathfrak{O}_0$ is embedded in a maximal order $\mathcal{O}$, either $\mathcal{O}$ contains $\mathfrak{O}_0$ or there exists a left-$\mathcal{O}$ integral primitive ideal I of norm D whose right order $\mathcal{O}_0$ contains $\mathfrak{O}_0$.*

Proof. Let us assume that $\mathfrak{O}_0$ is not contained in $\mathcal{O}$. We set $I = \{x \in \mathcal{O}, x\mathfrak{O}_0 \subset \mathcal{O}\}$. First, it is easy to verify that I is an integral left $\mathcal{O}$-ideal since it is contained in $\mathcal{O}$. Then, we are going to see that it has norm D. It suffices to show that $D\mathcal{O} \subsetneq I \subsetneq \mathcal{O}$. To see that $I \neq \mathcal{O}$, it suffices to note that $1 \notin I$ since $\mathfrak{O}_0 \not\subset \mathcal{O}$. Then, with $D\mathfrak{O}_0 \subset \mathcal{O}$ we have $Dx\mathfrak{O}_0 = xD\mathfrak{O}_0 \subset \mathcal{O}$ for every $x \in \mathcal{O}$, which proves that $D\mathcal{O} \subset I$. Finally, to prove that $D\mathcal{O} \neq I$, we take $x_0 \in \mathfrak{O}_0$ and not contained in $\mathcal{O}$. It is clear that $Dx_0 \in I$, but $Dx_0 \notin D\mathcal{O}$. Finally, from the definition of I it is quite clear that $\mathfrak{O}_0$ is contained in $O_R(I)$. This concludes the proof.

Proof. (Proposition 8) The forward direction is simply the translation under the Deuring Correspondence of Lemma 3 applied to $\mathfrak{O}_0 \cong \mathrm{End}(E_1)$. For the backward direction, it is clear that if $\mathrm{End}(E_1) \cong \mathrm{End}(E_2)$, $\mathbb{Z} + D\mathrm{End}(E_1) \hookrightarrow \mathrm{End}(E_2)$. Let us assume that there exists an isogeny $\varphi : E_1 \to E_2$ of degree D (possibly changing E_2 to E_2^p if necessary since $\mathrm{End}(E_2) \cong \mathrm{End}(E_2^p)$). Let us write $\iota : \mathbb{Z} \times \mathrm{End}(E_1) \to \mathrm{End}(E_2)$ defined as $\iota(d, \alpha) = [d] + \varphi \circ \alpha_0 \circ \hat{\varphi}$. It is easily verified that $\iota(\mathbb{Z}, \mathrm{End}(E_1))$ is an order of $\mathrm{End}(E_2)$. Then, with $\mathrm{tr}(\iota(d, \alpha)) = 2d+D\mathrm{tr}(\alpha) = \mathrm{tr}(d+D\alpha)$ and $n(\iota(d,\alpha)) = d^2+D^2n(\alpha)+dD\mathrm{tr}(\alpha) = n(d+D\alpha)$ for all $d, \alpha \in \mathbb{Z}\times\mathrm{End}(E_1)$, so we see that we must have $\iota(\mathbb{Z}, \mathrm{End}(E_1) \cong \mathbb{Z}+D\mathrm{End}(E_1)$.

4.2 Deriving the new witness from the ideal witness

The goal of this section is to introduce an algorithm $\mathsf{IdealToSuborder}$ that takes a maximal order $\mathcal{O}$ and a $\mathcal{O}$-ideal I of norm D and outputs a representation of the embedding $\mathbb{Z} + D\mathcal{O} \hookrightarrow \mathrm{End}(E_2)$. By a representation, we actually mean the embeddings of a *generating family* for $\mathbb{Z} + D\mathcal{O}$ (see Definition 2 below).

Definition 2. *A generating family $\theta_1, \cdots, \theta_n$ for an order $\mathfrak{O}$ is a set of elements in $\mathfrak{O}$ such that any element $\rho \in \mathfrak{O}$ can be written as a linear combination of 1 and $\prod_{j\in\mathcal{I}} \theta_j$ for all $\mathcal{I} \subset \{1, \cdots, n\}$. In that case, we write $\mathfrak{O} = \mathsf{Order}(\theta_1, \ldots, \theta_n)$.*

Our algorithm IdealToSuborder (Algorithm 3) is built upon a SmoothGen sub-algorithm that we will present in Section 5.3. The goal of this algorithm is to compute a generating family $\theta_1, \ldots, \theta_n \in B_{p,\infty}$ of smooth norm for the order $\mathbb{Z} + D\mathcal{O}$ on input $D, \mathcal{O}$. For Proposition 9 and Proposition 11, we are going to assume several things about this SmoothGen algorithm. We summarize them in Assumption 1.

Assumption 1 *The algorithm* $\mathsf{SmoothGen}_F$ *is deterministic, correct and terminates in* $O(\mathsf{poly}(\log(p) + \log(D) + C))$ *where* C *is the size of the coefficients of the basis of the maximal order given in input. It outputs* $n = O(1)$ *quaternion elements whose norms* $F_1, \ldots F_n$ *verify that* $F_i | F$ *and* $F_i = O(\mathsf{poly}(pD))$ *for all* $1 \leq i \leq n$*.*

Remark 1. We hide several heuristics under Assumption 1. We discuss these heuristics in Section 5.3.

IdealToSuborder can be divided in two main parts: SmoothGen to obtain quaternion elements $\theta_1, \ldots, \theta_n$ and an IdealToIsogeny step to convert the ideals $\mathcal{O}_R(I)\theta_i$ to isogenies $\varphi_i : E_2 \to E_2$.

For all the algorithms of this section, we are going to assume that a small constant prime ℓ has been fixed.

Algorithm 3 $\mathsf{IdealToSuborder}(I)$

Require: I an integral ideal of maximal orders inside $B_{p,\infty}$ of norm D.
Ensure: Endomorphisms $\varphi_i : E_2 \to E_2$ such that $\iota : \mathrm{End}(E_2) \xrightarrow{\sim} \mathcal{O}_R(I)$ sends $\varphi_1, \ldots, \varphi_n$ to a generating family $\theta_1, \ldots, \theta_n$ for $\mathbb{Z} + D\mathcal{O}_L(I)$.

1: Compute $D = n(I)$ and $\mathcal{O} = \mathcal{O}_L(I), \mathcal{O}' = O_R(I)$.
2: Compute $\theta_1, \ldots, \theta_n = \mathsf{SmoothGen}_{\ell^\bullet}(\mathcal{O}, D)$.
3: **for** $i \in [1, n]$ **do**
4:     Compute $\varphi_i : E_2 \to E_2 = \mathsf{IdealToIsogeny}_{\ell^\bullet}(\mathcal{O}'\theta_i)$.
5: **end for**
6: **return** $\mathcal{O}, (\varphi_i)_{1 \leq i \leq n}$.

The following lemma indicates that isogenies can always be compressed to a polynomial-sized string.

Lemma 4. *A cyclic isogeny of degree* N *can be compressed as a string of size* $O(\log(p) + \log(N))$*.*

Proof. In this lemma, we don't bother with efficiency so we don't restrict to powersmooth degrees. A representation of any isogeny can be obtained in the following manner. First, one needs the starting curve E, which can be described in $O(\log(p))$. Then, since any isogeny is uniquely defined by its kernel its suffices to use $\mathsf{DetBasis}(E, N)$ to obtain a basis P, Q of $E[N]$. A generator of the kernel of φ, can always be expressed as a linear combination of P, Q whose coefficients x, y are smaller than N. In the end, it suffice to publish $j(E), x, y$ to obtain a representation of φ of size $O(\log(p) + \log(N))$.

Proposition 9. *Under Assumption 1 and GRH,* IdealToSuborder *is correct and terminates in* $O(\mathsf{poly}(\log(pD)))$ *and the output has size* $O(\mathsf{poly}(\log(pD)))$.

Proof. Correctness follows from the correctness of IdealToIsogeny and SmoothGen. Similary to ideals and Lemma 1, left and right orders admits a representation of size $O(\mathsf{poly}(\log(pD)))$. Termination follows from Assumption 1 and Proposition 5 (with $n = O(1)$).

4.3 Verification of the suborder witness

This section focuses on the verification of the witnesses computed with IdealToSuborder. From Proposition 8, we know that it suffices to convince the verifier that $\mathbb{Z} + D\mathrm{End}(E_1)$ is embedded inside $\mathrm{End}(E_2)$ and $\mathrm{End}(E_1) \not\cong \mathrm{End}(E_2)$. The second part is easy to verify, it suffices to compute the j-invariants and verify that neither $j(E_1) = j(E_2)$ nor $j(E_1) = j(E_2)^p$. The first part of the verification is achieved with the endomorphisms $\varphi_1, \dots \varphi_n$. With, Lemma 5, we show that it suffices to check some traces and norms of endomorphisms computed from the $(\varphi_i)_{1 \le i \le n}$.

Lemma 5. *Two orders* $\mathcal{O}_1 = \mathsf{Order}(\theta_1, \dots, \theta_n)$ *and* $\mathcal{O}_2 = \mathsf{Order}(\omega_1, \dots, \omega_n)$ *of rank* 4 *in a quaternion algebra are isomorphic if* $n(\theta_i) = n(\omega_i)$ *for all* $i \in [1, n]$ *and* $\mathrm{tr}(\prod_{j \in \mathcal{I}} \theta_j) = \mathrm{tr}(\prod_{j \in \mathcal{I}} \omega_j)$ *for all* $\mathcal{I} \subset [1, n]$.

Proof. In our setting, two quaternion orders are isomorphic if their norm form are the same. Thus, we are going to give a bijection $\alpha : \mathcal{O}_1 \to \mathcal{O}_2$ and verify that it preserves norm and traces. We label $\theta'_0, \theta'_1, \dots, \theta'_m$ (resp. $\omega'_0, \omega'_1, \dots, \omega'_m$) with $m = 2^n - 1$ the set of multi-products obtained from $\theta_1, \dots, \theta_n$ (resp. $\omega_1, \dots, \omega_n$), the multi-product θ_0 (resp. ω_0) corresponding to the empty set is simply 1. By the definition of a generating family, any element $\alpha \in \mathcal{O}_1$ (resp. $\mathcal{O}_2$) can be written as a linear combination of $\theta'_0, \dots, \theta'_m$ (resp. $\omega'_0, \dots, \omega'_m$). We our going to prove that the map $\alpha : \sum_{i=0}^m x_i \theta'_i \mapsto \sum_{i=1}^m x_i \omega'_i$ is an isomorphism of quaternion orders. It is easy to verify that this map is bijective. It remains to check that it preserves the trace and the norm when $n(\theta'_i) = n(\omega'_i)$ and $\mathrm{tr}(\theta'_i) = \mathrm{tr}(\omega'_i)$ for all $i \in [0, m]$.

The trace being linear, its clear that $\mathrm{tr}(\alpha(\theta)) = \mathrm{tr}(\theta)$ for all $\theta \in \mathcal{O}_1$. For any $\theta = \sum_{i=0}^m x_i \theta'_i$, we have $n(\theta) = \sum_{0 \le i < j \le m} x_i x_j \mathrm{tr}(\theta'_i \hat{\theta}'_j) + \frac{1}{2} \sum_{i=0}^m x_i^2 \mathrm{tr}(\theta'_i \hat{\theta}'_i)$. Thus, we need to prove that we have equality of traces for all $\theta'_i \hat{\theta}'_j$ and $\omega'_i \hat{\omega}'_j$. Since $\mathrm{tr}(ab) = \mathrm{tr}(ba) = \mathrm{tr}(\hat{a}\hat{b})$ and $\mathrm{tr}(a)\mathrm{tr}(b) = \mathrm{tr}(ab) + \mathrm{tr}(\hat{a}b)$ for all $a, b \in B_{p,\infty}$, it suffices to verify the equality $\mathrm{tr}(\prod_{j \in \mathcal{I}} \theta_j) = \mathrm{tr}(\prod_{j \in \mathcal{I}} \omega_j)$ to get the desired result. This also proves that we have equality of norms between θ and $\alpha(\theta)$.

As Lemma 5 indicates, we need to compute some traces for the verification. This will be done by an algorithm $\mathsf{CheckTrace}_M$ (whose description we postpone until Section 5.4) that will verify the validity of the traces modulo the parameter M (see Proposition 18).

Lemma 6 below gives a bound above which equality will hold over $\mathbb{Z}$ if it holds mod$M$. In Appendix B, we will explore the option of choosing a value of

M below the bound of Lemma 6 producing a tradeoff between efficiency and soundness.

Lemma 6. *Given any $\theta \in \mathrm{End}(E_1)$, if $\mathrm{tr}(\theta) = t \mod M$ for $M > 4\sqrt{n(\theta)}$ and $|t| \leq M/2$, then $\mathrm{tr}(\theta) = t$.*

Proof. Over $B_{p,\infty}$, the norm form is $n : (x, y, z, w) \mapsto x^2 + qy^2 + pz^2 + qpw^2$ where $q > 0, p > 0$. Since $\mathrm{tr} : (x, y, z, w) \mapsto 2x$, we can easily verify that $\mathrm{tr}(\theta)^2 < 4n(\theta)$. This gives a bound of $2\sqrt{n(\theta)}$ on the absolute value of $\mathrm{tr}(\theta)$. The result follows.

Algorithm 4 $\mathsf{VerifSuborderProof}_M(x, \pi)$

Require: $x \in \mathbb{P} \times \mathcal{S}_p^2$ and π a suborder witness.
Ensure: A bit indicating if $x \in \mathcal{L}_{\mathsf{p-isog}}$.
1: Parse x as D, E_1, E_2 and $\pi = \mathcal{O}, (s_i)_{1 \leq i \leq n}$.
2: **if** If disc $\mathcal{O} \neq p$ **then**
3:     Return 0.
4: **end if**
5: Compute $\theta_1, \ldots, \theta_n = \mathsf{SmoothGen}_{\ell^\bullet}(\mathcal{O}, D)$.
6: Compute $J = \mathsf{ConnectingIdeal}_{\ell^\bullet}(\mathcal{O}_0, \mathcal{O})$ and $L = \mathsf{KLPT}(J)$.
7: Compute $\psi : E_0 \to E_1'$.
8: **if** $j(E_1) \neq j(E_1')$ **or** $j(E_1) \neq j(E_1')^p$ **then**
9:     Return 0.
10: **end if**
11: **for** $i \in [1, n]$ **do**
12:     Parse s_i as an isogeny of degree $n(\theta_i)$ and compute it as $\varphi_i : E_2 \to F_i$.
13:     **if** $j(F_i) \neq j(E_2)$ **then**
14:         Return 0.
15:     **end if**
16: **end for**
17: **return** $\mathsf{CheckTrace}_M(\varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n, E_2)$.

Proposition 10. *If $M > \max_{1 \leq j \leq n} 2\sqrt{n(\theta_j)^n}$, then for $x \in \mathbb{P} \times \mathcal{S}_p^2$, there exists a suborder witness π such that $\mathsf{VerifSuborderlProof}_M(x, \pi) = 1$ if and only if $x \in \mathcal{L}_{\mathsf{p-isog}}$.*

Proof. Assume that there exists a witness π passing the verification for a given $x = (D, E_1, E_2)$. The check in Step 2 proves that $\mathcal{O}$ is a maximal order of $B_{p,\infty}$. The second verification in Step 8 proves that $\mathrm{End}(E_1) \cong \mathcal{O}$. Finally, the verification is Step 13 proves that the φ_i are endomorphisms of E_2. Then, if $\mathsf{CheckTrace}_M(\varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n, E_2) = 1$, the correctness of $\mathsf{SmoothGen}$,$\mathsf{ChekTrace}$, Lemmas 5 and 6 imply that $\mathbb{Z} + D\mathcal{O}$ is embedded inside $\mathrm{End}(E_2)$ and Proposition 8 proves that $x \in \mathcal{L}_{\mathsf{p-isog}}$.

Now let us take $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{p-isog}}$. By definition there exists an ideal I of norm D and $\mathcal{O}_L(I) \cong \mathrm{End}(E_1)$, $\mathcal{O}_R(I) \cong \mathrm{End}(E_2)$. We are going to show that if $\pi = \mathsf{IdealToSuborder}(I)$, then we have $\mathsf{VerifSuborderProof}_M(x, \pi) = 1$. First, since $\mathcal{O}_L(I)$ is a maximal order, the verification of Step 2 passes succesfully. This is also the case for the verification of Step 8 since $\mathcal{O}_L(I) \cong \mathrm{End}(E_1)$. Then, by the correctness of IdealToSuborder showed in Proposition 9, we have that s_i can be parsed as isogenies $\varphi_i : E_2 \to E_2$ that corresponds to the $\mathcal{O}_R(I)\theta_i$ through the Deuring Correspondence (since SmoothGen is deterministic). Thus, it is clear that CheckTrace will output 1 and this concludes the proof.

With Assumption 1 and Proposition 10, we see that there exists a value $k \in \mathbb{N}$ such that if we take $M = p^k - 1$, the verification algorithm $\mathsf{VerifSuborderProof}_M$ is correct.

Proposition 11. *Let k be as defined above. Under GRH and Assumption 1,* $\mathsf{VerifSuborderProof}_{p^k-1}$ *terminates in probabilistic $O(\mathsf{poly}(\log(p) + \log(D)))$.*

Proof. Since $k = O(\mathsf{poly}(\log(pD)))$ by Proposition 10 and Assumption 1, the result follows from Assumption 1, Propositions 1, 2, 5 and 18

4.4 Evaluating with the suborder witness

In this section, we show that we can evaluate the isogeny φ_π from the suborder witness π (in Section 3.3, we described Algorithm 2 to do that same operation from an ideal witness). The algorithm SuborderEvaluation that we introduce below is going to be one of the major building blocks behind the NIKE scheme of Section 6.1 and the KeyExchange algorithm in particular. In fact, we achieve something slightly less powerful than IdealEvaluation as SuborderEvaluation computes images of cyclic subgroups rather than points. SuborderEvaluation can be extended to perform the same operation as IdealEvaluation but we do not need it here. For the sake of the application of SuborderEvaluation in KeyExhchange, we also choose to give the input as an ideal J rather than a subgroup. The output will then be $\varphi(E[J])$.

The SuborderEvaluation algorithm is built on a subprotocol IdealSuborderNormEquation that we will introduce in Section 5.2. This algorithm is only heuristic and we summarize in Assumption 2, what we expect of this algorithm.

Assumption 2 *The algorithm* $\mathsf{IdealSuborderNormEquation}_F$ *takes in input an integer D, two ideals I, J and outputs an element $\beta \in \mathbb{Z} + DI \cap J$ of norm $n(J)F'$ with $F'|F$. It terminates in expected $O(\mathsf{poly}(\log(pDn(I)n(J))))$ with overwhelming probability for all $F > B$ with $B = O(\mathsf{poly}(\log(pDn(I)n(J))))$.*

The principle of SuborderEvaluation is different from the one of IdealEvaluation. Indeed, as we argue in Section 4.5, solving the alternate path problem (which is the key step in IdealEvaluation) appears hard from the suborder representation. Instead, we propose to use the fact that the embedding of $\mathbb{Z} + D\mathrm{End}(E_1)$ inside $\mathrm{End}(E_2)$ is obtained by push-forward through φ_π. More precisely, this means that $\ker \iota(\beta) = \varphi_\pi(\ker \beta)$ for any $\beta \in \mathbb{Z} + D\mathrm{End}(E_1)$. Thus, to find $\varphi_\pi(E_1[J])$, we want to find an endomorphism $\beta \in \mathbb{Z} + D\mathrm{End}(E_1)$ such that $\ker \beta \cap E_1[n(J)] = E_1[J]$. By definition of $E_1[J]$, and Assumption 2, such a β is exactly found by $\mathsf{IdealSuborderNormEquation}$. After that, it suffices to compute $\ker \iota(\beta) \cap E_2[n(J)]$ and we are done.

Algorithm 5 $\mathsf{SuborderEvaluation}(\pi, D, J)$

Require: π a suborder witness for $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{p-isog}}$ and an ideal J of norm coprime with D.
Ensure: $\perp$ or $\varphi_\pi(E_1[J])$.

```
Take a powersmooth integer T coprime with ℓ and n(J) with B < T < 2B, where
  B is the bound in Assumption 2.
Parse π as O, φ_1, ..., φ_n
if O_L(I) ≇ O then
   Return ⊥.
end if
if VerifSuborderProof_{p^k−1}(x, π) = 0. then
   Return ⊥.
end if
Compute θ_1, ..., θ_n = SmoothGen_{ℓ•}(O, D).
Compute L = ConnnectingIdeal(O_0, O) and I = RandomEquivalentPrimeIdeal(L)
  with I = Lα.
Compute β = IdealSuborderNormEquation_T(D, I, α^{−1} J α).
Express αβα^{−1} = Σ_{I⊂{1,...,n}} c_{i,I} Π_{j∈I} θ_j.
Compute P, Q, a basis of E_1[n(J)].
Compute R, S = Σ_{I⊂{1,...,n}} c_{i,I} Π_{j∈I} φ_j(P, Q).
if S = 0 then
   return ⟨Q⟩.
end if
Compute a = DLP(R, S).
return ⟨P − [a]Q⟩.
```

Proposition 12. *Under GRH, $\mathsf{SuborderEvaluation}$ is correct when the output is not $\perp$ and terminates in probabilistic $O(\mathsf{poly}(\log(pD)) + C_{\mathsf{DLP}}(n(J))$ operations over the $n(J)$ torsion where $C_{\mathsf{DLP}}(n(J))$ is the complexity of the discrete logarithms in groups of order $n(J)$.*

Proof. First, we will prove correctness. The verification at the beginning proves that if the output is not $\perp$, π is a valid suborder witness. When $L = \mathsf{ConnnectingIdeal}(\mathcal{O}_0, \mathcal{O})$ and $I = \mathsf{RandomEquivalentPrimeIdeal(L)}$ with $I = L\alpha$, then if $\beta \in (\mathbb{Z} + DI) \cap \alpha^{-1}J\alpha$, then $\alpha\beta\alpha^{-1} \in (\mathbb{Z} + DL) \cap J \subset (\mathbb{Z} + D\mathcal{O}) \cap J$. This explains that we can decompose $\alpha\beta\alpha^{-1}$ on the generating family $\theta_1, \ldots, \theta_n$. Since π gives a correct embedding of $\mathbb{Z} + D\mathcal{O}$ inside $\mathrm{End}(E_1)$ and so $\sigma = \sum_{\mathcal{I} \subset \{1,\ldots,n\}} c_{i,\mathcal{I}} \prod_{j \in \mathcal{I}} \varphi_j$ is an endomorphism of E_2 whose degree is a multiple of $n(J)$. To conclude the proof of correctness, it suffices to show that $\ker \sigma \cap E_2[n(J)] = \varphi_\pi(E_1[J])$. If $\alpha\beta\alpha^{-1} = [d] + [D]\gamma$ for some $\gamma\mathrm{End}(E_1)$, we have that $\sigma = [d] + \varphi_\pi \circ \gamma \circ \hat{\varphi}_\pi$. Now let us take $P_0 \in E_1[J]$. Since $\alpha\beta\alpha^{-1} \in J$, we have $([d] + [D]\gamma)P_0 = 0$

and $\sigma(\varphi_\pi(P_0)) = [d]\varphi_\pi(P_0) + \varphi_\pi(\gamma \circ \hat{\varphi}_\pi \circ \varphi_\pi(P_0)) = \varphi_\pi(([d]+[D]\gamma)P_0 = 0$. This proves that $\varphi_\pi(E[j]) \subset \ker\sigma \cap E_2[n(J)]$. And we obtain equality since the two subgroups have the same order. Thus, we have showed that our protocol is correct.

The complexity follows from Assumptions 1 and 2, Propositions 1 and 11 and the fact that $n(I) = O(\mathsf{poly}(p))$.

4.5 Deducing the ideal witness from the suborder witness

We saw with Proposition 9 that our new suborder witness can be computed from the ideal witness in polynomial time. The goal of this section is to study the reverse problem of extracting an ideal witness from a suborder witness. We are going to try to argue that this problem is hard. This supposed hardness and the resulting gap between the ideal and suborder representations motivates our new construction. We will discuss cryptographic applications in Section 6 and some of the idea discussed there will specifically rely on the hardness of Problem 1.

Problem 1. (SubOrder to Ideal, SOI) Let $x = (D, E_1, E_2) \in \mathcal{L}_{\mathsf{p-isog}}$, and π be a suborder witness such that $\mathsf{VerifSuborderProof}(x, \pi) = 1$. Compute I, an ideal such that $\mathsf{VerifIdealProof}(x, I) = 1$ or $\mathsf{VerifIdealProof}((D, E_1, E_2^p), I) = 1$.

We will show in Proposition 13 the equivalence of Problem 1 with the problem of computing the endomorphism ring of the codomain from the suborder witness (Problem 2).

Problem 2. (SubOrder to Endormophism Ring (SOER)). Let $x = (D, E_1, E_2) \in \mathcal{L}_{\mathsf{p-isog}}$, and π be a suborder witness such that $\mathsf{VerifSuborderProof}(x, \pi) = 1$. Compute $\mathcal{O}_2 \subset B_{p,\infty}$ with $\mathcal{O}_2 \cong \mathrm{End}(E_2)$.

Proposition 13. *Under Assumption 1 and GRH, The SOI and SOER problems are equivalent.*

Proof. Since $\mathcal{O}_R(I) \cong \mathrm{End}(E_2)$ when $\mathsf{VerifIdealProof}((D, E_1, E_2), I) = 1$, it is clear that breaking the SOIP imply to break the SOERP in polynomial-time. The reverse direction is more complicated.

Assume that $\pi, \mathcal{O}_2$ is given with $\mathsf{VerifSuborderProof}((D, E_1, E_2), \pi) = 1$ and $\mathcal{O}_2 \cong \mathrm{End}(E_2)$. We describe an algorithm finding an ideal witness I for $x \in \mathcal{L}_{\mathsf{isog}}$ (up to swapping E_1 and E_1^p we can assume that it is true). Parse $\pi = \mathcal{O}_1, \varphi_1, \dots, \varphi_n$. The isogenies $\varphi_1, \dots, \varphi_n$ can be translated into ideals using an $\mathsf{IsogenyToIdeal}$ algorithm. In that way, we obtain $\mathcal{O}_2\alpha_1, \dots, \mathcal{O}_2\alpha_n$ principal ideals. Compute $\theta_1, \dots, \theta_n = \mathsf{SmoothGen}(\mathcal{O}_2, D)$. Select $\beta \in \mathcal{O}_1$ such that D is inert in $\mathbb{Z}[\beta]$ and $\gcd(n(\beta), D) = 1$. Express $D\beta$ as a linear combination of $\prod_{j \in \mathcal{I}} \theta_j$ for $\mathcal{I} \subset \langle 1, \cdots, n \rangle$ and compute α as the same linear combination of the $\prod_{j \in \mathcal{I}} \alpha_j$. Compute $J = \mathcal{O}_2\langle \alpha, D \rangle$. Find γ such that $\mathcal{O}_1 = \gamma \mathcal{O}_R(J) \gamma^{-1}$ and output $I = \gamma \overline{J} \gamma^{-1}$.

The important property is that if I_0 is the $\mathcal{O}_1$-ideal that we look for, then $\overline{I_0} = \mathcal{O}_R(I_0)\langle D\beta, D\rangle$ when $\beta \in \mathcal{O}_1$ is such that D is inert in $\mathbb{Z}[\beta]$ and $\gcd(n(\beta), D) = 1$. This is a consequence of [DFFdSG+21, Lemma 3.4]. The rest of the algorithm described above is just to compute the value of $D\beta$ through the isomorphism between $\mathcal{O}_R(I)$ and $\mathcal{O}_2$ to get the $\mathcal{O}_2$-ideal J. Finally we send J back through the inverse isomorphism to compute $I = I_0$.

With the knowledge of $\mathcal{O}_2$, the $\mathsf{IsogenyToIdeal}$ algorithm can be applied and its complexity is polynomial in our parameters due to Proposition 4. The same is true for $\mathsf{SmoothGen}$ due to Assumption 1 and all the other operations are performed over the quaternions and have polynomial complexity.

All the results and algorithms from Sections 4.1 to 4.4 were obtained under the assumption that D is a prime number, but in principle, the suborder representation can also be used for composite degree (under various modifications that we don't explain here, see Appendix A for more details). It is interesting to consider the case where D is not prime in the analysis of the SOIP because there are some cases where it is actually easy to solve. This happens, for instance, when D is powersmooth.

A polynomial time algorithm to solve the composite-SOIP when D is powersmooth. The algorithm below is inspired by the torsion point attacks from [Pet17] and the inversion mechanism in the one-way function from [DFFdSG+21]. Let us fix an element $x = (D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$ where each prime-power factor of D is in $O(\log(p))$. If we write φ for an isogeny of degree D between E_1, E_2, we are going to describe informally an algorithm to compute $\ker\hat{\varphi}$ in $\mathsf{poly}(\log(p))$. Since $\mathrm{End}(E_1)$ is known and D is power-smooth, an ideal witness for $x \in \mathcal{L}_{\mathsf{isog}}$ can be easily derived from $\ker\varphi$ (or equivalently from $\ker\hat{\varphi}$).

Let $D = \prod_{i=1}^m \ell_i^{e_i}$, it suffices to get $\ker\hat{\varphi} \cap E_2[\ell_i^{e_i}]$ for each i to be able to reconstruct $\ker\hat{\varphi}$. Let us fix an $i \in [1, m]$. The main idea introduced in [Pet17] is that if $\alpha = [d] + \varphi \circ \alpha_0 \hat{\varphi}$, then the equality $\ker(\alpha - d) \cap E_2[\ell_i^{e_i}] = \ker\hat{\varphi} \cap E_2[\ell_i^{e_i}]$ depends only on ℓ and $\mathbb{Z}[\alpha_0]$. In particular, when ℓ_i is inert in $\mathbb{Z}[\alpha_0]$, then we have $\ker(\alpha - d) \cap E_2[\ell_i^{e_i}] = \ker\hat{\varphi} \cap E_2[\ell_i^{e_i}]$. It is clear that such an α_0 always exists and that it can be computed in $O(\log(p) + \log(D))$. Once, the correct α_0 is found, $D\alpha_0$ can be expressed as a linear combination of the generating family obtained from $1, \varphi_1, \ldots, \varphi_n$. With the coefficients of this linear combination, it suffices to evaluate $E_2[\ell_i^{e_i}]$ through the $\varphi_1, \ldots, \varphi_n$ and solve a few DLPs to obtain $\ker\hat{\varphi} \cap E_2[\ell_i^{e_i}]$. This algorithm has to be repeated at most $O(\log(D))$ times to obtain the full description of $\ker\hat{\varphi}$.

On prime case vs. composite. Isogenies of degree $D_1 D_2$ can be decomposed as two isogenies of respective degree D_1 and D_2. Given the local-global principle on the objects of $B_{p,\infty}$, if the ideal that we look for can be decomposed as $I_1 \cdot I_2$ where $n(I_i) = D_i$, there does not seem to be any reason why finding I_2 from the suborder witness for $D_1 D_2, E_1, E_2$ should be different from solving Problem 1 when the degree is simply D_2. Once I_2 has been found, it is easy to see that recovering I_1 reduces to an instance of Problem 1 of degree D_1. This informal reasoning justifies that taking D composite shoud only make Problem 1 easier to solve. The efficient algorithm that we described above in the case of powersmooth D leads to the same conclusion. Indeed, in this algorithm we clearly recover each coprime part of the isogeny φ_I independently.

The generic case: a heuristic quantum-subexponential algorithm. This paragraph presents informally the best-known algorithms to solve Problem 1. We will implicitly focus on the prime case which appears to be the hardest case as argued in the previous paragraph. We start by classical algorithms and worst-case complexity estimates before introducing a subexponential quantum algorithm which is assumed to be the best known generic method to solve Problem 1.

We start by analyzing the complexity of the brute-force algorithm. In full generality, for a given D, the brute force will take $O(\min(p, D))$. The idea is that since $\mathrm{End}(E_1)$ is part of the suborder witness, it suffices to enumerate through all $\mathrm{End}(E_1)$ ideals of norm D until $\mathsf{VerifIdealProof}$ passes. There are $O(D)$ such ideals, but since there are only $O(p)$ curves, we need to test at most $O(p)$ of them. Thus, the generic complexity of the brute force is $O(\min(D, p))$. Note that when D is prime, there does not seem to be an adaptation of the meet-in-the-middle attack which is considered to be the most efficient method to find an isogeny of smooth degree between two curves.

Another way to solve the problem in a generic manner is by computing $\mathrm{End}(E_2)$ (see Proposition 13). Without using the proof π as a hint, the complexity is believed to be $\tilde{\Theta}(p^{1/2})$ for classical computers and $\tilde{\Theta}(p^{1/4})$ for quantum computers (see [EHL+20]).

Now, let us look at the algorithm described above for powersmooth D in the generic case. Indeed, the algorithm remains correct and valid for any value of D. The only problem is that it becomes exponentially hard for a generic D. First, we need to be able to perform operations over the D-torsion. The smallest field of definition for the D-torsion can have degree in $\Theta(D)$ over $\mathbb{F}_p$. In that case, any operation over the D-torsion will have exponential complexity. Even assuming that the degree of definition is logarithmic, we still need to perform a D-isogeny computation from its kernel. When D is prime, the best known algorithm has complexity $O(\sqrt{D})$ (see [BFLS20]). Thus, the complexity is exponential in the worst case.

We conclude by introducing a quantum algorithm with sub-exponential complexity in D. For that, we use the result from [KMPW21] that a one-way function $f : \mathcal{E} \to \mathcal{F}$ can be inverted at $f(e)$ by solving an instance of the hidden shift problem when there is a group action $\star : G \times \mathcal{E} \to \mathcal{E}$ for which there exists a malleability oracle: an efficient way to evaluate the function $g \mapsto f(g \star e)$ on any $g \in G$. The hidden shift problem can be solved in quantum subexponential time. The authors from [KMPW21] proposed a key recovery attack on an imbalanced version of the SIDH scheme by using the group action of $(\mathrm{End}(E_1)/D\mathrm{End}(E_1))^*$ on the set of cyclic subgroups of order D. This set is in correspondence with cyclic ideals of norm D inside $\mathrm{End}(E_1)$ and so we can invert the function $I \mapsto E/E[I]$ in subexponential time if we have a malleability oracle. In [KMPW21], it was shown that this malleability oracle could be obtained as soon as the image of a big enough torsion-group was given through the secret isogeny. With our algorithm SuborderEvaluation we presented a way to use the suborder witness π to evaluate φ_I on any torsion subgroup. As a consequence, we can evaluate φ_I on any subgroup of powersmooth suborder and this is more than enough to obtain a malleability oracle with the ideas of [KMPW21]. Thus, we can apply the reduction from [KMPW21] and get a sub-exponential quantum method to solve Problem 1.

Remark 2. The existence of a sub-exponential attack is inevitable as soon as one non-trivial endomorphism $\sigma : E_2 \to E_2$ is revealed. The attack stems from the existence of a group action of $\mathrm{Cl}(\mathbb{Z}[\sigma])$ on the set of $\mathbb{Z}[\sigma]$-orientations (i.e pairs E, ι where $\iota : \mathbb{Z}[\sigma] \hookrightarrow \mathrm{End}(E_1)$, see [CK19,DFFdSG+21] for more on orientations). With the knowledge of σ, one can apply the idea (first introduced by Biasse, Jao and Sankar [BJS14] in the special case where $\mathbb{Z}[\sigma] = \mathbb{Z}[\sqrt{-p}]$) that the algorithm from Childs et al. [CJS14] can be adapted to find a path of powersmooth degree between two $\mathbb{Z}[\sigma]$-oriented curves. When this algorithm is applied between E_2 and E_1, a curve of known endomorphism ring, the path obtained in output allows the attacker to compute the endomorphism ring of E_2. This algorithm has sub-exponential complexity in $\log h(\mathbb{Z}[\sigma])$ as it reduces to an instance of the hidden shift problem.

Further analysis of the security problem. Even after seeing our analysis, the hardness of the SOERP may still come as a surprise to a reader familiar with isogeny-based cryptography. In particular, the fact that we reveal several endomorphisms of E_2 might seem like a very troublesome thing to do. This concern is legitimate: the algorithm from [EHL+20] to compute the endomorphism ring of any supersingular curve is based on the principle that knowing two distinct non-trivial endomorphisms is enough to recover the full endomorphism ring in polynomial-time. The idea is that Bass orders are contained in a small number of maximal orders. Thus, when the two non-trivial endomorphisms generate a Bass order, it suffices to enumerate all the maximal orders containing that same Bass order to find the correct one. The authors from [EHL+20] prove their result under the conjecture that two random cycles will form a Bass order with good probability. However, the endomorphisms that we reveal in the suborder witness are not random cycles. By design, the suborder they generate is not Bass and we known that it is contained in an exponential number of maximal orders (this number is equal to the number of D-isogenies by Lemma 3). As such, when using the endomorphisms of the suborder witness, the algorithm described in [EHL+20] is essentially the brute force attack where each ideal of norm D is tested.

Readers might also be concerned with the quaternion alternate path problem. A way to break the SOERP would be to use the embedding of $\mathbb{Z} + D\mathrm{End}(E_1)$ inside $\mathrm{End}(E_2)$ to compute a path from E_2 to a curve E_0 of known endomorphism ring. Following the (now standard) blueprint that underlies most of the algorithm in this work, such an attack would be divided in two steps: first a computation over the quaternions (analog to KLPT) and then a conversion through the Deuring Correspondence to obtain an isogeny connecting E_2 to E_0 (analog to IdealToIsogeny). This supposed attack would have to work over orders of non-trivial Brandt invariant rather than maximal orders to exploit the suborder witness. It appears that the first part of this method can be made to work over non-gorenstein orders. In fact, the IdealSuborderNormEquation that we describe in Algorithm 7 is exactly the analog of KLPT for orders of the form $\mathbb{Z} + D\mathcal{O}$. However, the fact that the Brandt-Invariant is non-trivial appears like a serious obstacle to the second part of the proposed attack. Indeed, as the number of curves admitting an embedding of $\mathbb{Z} + D\mathcal{O}$ inside their endomorphism ring is big, it becomes hard to tell which pair of curves are connected by any ideal of the form $(\mathbb{Z} + D\mathcal{O}) \cap J$ (which was not the case for maximal orders because we have almost a $1 - to - 1$ correspondence between curves and maximal orders). Thus, it seems implausible to be able to find a path between E_2 and a given curve E_0 in that manner. Another way of seeing this is that since $\mathbb{Z} + D\mathcal{O}$ is a generic suborder shared by a lot of curves, we cannot compute anything that will be specific to a given curve from the knowledge of $\mathbb{Z} + D\mathcal{O}$ only.

5 Sub-algorithms over the quaternion algebra

In this section, we fill the blanks left in the Section 4. We provide precise descriptions of the algorithms IdealSuborderNormEquation,SmoothGen, and $\mathsf{CheckTrace}_M$ in Sections 5.2 to 5.4 respectively. We recall that the first algorithm is used to evaluate isogenies from the suborder witness in SuborderEvaluation (Algorithm 5 of Section 4.4) and the last two are building blocks for VerifSuborderProof (Algorithm 4 of Section 4.3) for the verification of our new suborder witness. Note that IdealSuborderNormEquation and SmoothGen are only heuristic as for the algorithms from [KLPT14,DFKL+20].

Following the classical approach in the literature [KLPT14,DFKL+20], we take $B_{p,\infty}$ to be the quaternion order generated by $1, i, j, k$ where $i^2 = -q$, $j^2 = -p$ and $k = ij = -ji$ for some small integer q (when $p = 3 \mod 4$ we can take $q = 1$). Then, we assume that $\mathcal{O}_0 \subset B_{p,\infty}$ is a special extremal order containing a suborder with orthogonal basis $\langle 1, \omega, j, \omega j \rangle$ where $\mathbb{Z}[\omega] \subset \mathbb{Q}[i]$ is a quadratic order of small discriminant.

5.1 Algorithms from previous works

In the next sections, we rely upon several algorithms existing in the literature. The full version of [DFKL+20] is a good reference for all these algorithms. We briefly recall their purpose next.

- RandomEquivalentPrimeIdeal(I), given a left $\mathcal{O}_0$-ideal I, finds an equivalent left $\mathcal{O}_0$-ideal of prime norm.
- IdealModConstraint(I, γ), given an ideal I of norm N, and $\gamma \in \mathcal{O}_0$ of norm n coprime with N, finds $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $\mu_0 = j(C_0 + \omega D_0)$ satisfies $\gamma\mu_0 \in I$.

– $\mathsf{EichlerModConstraint}(I, \gamma)$, given an ideal I of norm N, and $\gamma \in \mathcal{O}_0$ of norm n coprime with N, finds $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $\mu_0 = j(C_0 + \omega D_0)$ satisfies $\gamma\mu_0 \in \mathbb{Z} + I$.
– $\mathsf{StrongApproximation}_\mathsf{F}(N, C_0, D_0)$, given a prime N and $C_0, D_0 \in \mathbb{Z}$, finds $\mu = \lambda\mu_0 + N\mu_1 \in \mathcal{O}_0$ of norm dividing F, with $\mu_0 = j(C_0 + \omega D_0)$. We write $\mathsf{StrongApproximation}_{\ell^\bullet}$ when the expected norm is a power of ℓ.

Remark 3. The $\mathsf{StrongApproximation}_F$ algorithm was originally introduced for a prime number N in [KLPT14]. The probability of success depends on the factorization of F and some quadratic reduosity condition mod N. In general when N is prime, this condition has a $1/2$ chance to be satisfied heuristically. We can easily extend StrongApproximation to the case of composite N (and this is the version that we use in the algorithms below) if we allow the success probability to decrease. The case where N has two large primes divisors is treated in [DFKL+20], and they show that the success probability is $1/4$. In general, it is easy to see that the success probability is $1/2^k$ where k is the number of distinct prime divisors of N. Below, we are going to use the algorithm with N having at most three large prime divisors.

5.2 Solving Norm Equations inside non-Gorenstein orders

In this section, we extend the range of 4-dimensional lattices $\Lambda \subset B_{p,\infty}$ inside which we know how to solve norm equations. The first algorithms targetting that task were introduced in [KLPT14] where Λ was either a special extremal maximal order like $\mathcal{O}_0$ or an ideal of maximal orders. In [DFKL+20], new methods were introduced to work inside Eichler orders and their ideals, thus covering lattices of the form $\mathbb{Z} + I$ and $(\mathbb{Z} + I) \cap J$ where I, J are cyclic integral ideals with $\gcd(n(I), n(J)) = 1$. We continue this trend of work by exploring the case of non-Gorenstein orders with Gorenstein closure equal to Eichler orders and their ideals. Concretely, this means lattices of the form $\mathbb{Z}+DI$ and $(\mathbb{Z}+DI)\cap J$ where I, J are cyclic integral ideals and $\gcd(n(I), n(J), D) = 1$.

Our motivation is the resolution of norm equations inside $\mathbb{Z} + D\mathcal{O}$ for any maximal order $\mathcal{O} \subset B_{p,\infty}$. In the particular case where $\mathcal{O}$ is a maximal extremal order as $\mathcal{O}_0$, an algorithm to find elements of given norm inside $\mathbb{Z} + D\mathcal{O}$ was introduced in [Pet17]. Unfortunately, the generic case requires a different treatment. We apply the idea from De Feo et al. in [DFKL+20] that consist in restricting the resolution to the suborder $(\mathbb{Z} + D\mathcal{O}) \cap \mathcal{O}_0$. Since $\mathcal{O} \cap \mathcal{O}_0 = \mathbb{Z} + I$ where $I = \mathsf{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$, our main tool is an algorithm EichlerSuborderNormEquation to solve norm equations inside $\mathbb{Z} + DI = (\mathbb{Z} + D\mathcal{O}) \cap (\mathbb{Z} + I)$. This algorithm is going to be the main building block of SmoothGen (whose description we give in Section 5.3). In the end of this section, we show with IdealSuborderNormEquation how to extend EichlerSuborderNormEquation to solve norm equations inside $(\mathbb{Z} + DI) \cap J$ where $\gcd(n(J), n(I)) = 1$.

To clarify the explanations, we try to extract a pattern in the formulations of the algorithms from [KLPT14,DFKL+20] and ours. We will explain how the ideas from [KLPT14,DFKL+20] fit into the common framework before introducing our approach. We hope that it might provide some insights on these algorithms and help the reader understand how they work and how they were designed.

Each algorithm is parametrized by two integers N_1, N_2. We use an abstract symbol F to denote the targetteted norm of the output. As for StrongApproximation (see Section 5.1), in practice F is going to be either $\ell^\bullet$ or a powersmooth integer T. The goal is to find elements of norm dividing F. When $F = \ell^\bullet$, we mean that the norm must be a power of ℓ. The algorithms can be decomposed as follows:

1. Find γ satisfying a set of conditions and having a norm dividing $N_1 F$.
2. Find $C, D \in \mathbb{Z}$ such that $\gamma j(C + D\omega) \in \Lambda$.
3. Compute $\mu = \mathsf{StrongApproximation}_F(N_2, C, D)$.
4. Output $\gamma(j(C + D\omega) + N_2\mu)$.

The goal of these "conditions" on γ in the first step is to ensure that the second step will always have a solution. As we are going to see, the only real difference between the several algorithms are the values of N_1, N_2 and these conditions on γ. The second step is always solved using linear algebra mod N_2. When N_2 is composite, we will decompose it in sub-operations modulo the different factors before using a CRT to put everything together.

In the rest of this section, we may assume for simplicity that ideals have prime norm. When not, the algorithm EquivalentRandomPrimeIdeal can be used to reduce the computation to the prime case. The first algorithm fitting the framework above was introduced in [KLPT14] and targetted the case where Λ is an $\mathcal{O}_0$-ideal of norm N. The condition on γ is summarized by Lemma 7 that is a reformulation of some of the results from [KLPT14]. We have $N_1 = N$ and $N_2 = N$.

Lemma 7. *[KLPT14] Let I be an $\mathcal{O}_0$ ideal of norm N and $\gamma \in \mathcal{O}_0$. When $\gcd(n(\gamma), N^2) = N$, there exists $C, D \in \mathbb{Z}$ such that $\gamma j(C + D\omega) \in I$ with overwhelming probability.*

Thus, a correct γ is any element of $\mathcal{O}_0$ of norm NF' where $F'|F$.

The goal of the authors of [DFKL+20] was to obtain a generalization of the algorithm of [KLPT14] when Λ is an $\mathcal{O}$-ideal K for any maximal order $\mathcal{O}$ (and not just the special case $\mathcal{O}_0$). To do that, they proposed to solve the norm equation inside $K \cap \mathcal{O}_0$ which can be written as $(\mathbb{Z} + I) \cap J$ for two $\mathcal{O}_0$-ideals I, J. To achieve that goal they started by implicitly introducing a method to solve the norm equation inside $\mathbb{Z} + I$ before combining that with the ideas from [KLPT14] to get the full method.

For the case $\Lambda = \mathbb{Z} + I$ where I has norm N, the condition on γ can be summarized with Lemma 8. In that case, $N_1 = 1$ and $N_2 = N$.

Lemma 8. *[DFKL+20] Let I be an $\mathcal{O}_0$ ideal of norm N. When $\gcd(\gamma, N) = 1$, there exists $C, D \in \mathbb{Z}$ such that $\gamma j(C+D\omega) \in \mathbb{Z}+I$ with overwhelming probability.*

When $\Lambda = (\mathbb{Z}+I)\cap J$ with $n(I) = N$ and $n(J) = N'$, the solution presented in [DFKL+20, Section 5] is simply obtained by combining Lemmas 7 and 8 with $N_1 = N'$, $N_2 = NN'$.

Norm equations inside $\mathbb{Z}+DI$. Next, we explain our method for the case $\Lambda = \mathbb{Z}+DI$. This time, we need γ to satisfy more conditions than a simple constraint on its norm. We will introduce the necessary condition in Proposition 14. The constraint proves to be slightly inconvenient, and will impact the size of the final solution, but we managed to find a way to keep some control on the norm of γ while ensuring that the linear algebra step always have a solution.

Proposition 14. *Let I be an integral left $\mathcal{O}_0$-ideal of norm N and let D be a distinct prime number. If $\gamma \in \mathcal{O}_0$ can be written as $j(C_2+\omega D_2)+D\mu_2$ with $\mu_2 \in \mathcal{O}_0$ and γ has norm coprime with N, then there exists $C_1, D_1 \in \mathbb{Z}$ such that $\gamma j(C_1+\omega D_1) \in \mathbb{Z}+DI$.*

Proof. If γ has norm coprime with N, we know from [DFKL+20] that there exists C_0, D_0 such that $\gamma j(C_0+\omega D_0) \in \mathbb{Z}+I$ (this is Lemma 8). Then, if we set $C_2' = -D_2'C_2(D_2)^{-1} \mod D$ for any D_2', it is easy to verify that $\gamma j(C_2'+\omega D_2') \in \mathbb{Z}+D\mathcal{O}_0$. Hence, if C_1, D_1 satisfies $C_1, D_1 = C_0, D_0 \mod N$, $C_1, D_1 = C_2', D_2' \mod D$ and $\gcd(N,D) = 1$, we have that $\gamma j(C_1+\omega D_1) \in \mathbb{Z}+D\mathcal{O}_0 \cap (\mathbb{Z}+I) = \mathbb{Z}+DI$. By the CRT, we know we can find such C_1, D_1.

With Proposition 14, we see that we must take $N_1 = 1$ and $N_2 = ND$ and that we must also apply a strong approximation $\mod D$ to compute exactly γ. When we apply these ideas to the framework described above, we obtain $\mathsf{EichlerSuborderNormEquation}$.

Algorithm 6 $\mathsf{EichlerSuborderNormEquation}_F(D, I)$

Require: I a left $\mathcal{O}_0$-ideal of norm N coprime with D.
Ensure: $\beta \in \mathbb{Z}+DI$ of norm dividing F.
1: Select a random class $(C_2 : D_2) \in \mathbb{P}^1(\mathbb{Z}/D\mathbb{Z})$.
2: Compute $\mu_2 = \mathsf{StrongApproximation}_F(D, C_2, D_2))$ and set $\gamma = j(C_2+\omega D_2)+D\mu_2$. If the computation fails, go back to Step 1.
3: Compute $(C_0 : D_0) = \mathsf{EichlerModConstraint}(\gamma, I)$.
4: Sample a random D_2' in $\mathbb{Z}/D\mathbb{Z}$, compute $C_2' = -D_2'C_2(D_2)^{-1} \mod D$.
5: Compute $C_1 = \mathsf{CRT}_{N,D}(C_0, C_2')$, $D_1 = \mathsf{CRT}_{N,D}(D_0, D_2')$.
6: Compute $\mu_1 = \mathsf{StrongApproximation}_F(ND, C_1, D_1))$. If it fails, go back to step 1.
7: **return** $\beta = (j(C_2+\omega D_2)+D\mu_2)(j(C1+\omega D_1)+ND\mu_1)$.

Proposition 15. *Assuming various plausible heuristics, when N, D are distinct prime, Algorithm 6 terminates in expected $O(\mathsf{poly}(\log(DN)))$ and outputs an element of $\mathbb{Z}+DI$ of norm dividing F. The expected norm is in $O(\mathsf{poly}(p, D, N))$.*

Proof. As mentioned in Remark 3, under plausible heuristics the algorithm $\mathsf{StrongApproximation}_F(D,\cdot)$ finds a solution of norm dividing F with heuristic probability at least $1/2$ in polynomial time. As a result of Proposition 14, $\mathsf{EichlerModConstraint}$ always succeeds in finding a solution $(C_0 : D_0)$. Then, the second $\mathsf{StrongApproximation}$ has a $1/4$ success probability when N, D are prime. Assuming that a new choice of $(C_2 : D_2)$ randomizes $(C_1 : D_1)$ sufficiently we can show that a solution can be found with overwhelming probability after a constant number of repetitions. This proves the algorithm's termination.

For correctness, we can verify easily that $j(C_2+D_2\omega)j(C_2'+\omega D_2') \in \mathbb{Z}+D\mathcal{O}_0$. Since $\beta - j(C_2+D_2\omega)j(C_2'+\omega D_2') \in D\mathcal{O}_0$ this proves that $\beta \in \mathbb{Z}+D\mathcal{O}_0$. By the correctness of $\mathsf{EichlerModConstraint}$ and the fact that $N\mathcal{O}_0$ is contained in I we can also show that $\beta \in \mathbb{Z}+I$. Hence, $\beta \in (\mathbb{Z}+D\mathcal{O}_0)\cap(\mathbb{Z}+I) = \mathbb{Z}+DI$. The estimates provided in [DFKL$^+$20] allow us to predict that we can find a solution β of norm $F'|F$ where $\log F' \sim 2\log_\ell(p) + 6\log_\ell(D) + 3\log_\ell(N)$. This comes from the fact that a strong approximation mod N' can find solutions of norm approximately equal to pN'^3.

Norm equations inside $(\mathbb{Z}+DI)\cap J$. We set $N = n(I)$ and $N' = n(J)$. For this final case, it suffices to combine Lemmas 7 and 8 and Proposition 14 and take $N_1 = N'$, $N_2 = NN'D$. This yields Algorithm 7.

Algorithm 7 $\mathsf{IdealSuborderNormEquation}_F(D, I, J)$

Require: An integer$D$, I, J two left $\mathcal{O}_0$-ideals of norm N, N' with $\gcd(N, N', D) = 1$.
Ensure: $\beta \in \mathbb{Z}+DI\cap J$ of norm $N'F'$ where $F'|F$.
1: Select a random class $(C_2 : D_2) \in \mathbb{P}^1(\mathbb{Z}/D\mathbb{Z})$.
2: Compute $\mu_2 = \mathsf{StrongApproximation}_{FN'}(D, C_2, D_2))$ and set $\gamma = j(C_2+\omega D_2) + D\mu_2$. If the computation fails or if $\gcd(n(\gamma), N') = 1$, go back to Step 1.
3: Compute $(C_0 : D_0) = \mathsf{EichlerModConstraint}(\gamma, I)$.
4: Compute $(C_3 : D_3) = \mathsf{IdealModConstraint}(\gamma, J)$.
5: Sample a random D_2' in $\mathbb{Z}/D\mathbb{Z}$, compute $C_2' = -D_2'C_2(D_2)^{-1} \mod D$.
6: Compute $C_1 = \mathsf{CRT}_{N,D,N'}(C_0, C_2', C_3)$, $D_1 = \mathsf{CRT}_{N,D,N'}(D_0, D_2', D_3)$.
7: Compute $\mu_1 = \mathsf{StrongApproximation}_F(NDN', C_1, D_1))$. If it fails, go back to step 1.
8: **return** $\beta = (j(C_2+\omega D_2) + D\mu_2)(j(C1+\omega D_1) + NN'D\mu_1)$.

Proposition 16. *Under various plausible heuristics, Assumption 2 holds.*

Proof. Due to Lemmas 7 and 8 and Proposition 14, we know that we can find $(C_0 : D_0), (C_3 : D_3)$ and $(C-2' : D_2')$ with overwhelming probability and that the result will be correct. The computation takes $O(\mathsf{poly}(\log(DNN')))$ since it consists of linear algebra $\mod D, N, N'$. The executions of $\mathsf{Strong\ Approximations}$ terminates in probabilistic polynomial time and output a value with constant probability. So the global computations terminates in probabilistic $O(\mathsf{poly}(\log(DNN')))$. It is correct because $\mathsf{StrongApproximation}$ is correct. The computation succeeds as soon as the target norm can have size bigger than $2\log_\ell(p) + 6\log_\ell(D) + 3\log_\ell(N) + 2\log_\ell(N')$ (the first approximation give an element of size $\sim pD^3/N'$ and the second $p(DNN')^3$).

5.3 Computing a smooth generating family

In this section, we describe how to perform the SmoothGen protocol. The goal of this algorithm is to find a generating family of smooth norm for the order $\mathbb{Z} + D\mathcal{O}$ from the inputs $D, \mathcal{O}$. The idea is quite straightforward: sample several random smooth elements until we obtain a generating family.

For a generic order $\mathcal{O}$, we have introduced with EichlerSuborderNormEquation, a method to solve norm equations over $(\mathbb{Z} + D\mathcal{O} \cap \mathcal{O}_0) = \mathbb{Z} + DI$ for the $\mathcal{O}_0, \mathcal{O}$-ideal I. Thus, we propose to repeat the following procedure: generate a random ideal I between $\mathcal{O}$ and $\mathcal{O}_0$ and then apply EicherSuborderNormEquation. Experimental results show that taking three elements in that manner is already enough. We formulate this as Conjecture 1.

Conjecture 1. Let $\mathcal{O}$ be a maximal order in $B_{p,\infty}$. Let I_1, I_2, I_3 be random $\mathcal{O}_0$-ideals of prime norms with $\alpha_i \mathcal{O}_R(I_i)\alpha_i^{-1} = \mathcal{O}$ for some $\alpha_i \in B_{p,\infty}^*$. If $\theta_1, \theta_2, \theta_3$ are random outputs of $\mathsf{EichlerSuborderNormEquation}(\mathsf{D}, \mathsf{I_i})$ for $i = 1, 2, 3$ and then $\mathbb{Z} + D\mathcal{O} = \mathsf{Order}(\alpha_1\theta_1\alpha_1^{-1}, \alpha_2\theta_2\alpha_2^{-1}, \alpha_3\theta_3\alpha_3^{-1})$ with good probability.

Algorithm 8 $\mathsf{SmoothGen}_F(\mathcal{O}, D)$

Require: A maximal order $\mathcal{O}$ and a prime D.
Ensure: A generating family $\theta_1, \ldots, \theta_3$ for $\mathbb{Z} + D\mathcal{O}$ where each θ_j has norm ℓ^{e_j}.

1: Set $L = \emptyset$ and $I_0 = \mathsf{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$.
2: **while** There does not exist $\theta_1, \theta_2, \theta_3 \in L$ s.t $\mathbb{Z} + D\mathcal{O} = \mathsf{Order}(\theta_1, \theta_2, \theta_3)$ **do**
3: $\quad I = \mathsf{RandomEquivalentPrimeIdeal}(I_0)$ and $I = I_0\alpha$.
4: $\quad$ Compute $\theta = \mathsf{EichlerSuborderNormEquation}_F(D, J)$.
5: $\quad L = L \cup \{\alpha\theta\alpha^{-1}\}$.
6: **end while**
7: **return** $\theta_1, \theta_2, \theta_3$.

Proposition 17. *Assuming Conjecture 1 and various plausible heuristics, Assumption 1 holds.*

Proof. By Proposition 1 the runnning time of ConnectingIdeal is polynomial in the size of the basis coefficients. The same holds for RandomEquivalentPrimeIdeal and the output of this algorithm have norms in $O(\mathsf{poly}(p))$. By Conjecture 1, $n = 3$ and we need only to repeat a polynomial number of times the algorithm EichlerSuborderNormEquation which terminates in polynomial time by Proposition 15 and the outputs have norm in $O(\mathsf{poly}(pD))$. By the termination condition, the output is a generating family of $\mathfrak{O}$. Algorithm 8 is randomized by design, but it can be easily made deterministic by setting a deterministic way to generate the randomness.

5.4 Checking traces

In this section, we present an algorithm $\mathsf{CheckTrace}_M$ to perform the verification of the suborder witness.

Computing the trace of an endomorphism is a well-studied problem, as it is the primary tool of the point counting algorithms such as SEA [Sch95]. For our application the task is even simpler as we merely have to verify the correctness of the alleged trace value and not compute it. With the formula $\mathrm{tr}(\theta) = \theta + \hat{\theta}$, it suffices to evaluate θ and $\hat{\theta}$ on a basis of the M-torsion, and then verify the relation. In particular, we do not need M to be smooth since we just to check equality.

Algorithm 9 $\mathsf{CheckTrace}_M(E, \varphi_1, \dots, \varphi_n, \theta_1, \dots, \theta_n)$

Require: $\theta_1, \dots, \theta_n$, n endomorphisms of E and n elements of $B_{p,\infty}$ $\omega_1, \dots, \omega_n$.
Ensure: A bit b equal to 1 if and only if $\mathrm{tr}(\theta_i) = \mathrm{tr}(\omega_i) \mod M$ for all $i \in [1, n]$.
1: Compute P, Q a basis of $E[M]$ over the appropriate field extension. Set $b = 1$.
2: **for** All $\mathcal{I} \subset [1, n]$ **do**
3: Set $\theta_{\mathcal{I}} = \prod_{j \in I} \theta_j$ and $\varphi_{\mathcal{I}} = \prod_{j \in I} \varphi_j$.
4: Verify $\varphi_{\mathcal{I}}(R) + \hat{\varphi}_{\mathcal{I}}(R) = [\mathrm{tr}(\theta_{\mathcal{I}})]R$ for $R \in \{P, Q\}$. If not, set $b = 0$.
5: **end for**
6: **return** b.

Proposition 18. *When $M = p^k - 1$, $n = O(1)$ and $\deg \varphi_i = O(\mathsf{poly}(p))$ and have smoothness bound in $O(\mathsf{poly}(\log(p)))$ for all $1 \leq i \leq n$, $\mathsf{CheckTrace}_M$ terminates in $O(\mathsf{poly}(k \log(p))$*

Proof. If $M = p^k - 1$, then P, Q are defined over $\mathbb{F}_{p^k}$ and so operations over the M-torsions have $O(\mathsf{poly}(k \log(p))$ complexity. By the assumption on the degree of the φ_i, computing all the $\varphi_{\mathcal{I}}(P, Q)$ can be done in $O(\mathsf{poly}(\log(p)))$ since $n = O(1)$ and this concludes the proof.

6 Prospects for isogeny-based cryptography

In this section, we discuss how to use our new suborder representation as a building block for cryptographic primitive.

6.1 A new NIKE based on a generalization of SIDH for big prime degrees.

We present here pSIDH (prime-SIDH) a new NIKE scheme. It is based on a SIDH-style isogeny diagram (see Fig. 1 and Fig. 2) but with prime degrees. For secret keys we propose to use ideal witnesses and then take suborder witnesses as public keys. The key exchange will be made possible with SuborderEvaluation (Algorithm 5 of Section 4.4). In terms of security, the pSIDH key recovery problem is exactly the SOIP and the NIKE is secure under the hardness of a decisional variant of Problem 1 in a similar manner to SIDH with the CSSI and SSDDH problems introduced in [JDF11]. We stress that we leave efficiency considerations to future work and merely show that the scheme can be executed in polynomial-time.

The idea of SIDH is the following: the two participants Alice and Bob generate isogenies φ_A, φ_B of degree $\gcd(N_A, N_B) = 1$. Their public keys are the curves E_A, E_B, together with additional pieces of information to make possible the computation of the two push-forward isogenies $[\varphi_A]_*\varphi_B$ and $[\varphi_B]_*\varphi_A$ depicted in Fig. 1. It is possible to show that the codomains of these push-forward isogenies are isomorphic (thus providing a way to derive the common key from $j(E)$). In the case of SIDH (or the B-SIDH variant [Cos20]), the degrees are smooth which makes isogeny computations efficient from the kernels if the N_A, N_B torsion is defined over $\mathbb{F}_{p^2}$. We have $\ker[\varphi_A]_*\varphi_B = \varphi_A(\ker \varphi_B)$ and this is why Alice's SIDH-public key is the curve E_A together with $\varphi_A(P_B), \varphi_A(Q_B)$ where $\langle P_B, Q_B \rangle = E_0[N_B]$ (and the reverse for Bob's).

$$\begin{tikzcd}
& E_A \arrow[dr, "{[\varphi_A]_*\varphi_B}"] & \\
E_0 \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E \\
& E_B \arrow[ur, "{[\varphi_B]_*\varphi_A}"'] &
\end{tikzcd}$$

Fig. 1. SIDH-isogeny diagram.

To do the same thing for two prime degrees D_A, D_B, we need a new method to compute the codomain of the push-forward isogenies (since the Vélu Formulas are not practical for prime degrees). We propose to use the ideal witnesses as secret keys and the suborder witnesses as public keys. The computation of the common key $j(E)$ can be dones as follows. Given an ideal I of norm D_A and the suborder $\mathbb{Z} + D_B\mathcal{O}$, it is possible to find an element $\theta \in (\mathbb{Z} + D_B\mathcal{O}_0) \cap I$ of norm $D_A S$ where S is a powersmooth integer with the algorithm IdealSuborderNormEquation (Algorithm 7 in Section 5.2). The embedding $\iota_B : \mathbb{Z} + D_B\mathcal{O}_0 \hookrightarrow \mathrm{End}(E_B)$, is obtained by pushing forward the embedding of $\mathbb{Z} + D_B\mathcal{O}_0$ inside $\mathrm{End}(E_0)$ through φ_B and so we have $\iota_B(\theta) = \psi_A \circ [\varphi_B]_*\varphi_A$ where ψ_A has degree S. Thus, using π_B, the suborder representation of φ_B, we can use SuborderEvaluation to compute $\ker \hat{\psi}_A$ and $\hat{\psi}_A$. The codomain of $\hat{\psi}_A$ is isomorphic to E and so the common secret $j(E)$ can be derived from that.

These ideas are summarized in Fig. 2 and the full description of the key exchange mechanism is given as Algorithm 11. The key generation algorithm is also described in Algorithm 10. The public parameters should include a prime p and a starting curve E_0 together with a description of $\mathrm{End}(E_0)$.

$$\begin{tikzcd}
 & E_A \arrow[dr, "{[\varphi_A]_*\varphi_B}"] \arrow[dr, bend left, "\hat{\psi}_B"] & \\
E_0 \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"'] & & E \\
 & E_B \arrow[ur, "{[\varphi_B]_*\varphi_A}"'] \arrow[ur, bend right, "\hat{\psi}_A"'] &
\end{tikzcd}$$

Fig. 2. pSIDH-isogeny diagram.

Algorithm 10 $\mathsf{KeyGeneration}(D)$

Require: A prime number $D \neq p$.

Ensure: The pSIDH public key $\mathsf{pk} = E, \pi$ and the pSIDH secret key $\mathsf{sk} = I$ where π is a suborder witness and I an ideal witness for $(D, E_0, E) \in \mathcal{L}_{\mathsf{p-isog}}$.

1: Sample I as a random $\mathcal{O}_0$-ideal of norm D.

2: Compute $\pi = \mathsf{IdealToSuborder}(I)$ and set E as the domain of the endomorphisms in π.

3: **return** $\mathsf{pk}, \mathsf{sk} = (E, \pi), I$.

Proposition 19. *Under GRH, Assumption 1, Assumption 2,* $\mathsf{KeyExchange}$ *terminates in expected* $\mathsf{poly}(\log(pD'D)$.

Proof. Since $B = O(\mathsf{poly}(\log(pDD')))$, T can be be chosen with a smoothness bound equal in $O(\mathsf{poly}(\log(pDD')))$. Thus, the final computation of ψ can be done in $O(\mathsf{poly}(\log(pD'D)))$. The remaining computations terminate in expected $O(\mathsf{poly}(\log(pD'D)))$ due to Assumptions 1 and 2 and Propositions 1, 11 and 12.

Proposition 20. *Let $D_A, D_B \neq p$ be two distinct prime numbers. If $E_A, \pi_A, I_A =$* $\mathsf{KeyGen}(D_A)$ *and $E_B, \pi_B, I_B =$* $\mathsf{KeyGen}(\mathsf{D_B})$*, then*

$$\mathsf{KeyExchange}(I_A, D_B, E_B, \pi_B) = \mathsf{KeyExchange}(I_B, D_A, E_A, \pi_A).$$

Proof. Let us write φ_A, φ_B the isogenies corresponding to the two ideals I_A, I_B. Then, the quaternion element $\overline{\alpha_A^{-1}\theta_A\alpha_A}$ (resp. B) obtained at Step 8 during the execution of $\mathsf{KeyExchange}(I_A, D_B, E_B, \pi_B)$ (resp. B/A) corresponds to the endomorphism $\psi_{0,A} \circ \varphi_A \in (\mathbb{Z} + D_B \mathrm{End}(E_0)) \cap I_A \hookrightarrow \mathrm{End}(E_B)$ (resp. B/A/B/A). Since it is contained in $(\mathbb{Z} + D_B \mathrm{End}(E_0)) \cap I_A$, this endomorphism is equal to $\psi_A \circ [\varphi_B]_* \varphi_A$ (resp. B/A/B) where $\hat{\psi}_A = [\varphi_B]_* \hat{\psi}_{A,0}$ for some isogeny $\psi_{A,0} : E_0 \to E_A$ (resp. B/A/B). In particular, the codomain of $\hat{\psi}_A$ (resp. B) is isomorphism to the codomain of $[\varphi_B]_* \varphi_A$ (resp. A/B). Thus, by definition of push-forward isogenies and Proposition 12, the two j-invariants obtained at the end of the two executions of $\mathsf{KeyExchange}$ are equal.

Algorithm 11 $\mathsf{KeyExchange}(I, D', E', \pi)$

Require: I an ideal of degree D and a prime $D' \neq D, p$. A curve E' and a suborder witness π.
Ensure: A j-invariant or $\perp$.

Parse $\pi = (\mathcal{O}, \varphi_1, \ldots, \varphi_n)$.
Compute $\theta_1, \cdots, \theta_n = \mathsf{SmoothGen}_{\ell^\bullet}(\mathcal{O}_0, D')$.
if $!\mathsf{VerifSuborderProof}_{p^k-1}((D', E_0, E'), \pi)$ or $\mathcal{O} \neq \mathcal{O}_0$ **then**
  Return $\perp$.
end if
Take a powersmooth integer T coprime with ℓ with $B < T < 2B$ where B is the bound in Assumption 2 and T has the smallest possible smoothness bound.
Compute $L = \mathsf{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$ and $J = \mathsf{RandomEquivalentPrimeIdeal}(L)$ with $J = L\alpha$.
Compute $\theta = \mathsf{IdealSuborderNormEquation}_T(D', J, I)$.
Factorize $T = \prod_{i=1}^m \ell_i^{e_i}$.
Set $G = \langle 0_{E'} \rangle$.
for $i \in [1, m]$ **do**
  Compute $J_i = \mathcal{O}_0\langle \overline{\alpha^{-1}\theta\alpha}, \ell_i^{e_i} \rangle$.
  $G = G + \mathsf{SuborderEvaluation}(\pi, D', J_i)$.
end for
Compute $\psi : E' \to E'/G$.
return $j(E'/G)$.

Security. By design, we have the algorithm $\mathsf{VerifSuborderWitness}$ to validate public keys and so we obtain a NIKE. For key validation, the public parameters for pSIDH also include a value $M = p^k - 1$ as in Proposition 10. By design, the pSIDH key recovery problem is simply the SOIP (Problem 1). To prove security of our key exchange, we need a decisional variant which we call the pSSDDH (prime supersingular DDH) problem (see Problem 3).

Problem 3. (pSSDDH) Let $D_A, D_B \neq p$ be two distinct prime numbers and $E_A, \pi_A, I_A = \mathsf{KeyGen}(D_A)$ and $E_B, \pi_B, I_B = \mathsf{KeyGen}(\mathsf{D_B})$. The problem is to distinguish between the two distributions:

1. $(E_A, \pi_A), (E_B, \pi_B), E_{AB}$ where $\mathrm{End}(E_{AB}) \cong \mathcal{O}_R(I_A \cap I_B)$.
2. $(E_A, \pi_A), (E_B, \pi_B), E_C$ where E_C is a random curve $N_A N_B$-isogenous to E_0.

With the pSSDDH problem, we can state the security of the key agreement protocol we just outlined. The proof mimicks the one made in [JDF11].

Proposition 21. *Under the pSSDDH assumption, the key-agreement protocol made of Algorithms 10 and 11 is session-key secure in the authenticated-links adversarial model of Canetti and Krawczyk [CK01].*

6.2 Potential for other cryptographic applications

We have introduced a new NIKE scheme, pSIDH, as a way to illustrate the possibilities offered by our new isogeny representation. When making the comparison with SIDH, the two main advantages of our construction are the different security assumption and the non-interactive key validation mechanism. These two properties probably do not make up for the huge efficiency gap between SIDH and pSIDH (see Section 6.3) but they could be important for more complicated primitives. As such, pSIDH should only be considered as a first example of what can be done with our suborder representation. We discuss below other potential applications. We propose directions to explore for future work rather than concrete protocols.

Adaptation of protocols based on SIDH. A lot of isogeny-based primitives are based on the mechanism underlying the SIDH key exchange. We can mention n-party key exchange [AJJS19], signatures [YAJ+17] built upon the SIDH identification scheme from [JDF11], oblivious transfers [BOBN19,dSGOPS20] and oblivious PRF [BKW20].

A multi-party key exchange can easily be designed in the SIDH setting. It suffices to take coprime degrees $D_1, D_2, \ldots, D_n$ and the commutative diamond in Fig. 1 can be extended to an n-dimensional commutative diagram that leads naturally to a multi-party key exchange. The main problem with this protocol in the setting of SIDH is security as it is under serious threat of the most recent advances on torsion point attacks from [KMP+20] (the construction is broken as soon as $n \geq 6$). It seem plausible to adapt this multi-party key exchange to the setting of pSIDH using the successive suborders $\mathbb{Z} + D_i D_j \mathcal{O}$, $\mathbb{Z} + D_i D_j D_k \mathcal{O}, \ldots$. In terms of security, this n-party pSIDH could be addressing some of the shortcomings of the SIDH version. Indeed, as explained in Section 4.5, the composite version of the SOIP (Problem 1) appears to be reducing to the prime case which tends to suggest that the multi-party key exchange could be as secure as the two-party version. Remains to see how exactly the successive suborders can be computed from the suborder representations. We leave that to future work.

Contrary to the multi-party key exchange, the adaptation of SIDH signatures to the setting of pSIDH seems like a complicated task. It would require a zero-knowledge ideal-witness proof of knowledge which seem hard to build as we highlighted in Section 3.3. However, if it is possible to build one, the suborder representation appear like a good starting point so there could be more to that story.

The OT protocols that we mentioned should not be complicated to adapt to pSIDH given that they mostly require a DDH commutative diagram. The oblivious PRF from [BKW20] also appears like an interesting application. First, verifiability is a big issue for this primitive and the construction proposed in [BKW20] includes some zero-knowledge isogeny proof-of-knowledges which are quite expensive and not very compact in the setting of SIDH. Given that verifying computations is inherently a lot easier with pSIDH, it might prove a good match. Second, [BKM+21] have presented some attacks against the SIDH-based OPRF from [BKW20]. These attacks might be avoided with a pSIDH variant. Of course, as for the n-party key exchange, new algorithmic tools are needed before we can hope to obtain the analog of the OPRF in the setting of pSIDH and it requires some more work.

Group action. The sub-exponential quantum attack that we presented in Section 4.5 was based on the existence of a group action on the set of ideals of norm D. After a quick glance, it seems like this group action could also be cryptographically relevant and be used to instantiate the increasing list of group-action based protocols in the literature. It is not exactly clear that this new group action could be more interesting than the one based on CSIDH [CLM+18], but it is probably worth studying further to understand it better the differences between the two.

Zero-knowledge proof of suborder witness knowledge. We mentioned several time already the interest of zero-knowledge proofs of isogeny-knowledge. We know there exist somewhat practical instantiations in the setting of SIDH and CSIDH. We explained and argued that it seems complicated to do the same with ideal witnesses. The next natural question is whether we can hope to do it for the new suborder witnesses. Proving the knowledge of several endomorphisms of given norm might be feasible but making the additional verification that they generate a specific quaternion order might prove a lot more arduous. As of yet, there does not seem to be an easy way to do that.

Trapdoor mechanism from endomorphisms revelation. One of the main novelty behind our suborder witness construction is the revelation of suborders of rank 4 contained inside endomorphism rings of supersingular curves. Until our work, revealing more than one non-trivial endomorphisms has always been considered as a dangerous thing, but we conjecture with the hardness of the SOIP that it is not problematic when done carefully. It might be possible to exploit this mechanism for further applications. For instance, we can look at the trapdoor one-way function (TOWF) of the SETA scheme from [DFFdSG+21]. In this primitive, the trapdoor is some endomorphism of the public key curve. In the instantiation proposed in [DFFdSG+21], the endomorphism ring of the public key curve is typically computed during key generation, but we could imagine a situation where one participant P_1 generates a curve E (and compute its endomorphism ring along the way) before revealing a well-chosen endomorphism of E to another participant P_2. Then, P_2 could use this endomorphism to perform some protocols (for instance the SETA-TOWF) without knowing anything else on the curve E.

It seems tempting to try to build IBE from this setting. For instance, the master public key could be a curve E with the master secret key as $\mathrm{End}(E)$, identities would be isogenies from E to curves E_{id} and the corresponding secret key would be an endomorphism of E_{id} that could be used as a SETA secret key. Unfortunately, it seems hard to choose these secret keys in a way that would prevent an adversary who has access to several o them to recover enough information to generate secret keys for himself. Even though IBE appears to be out of reach from this idea, lesser primitive could still be achievable.

6.3 About efficiency

We have proven (at least heuristically) that all our new algorithms can be executed in polynomial time. However, this does not prove anything on the concrete efficiency. For instance, it would be interesting to compare pSIDH with other existing isogeny-based key exchanges. The only thing that we can claim with certainty is that pSIDH will be a lot slower than SIDH. In fact, we will rather estimate the complexity of pSIDH by comparing it with SQISign. This comparison is relevant for two reasons: we can take the same size of prime p (and measure relative efficiency by counting the number of operations over $\mathbb{F}_{p^2}$) and the bottlenecks should be the same. We elaborate on that below.

Our analysis in Section 4.5 indicates that the only security constraint on the prime p is that it needs to be big enough to prevent the exponential attacks against the endomorphism ring problem (which is the SQISign key recovery problem). Once p has been fixed, the hardness of our new SOIP depends on the value of D. The main attack against the SOIP that we introduce in Section 4.5 has quantum sub-exponential complexity in D. So we can expect the value of D to be significantly bigger than p. This gap between p and D will also induce a gap between the performances of SQISign and the performances of pSIDH. Based on empirical observations, we can predict that the bottleneck in our algorithms is going to be the same as the bottleneck in SQISign's signature: executions of the $\mathsf{IdealToIsogeny}$ sub-algorithm. The method introduced in [DFKL+20] for $\mathsf{IdealToIsogeny}$ requires to perform a number of arithmetic operations over $\mathbb{F}_{p^2}$ that is linear in the length of the isogeny to be translated. For SQISign the length is equal to $O(\log(p)) = O(\lambda)$ where λ is the security parameter. For pSIDH, the size estimates from Section 5.2 show that the length is in $O(\log(pD)) = O(\lambda^2)$. Thus, we can expect pSIDH to be asymptotically slower than SQISign by a factor $C\lambda$ (a more concrete analysis is required to oba C).

Needless to say than anyone wanting to implement concretely any of the algorithm in this work should not use the version given by Wesolowski in [Wes22] but rather one of the heuristic variant (see the algorithms in [DFKL+20] for instance).

6.4 Conclusion

We have introduced the *suborder representation*, a new way to witness membership to the language of isogenous supersingular curves. We have shown that this representation could be computed and verified in polynomial-time and we have exhibited how to evaluate efficiently isogenies using this suborder witness. In the process, we have introduced several new algorithms to solve norm equations inside new families of orders and ideals of the quaternion algebra $B_{p,\infty}$ that may be of independent interest.

We have also introduced pSIDH, a new NIKE based on the suborder representation that can be seen as a generalized version of SIDH for prime degrees. The security of this new protocol rely on the hardness of new problems: the SOIP and its decisional variants. Assuming the hardness of this problem, our new idea may lead to interesting new applications.

References

AJJS19. Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. Practical supersingular isogeny group key agreement. *IACR Cryptol. ePrint Arch.*, 2019:330, 2019.

BDF21. Jeffrey Burdges and Luca De Feo. Delay encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 302–326. Springer, 2021.

BFLS20. Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *ANTS*, 2020.

BJS14. Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *International Conference on Cryptology in India*, pages 428–442. Springer, 2014.

BKM+21. Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Antonio Sanso. Cryptanalysis of an oblivious prf from supersingular isogenies. *Cryptology ePrint Archive*, 2021.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. Csi-fish: Efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019.

BKW20. Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 520–550. Springer, 2020.

BOBN19. Paulo Barreto, Glaucio Oliveira, Waldyr Benits, and Anderson Nascimento. Supersingular isogeny oblivious transfer. In *Anais do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 99–112. SBC, 2019.

CJS14. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

CK01. Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *International conference on the theory and applications of cryptographic techniques*, pages 453–474. Springer, 2001.

CK19. Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Number-Theoretic Methods in Cryptology 2019*, 2019.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.

Cos20. Craig Costello. B-sidh: supersingular isogeny diffie-hellman using twisted torsion. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 440–463. Springer, 2020.

CSHT21. Jorge Chavez-Saab, Francisco Rodríguez Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum vdf. Cryptology ePrint Archive, Report 2021/1289, 2021. https://ia.cr/2021/1289.

DFFdSG$^+$21. Luca De Feo, Tako Boris Fouotsa, Cyprien Delpech de Saint Guilhem, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski. Séta: Supersingular encryption from torsion attacks. In *ASIACRYPT*, 2021.

DFG19. Luca De Feo and Steven D Galbraith. Seasign: Compact isogeny signatures from class group actions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 759–789. Springer, 2019.

DFKL$^+$20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 64–93. Springer, 2020.

DFMPS19. Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 248–277. Springer, 2019.

dSGOPS20. Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, Christophe Petit, and Nigel P Smart. Semi-commutative masking: A framework for isogeny-based protocols, with an application to fully secure two-round isogeny-based ot. In *International Conference on Cryptology and Network Security*, pages 235–258. Springer, 2020.

EHL$^+$18. Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 329–368. Springer International Publishing, 2018.

EHL$^+$20. Kirsten Eisenträger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. *Open Book Series*, 4(1):215–232, 2020.

FKM21. Tako Boris Fouotsa, Péter Kutas, and Simon-Philipp Merz. On the isogeny problem with torsion point information. *IACR Cryptol. ePrint Arch.*, 2021:153, 2021.

GPS17. Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In *ASIACRYPT*, 2017.

GPST16. Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 63–91. Springer, 2016.

HS09. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106. 01 2009.

JDF11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

JS14. David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In *International Workshop on Post-Quantum Cryptography*, pages 160–179. Springer, 2014.

KLPT14. David Kohel, Kristin E. Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *IACR Cryptology ePrint Archive*, 2014:505, 2014.

KMP+20. Péter Kutas, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Weak instances of sidh variants under improved torsion-point attacks. Cryptology ePrint Archive, Report 2020/633, 2020. https://eprint.iacr.org/2020/633.

KMPW21. Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Charlotte Weitkämper. One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 242–271. Springer, 2021.

Koh96. D. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkeley, 1996.

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 330–353. Springer, 2017.

Sch95. René Schoof. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux*, 7(1):219–254, 1995.

Vél71. J. Vélu. Isogénies entre courbes elliptiques. *Comptes-Rendus de l'Académie des Sciences, Série I*, 273:238–241, juillet 1971.

Voi18. John Voight. *Quaternion Algebras*. Springer Graduate Texts in Mathematics series, 2018.

Wat69. William C. Waterhouse. Abelian varieties over finite fields. *Annales Scientifiques de l'E.N.S*, 1969.

Wes22. Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *FOCS 2021-62nd Annual IEEE Symposium on Foundations of Computer Science*, 2022.

YAJ+17. Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *International Conference on Financial Cryptography and Data Security*, pages 163–181. Springer, 2017.

A Suborder witnesses for composite degree isogenies

In this section, we explain how to extend the results from Section 4 to the case of composite degree. The main obstacle is that when D is not prime, Lemma 3 does not hold anymore. In fact, the problem is already there when D is prime but it is manageable. Indeed, the formulation of Proposition 8 that we would have liked is that E_1, E_2 are D-isogeneies if and only if $\mathbb{Z} + D\mathrm{End}(E)$ is embedded inside $\mathrm{End}(E_2)$. Instead, we have to take into account the case where $\mathrm{End}(E)$ and $\mathrm{End}(E_2)$ are isomorphic as $\mathbb{Z} + D\mathcal{O} \subset \mathcal{O}$ for any quaternion order $\mathcal{O}$. This not really problematic as checking that $\mathrm{End}(E) \cong \mathrm{End}(E_2)$ is very easy. However, the problem become a lot more serious when D is composite. Let us take $\varphi_2 \circ \varphi_1 : E_0 \to E_1 \to E_2$ of degree $D_1 D_2$. Then, $\mathbb{Z} + D_1 D_2 \mathrm{End}(E_0)$ in the three endomorphism rings $\mathrm{End}(E_0), \mathrm{End}(E_1)$ and $\mathrm{End}(E_2)$. Thus, if we prove that E_0, E_2 are $D_1 D_2$ isogenous we need a way to rule out the case where E_2 is only D_1 or D_2 isogenous to E_0.

This is where the definition of **primitive** embeddings comes into play. We say that the embedding $\iota : \mathfrak{O} \hookrightarrow \mathcal{O}$ primitive if there does not exists any order $\mathcal{O}' \subsetneq \mathcal{O}$ such that $\iota(\mathfrak{O}) = \mathbb{Z} + N\mathcal{O}'$. With this definition of primitive embeddings we can state the generalization of Proposition 8.

Proposition 22. *Let $D \neq p$ be a prime number and E_1, E_2 be two supersingular curves, $\mathcal{O} \subset B_{p,\infty}$ is a maximal order isomorphic to* $\mathrm{End}(E)$*. The order $\mathbb{Z} + D\mathcal{O}$ is* ***primitively*** *embedded inside* $\mathrm{End}(E_2)$ *if and only if $(D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}}$ or $(D, E_1^p, E_2) \in \mathcal{L}_{\mathsf{isog}}$.*

Proof. For the forward direction, we need the equivalent of Lemma 3 for primitive embeddings. Thus, we are going to show that when, $\mathfrak{O} = \mathbb{Z} + D\mathfrak{O}_0$ is primitively embedded inside $\mathcal{O}$, then there exists a left integral primitive $\mathcal{O}$-ideal of norm D whose right order contains $\mathfrak{O}_0$. We can prove this by applying recursively Lemma 3 on $\mathfrak{O} = \mathbb{Z} + \ell((D/\ell)\mathfrak{O}_0)$ for any prime ℓ dividing D. At each given iteration, we will obtain an ideal of norm ℓ (and the fact that $\mathfrak{O}$ is primitively embedded rules out the case where $\mathfrak{O}_0 \subset \mathcal{O}$). In the end, multiplying all these ideals together, we obtain an ideal of norm D between $\mathcal{O}$ and a maximal ideal containing $\mathfrak{O}_0$. The final ideal is primitive as otherwise we could divide by some constant $d'|D$ and obtain that $\mathbb{Z} + (D/d')\mathfrak{O}_0$ is embedded inside $\mathcal{O}$.

For the backward direction, using the same construction as in the proof of Proposition 8, we obtain that $\mathbb{Z} + D\mathrm{End}(E)$ is embedded inside $\mathrm{End}(E_2)$. Remains to see that this embedding is primitive. Let us assume that the embedding is not primitive. Then there exists $\iota : \mathcal{O}' \hookrightarrow \mathrm{End}(E_2)$ such that the elements of $\mathbb{Z} + D\mathrm{End}(E)$ are contained inside $\mathbb{Z} + N\iota(\mathcal{O}')$. First, it is clear that N must be dividing D. If we write $\varphi : E_1 \to E_2$ for the isogeny of degree D. This isogeny can be decomposed as $\psi_N \circ \psi$ where ψ_N has degree N. By our assumption any endomorphism $\gamma = d + \varphi\alpha\hat{\varphi}$ must equal to $d + N\alpha_N$ where $\alpha_N \in \mathrm{End}(E_2)$. Thus, the action of γ on the N-torsion must be equal to the scalar multiplication by d. It is easy to see that it cannot be the case for all $\alpha \in \mathrm{End}(E)$. So there is a contradiction and this proves that the embedding is primitive.

Verification in the composite case. Now, we explain briefly how to extend the VerifSuborderProof to perform the verification when the degree D is composite. The current verification mechanism simply check that there is an embedding $\iota : \mathbb{Z} + D\mathrm{End}(E) \hookrightarrow \mathrm{End}(E_2)$. With Proposition 22, we see that we also need to check that this embedding is primitive. To do that, it suffices to check that $\iota(\mathbb{Z} + D\mathrm{End}(E)) \neq \mathbb{Z} + N\mathfrak{O}$ for some order $\mathfrak{O} \subset \mathrm{End}(E_2)$ and $N|D$. Since $\mathfrak{O} \cong \mathbb{Z} + (D/N)\mathrm{End}(E)$, it suffices to find one endomorphism $\beta = d + \varphi \circ \alpha \circ \hat{\varphi}$ and prove that $d' + (\beta - d)/N$ is not an endomorphism of E_2 to prove that $\iota(\mathbb{Z} + D\mathrm{End}(E)) \neq \mathbb{Z} + N\mathfrak{O}$ for any N of $\mathfrak{O}$. If the norm of $d' + (\beta - d)/N$ is powersmooth and coprime with N, $G_N = \ker(d' + (\beta - d)/N)$ and E_2/G_N can be computed efficiently. Thus, the additional verification mechanism work as follows: for every prime N dividing D, use SmoothGen to compute a generating family $\theta_1, \ldots, \theta_n$ of norm coprime with N of $\mathbb{Z} + (D/N)\mathrm{End}(E)$, express each θ_i as $d' + (\beta_i - d)/N$ where $\beta_i \in \mathbb{Z} + D\mathrm{End}(E)$ and compute $G_{N,i} = \ker d' + (\iota(\beta_i) - d)/N$. If there exists one N such that $j(E_2/G_{i,N}) = j(E_2)$ for all $1 \leq i \leq n$, the verification fails.

B Faster verification with computational soundness.

Proposition 10 is conditioned on the size of the value $M = p^k - 1$. As explained in Section 5.4, $\mathsf{CheckTrace}_M$ works by evaluating the endomorphisms in input over the M-torsion. Given the big bound on M (see Lemma 6), the field of definition of the M-torsion might be quite big in practice. To speed-up the computation it might be possible to take a value of M below the bound of Proposition 10. In that case, we would obtain a proof system with computational soundness. The underlying hard problem would be the following.

Problem 4. Let M be some integer. Given a maximal order $\mathcal{O} \subset B_{p,\infty}$ and an integer D. The problem is to find E and $\varphi_1, \varphi_2, \ldots, \varphi_n \in \mathrm{End}(E)$ such that $\mathsf{CheckTrace}_M(E, \varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n) = 1$ with $\theta_1, \ldots, \theta_n = \mathsf{SmoothGen}_{\ell^\bullet}(\mathcal{O}, D)$ but the order generated by $\varphi_1, \ldots, \varphi_n$ is not isomorphic to $\mathbb{Z} + D\mathcal{O}$.

Analysis of Problem 4. First, we would like to highlight that the hardness of Problem 4 is a type of assumption quite unusual in isogeny-based cryptography. Contrary to Problem 1 (which is new but remains related to rather classical problem given the equivalence with Problem 2), the hardness of Problem 4 is related to the hardness of solving some set of quadratic equations.

Problem 4 is difficult to analyze. Indeed, in Lemma 6 we give an upper bound on the value of M for which there are no solutions to the problem. However, it is not clear what is the optimal such value. It may be that for some values asymptotically smaller than the proven bound, there is already no possible solutions. However, since we were unable to prove that, the conservative approach is to assume that there may be some solutions. In that case, finding a solution amounts to finding a curve E a and endomorphisms of $\mathrm{End}(E)$ satisfying a bunch of norm equations in $\mathbb{Z}$ and trace equations $\bmod M$. These equations can be seen as quadratic equations that can be solved mod M, but since we also need equality of the norms over $\mathbb{Z}$, it is not clear whether there are solutions and if they are easy to find. The usual tools used to solve equations over quaternion orders (for instance in [KLPT14,DFKL$^+$20]) are not sufficient to address our problem.

Let us look at the simple example where $n = 2$. Then, the order is $\mathfrak{O} = \mathsf{Order}(\theta_1, \theta_2) = \langle 1, \theta_1, \theta_2, \theta_1\theta_2 \rangle$. The goal is to find θ_1, θ_2 with a precise constraint on their norm, and a constraint mod M for the three traces $\mathrm{tr}(\theta_1), \mathrm{tr}(\theta_2), \mathrm{tr}(\theta_1\theta_2)$. While it is easy to find θ_1 and θ_2 with the correct norm and trace, it seems difficult to ensure the additional constraint on $\mathrm{tr}(\theta_1\theta_2)$. Let us look at that constraint when $\theta_1 = a + ib + jc + kd$ and $\theta_2 = e + if + jg + kh$, then $\mathrm{tr}(\theta_1\theta_2) = ae - (qbf + p(cg + qdh))$. Thus, the problem is: given $n_1, n_2, t_1, t_2, t_3, M$ find a, b, c, d, e, f, g, h such that $a^2 + qb^2 + pc^2 + qpd^2 = n_1, e^2 + qf^2 + pg^2 + pqh^2 = n_2$ and $2a = t_1 \mod M$, $2e = t_2 \mod M$ and $ae - (qbf + p(cg + qdh)) = t_3 \mod M$. This appears to be hard when M is too big for the equation $\mod M$ to be satisfied at random. In practice, as explained in Section 5.3, we take $n = 3$ and $\mathfrak{O}$ has an even more complicated structure which only increases the number of equations to be verified, as highlighted in Lemma 5.

Remark 4. Additionally, we highlight that progress toward solving the kind of equations above, would probably allow us to devise an algorithm $\mathsf{SmoothGen}$ finding solutions of smaller norm, which would make Problem 4 more difficult.