## A New Isogeny Representation and Applications to Cryptography

Antonin Leroux

<sup>1</sup> DGA <sup>2</sup> LIX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris <sup>3</sup> INRIA antonin.leroux@polytechnique.org

Abstract. This paper focuses on isogenv representations, defined as ways to evaluate isogenies and verify 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  $E_2$ ). The tasks of evaluating and verifying isogenies are fundamental for isogeny-based cryptography. 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. This new representation appears 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. Studying new assumption appears to be particularly crucial in the light of the recent attacks against isogeny-based cryptography. In order to manipulate efficiently the suborder representation, we 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 primitives achievable from isogenies is expanding, new problems are arising. The problem of proving the knowledge of an isogeny between two elliptic curves is one that appears more and more central in isogeny-based cryptography. It has applications in validation of SIDH public keys [JDF11,GPST16,FP22,UXT<sup>+</sup>22], digital signatures [YAJ<sup>+</sup>17,DFG19,BKV19,JS14], VDFs [DFMPS19,CSRHT22], 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 [BdFLS20]). 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 above 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.

This story begins with the KLPT algorithm from Kohel, Lauter, Petit and Tignol [KLPT14] to solve the quaternion analog of the isogeny path problem. In [EHL<sup>+</sup>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 original motivation behind the study of the Deuring correspondence in [KLPT14,EHL<sup>+</sup>18] is cryptanalysis. The tools developped toward that end have only recently started to be used constructively. The main building blocks of the signature scheme from Galbraith, Petit and Silva [GPS17] and the later generalization of SQISign by De Feo, Kohel, Leroux, Petit and Wesolowski [DFKL<sup>+</sup>20] are variants of the KLPT algorithm from Kohel et al. The key generation of the encryption scheme Séta [DFFdSG<sup>+</sup>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 that are manipulated as ideals. In  $[DFKL^{+}20]$ , the authors argue that using a secret key of prime degree provides better efficiency for the same level of security. The motivation of our paper is to provide a new way of representing isogenies of prime degree that can open up some interesting cryptographic applications. This appears particularly interesting in light of the recent attacks [CD22,MM22,Rob22] that break SIDH and Séta. These attacks are targetting smooth degree secret isogenies and we will see how these attacks fail to break the assumption based on the new representation we introduce.

A first small contribution of this work is to introduce a new terminology of *isogeny representation*, hoping that it can help formalizing some results about isogenies by providing a common framework on the different methods of isogeny computations.

Our main contribution is a new generic isogeny representation that we call a *suborder representation*. This representation is constituted of the endomorphism ring of the domain and several endomorphisms of the isogeny's codomain. We present heuristic polynomial-time algorithms to compute and verify the suborder representation when the degree D is prime. The case of composite D is more complicated and does not seem to be more interesting for cryptography, so it is treated in the full version. 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 implies that the knowledge of a suborder of rank 4 is not always 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 where we also prove that the SOIP is equivalent to some instances of the Torsion to Ideal Problem (TIP), a new problem that can be seen as a generalization of the CSSI, the key recovery problem of SIDH [JDF11]. Because we consider an instance of this problem where the degree of the secret isogeny is prime, the recent attacks on SIDH does not seem to apply directly.

Our new isogeny representation requires to solve norm equations inside a new family of quaternion orders and ideals and we develop the necessary heuristic 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<sup>+</sup>20,DFFdSG<sup>+</sup>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. The efficiency of pSIDH is not likely to be competitive and it needs to be considered as a first step toward more involved applications.

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  $\mathcal{L}_{isog}$ , the language of isogenous curves, and show that it is in NP using the *ideal representation* of isogenies. In Section 4, we introduce our new *suborder representation*. We provide some algorithms to compute and verify these representations, and analyze how they differ from *ideal representations*. 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, in Section 6, we introduce a new isogeny-based NIKE scheme based on the suborder representation.

## 2 Background material

The set of prime numbers is denoted  $\mathbb{P}$ . For a prime  $\ell \in \mathbb{P}$ , we define  $\ell^{\bullet} = \{\ell^k | k \in \mathbb{N}\}.$ 

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 Notations and simplifications.

Throughout this work, p > 3 is a prime number and  $B_{p,\infty}$  is the unique quaternion algebra ramified at p and  $\infty$ . For ease of exposition, we use a simplified terminology and conventions that we will keep during the entire paper. We introduce them below.

When talking about elliptic curves and isogenies, we always consider isomorphism classes of curves and isogenies respectively. This means that when needed (in an algorithm for instance) we represent curves by their *j*-invariant that we write j(E) for a curve E (implicitly deriving a full equation of a canonical representative of the isomorphism class if needed). For an isogeny  $\varphi$ , we implicitly adapt whatever isogeny representation we use to this convention, so we pre and post-compose with the relevant isomorphisms to have an isogeny defined on the canonical representatives of the domain and codomain. For an isogeny of domain E and kernel G, we note the codomain class as E/G.

Any four dimensional lattice  $\Lambda$  of  $B_{p,\infty}$  is given by 16 coefficients in  $\mathbb{Q}$  corresponding to the decomposition over a basis of  $B_{p,\infty}$  of a basis of  $\Lambda$ . This is what we call the *representation* of an order or an ideal and is what is used when a computation is required. For an order  $\mathcal{O} \in B_{p,\infty}$ , an  $\mathcal{O}$ -ideal of  $B_{p,\infty}$  will always be a left integral  $\mathcal{O}$ -ideal of norm coprime with p unless said otherwise. An isogeny will always be a cyclic separable isogeny.

Quaternion notations. We list below a few notations that are used throughout the paper. We refer the reader to the full version for more details. The *left and* right orders of an ideal are written  $\mathcal{O}_L(\cdot), \mathcal{O}_R(\cdot)$ . Two ideals are equivalent if  $J = I\beta$  for  $\beta \neq 0$  and we say that I, J are equivalent. The norm of an ideal I is denoted by n(I).

#### 2.2 The Deuring correspondence

The Deuring correspondence is an equivalence of categories between isogenies of supersingular elliptic curves and the left ideals over maximal order  $\mathcal{O}$  of  $\mathcal{B}_{p,\infty}$ , inducing a bijection between conjugacy classes of supersingular *j*-invariants and maximal orders (up to equivalence) [Koh96]. This bijection is explicitly constructed as  $E \to \text{End}(E)$ . Hence, given a supersingular curve  $E_0$  with endomorphism ring  $\mathcal{O}_0$ , the pair  $(E_1, \varphi)$ , 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 (obtained by considering kernel ideals [Wat69]) with  $\mathcal{O}_R(I) \cong \text{End}(E_1)$ .

**Definition 1.** Let  $I \subset \operatorname{End}(E_0)$  be an integral ideal, we define  $E_0[I] = \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0 \text{ for all } \alpha \in I\}$  and the isogeny corresponding to I is  $\varphi_I : E_0 \to E_0/E_0[I]$ . Conversely, given an isogeny  $\varphi$  with domain  $E_0$ , the corresponding ideal is  $I_{\varphi} = \{\alpha \in \mathcal{O}_0 : \alpha(P) = 0 \text{ for all } P \in \ker(\varphi)\}.$ 

Quaternion orders admit what we call a Gorenstein decomposition. Any quaternion order  $\mathcal{O}$  can be expressed as  $\mathbb{Z} + f\mathcal{O}'$ , where f is the Brandt Invariant and  $\mathcal{O}'$  is the Gorenstein closure. We will try to understand the Brandt Invariant through the Deuring correspondence in Section 4.1

Supersingular <i>j</i> -invariants over $\mathbb{F}_{p^2}$	Maximal orders in $B_{p,\infty}$
j(E) (up to Galois conjugacy)	$\mathcal{O} \cong \operatorname{End}(E)$ (up to isomorphia)
$(E_1, \varphi)$ with $\varphi: E \to E_1$	$I_{\varphi}$ integral left $\mathcal{O}$ -ideal and right $\mathcal{O}_1$ -ideal
$\theta \in \operatorname{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$n(I_{arphi})$
$\hat{arphi}$	$\overline{I_{\varphi}}$
Supersingular <i>j</i> -invariants over $\mathbb{F}_{n^2}$	Ideal class set of a maximal order $\mathcal{O}$

	1		
Table 1. The Dep	uring correspondence	a summary from	1 [DFKL <sup>+</sup> 20]

# 3 The language of isogenous curves and the ideal representation

Let us fix a prime p. We will study  $\mathcal{L}_{isog}$ , the language of isogenous supersingular curves in characteristic p.

We write  $S_p$  as the set of isomorphism classes of supersingular elliptic curves in characteristic p, and  $lsog_D$  the set of cyclic D-isogenies between curves of  $S_p$ .

**Definition 2.** The language of isogenous supersingular curves is

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

An isogeny representation is a string  $s_{\varphi}$  associated to an isogeny  $\varphi: E_1 \to E_2$ of degree D. This string can be used as input to two algorithms: one that can verify that the element  $D, E_1, E_2$  is in  $\mathcal{L}_{isog}$  and one that can compute  $\varphi(P)$  for some point  $P \in E_1$ .

We call the former a *verification algorithm* and the latter an *evaluation algorithm*. We can regroup isogeny representations in *families of representations* by looking at the associated verification and evaluation algorithms. Thus, to a family XX of representations we associate two algorithms XXVerification and XXEvaluation.

Standard isogeny representation. The default isogeny representation of  $\varphi \in \mathsf{Isog}_D$  is made of the rational maps  $f_1, f_2 \in \mathbb{F}_{p^m}(x, y)$  such that the image under  $\varphi$  of any point (x, y) of the domain is  $(f_1(x, y), f_2(x, y))$  and  $\mathbb{F}_{p^m}$  is the field of definition of ker  $\varphi$ . The degree of the polynomials used in  $f_1, f_2$  are in  $O(\mathsf{poly}(D))$ . Since any isogeny of degree  $D_1D_2$  is the composition of a  $D_1$ -isogeny and a  $D_2$ -isogeny, decomposing  $\varphi$  in smaller isogenies allows us to get a default representation of size  $O(\mathsf{poly}(\log(pD)))$  when D has smoothness bound in  $O(\mathsf{poly}(\log(pD)))$ . When not said otherwise, this default representation is used for the computation of isogenies of smooth degree. It is also standard in the literature to use a generator of ker  $\varphi$  as a representation, we call that the kernel representation. This representation can be used to compute the default isogeny representation with the Vélu Formulae [Vél71]. The computational cost is also  $O(\mathsf{poly}(\log(pD)))$  when D has smoothness bound in  $O(\mathsf{poly}(\log(pD)))$ .

#### 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. We use the default representation of isogenies.

- 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  $\mathcal{O}_R(I) = \mathcal{O}_2$ .
- $\mathsf{KLPT}_{\ell}$ : takes a left  $\mathcal{O}$ -ideal I and outputs  $J \sim I$  of norm  $\ell^e$ .
- $\mathsf{KLPT}_{\mathsf{PS}}$ : takes a left  $\mathcal{O}\text{-}\mathrm{ideal}\;I$  and outputs  $J\sim I$  of powersmooth norm.
- IdealTolsogeny<sub>T</sub>: takes a left  $\mathcal{O}$ -ideal I of norm T and computes  $\varphi_I$ .
- IsogenyToldeal<sub>T</sub>: takes an isogeny  $\varphi : E \to E'$  of degree T, a maximal order  $\mathcal{O} \cong \operatorname{End}(E)$  and computes  $I_{\varphi}$ .

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

**Proposition 1.** ConnectingIdeal terminates in  $O(\text{poly}(\log(p) + C)))$  where C is the size of the representation of  $\mathcal{O}_1, \mathcal{O}_2$ .

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

**Proposition 3.** Assuming GRH, KLPT<sub>PS</sub> terminates in expected  $O(\operatorname{poly}(\log(pD) + C))$  where D is the norm of the input and outputs an ideal of norm in  $O(\operatorname{poly}(p))$  with smoothness bound in  $O(\operatorname{poly}(\log(p)))$  and the representation of  $\mathcal{O}$  has C bits.

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

**Proposition 5.** For any number T = O(poly(p)) with smoothness bound in  $O(poly(\log(p)))$ ,  $IdealTolsogeny_T$  terminates in expected  $O(poly(\log(p)+C))$  and the output has size  $O(poly(\log(p)))$  and the representation of  $\mathcal{O}$  has C bits.

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

We define the *ideal representation* for the isogeny  $\varphi$  as a representation of the associated kernel ideal  $I_{\varphi}$ . Note that this implies the knowledge of the endomorphism rings of both  $E_1$  and  $E_2$ . With Lemma 1 below, we prove that the ideal representation can be compact.

**Lemma 1.** Any ideal I of norm D is isomorphic to an ideal J with a representation of size  $O(\log(pD))$ .

*Proof.* It was shown in [EHL<sup>+</sup>18] that any maximal order is isomorphic to an order with a representation of size  $O(\log(p))$ . Let us write  $\mathcal{O}$  for the maximal order with the small representation isomorphic to  $\mathcal{O}_L(I)$ . We can write J for the image of I under the isomorphism between  $\mathcal{O}_L(I)$  and  $\mathcal{O}$ . Since  $D\mathcal{O} \subset J$  (this is true for any cyclic  $\mathcal{O}$ -ideal of norm D), we see that we can choose a basis of J inside the basis of  $\mathcal{O}$  with coefficients of size  $O(\log(D))$ . Thus, there exists a representation of J of size  $O(\log(p) + \log(D))$ .

For simplicity, we assume in the rest of this work, that ideals and orders are given with a compact representation as in Lemma 1.

We now present IdealVerification as Algorithm 1. It is a verification algorithm that takes a triple  $x = (D, E_1, E_2)$  and an ideal I and decides if  $x \in \mathcal{L}_{isog}$ . The idea is to compute the curves whose endomorphism ring are isomorphic to the left and right order of I. If the curves obtained in this way are isomorphic to  $E_1, E_2$ , the verification passes. To do that, we will 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 codomain of the corresponding isogeny with IdealTolsogeny. Since these isogenies have smooth norm, they can be efficiently computed.

## **Algorithm 1** IdealVerification(x, I)

**Input:**  $x \in \mathbb{N} \times S_p^2$  and I an ideal of  $B_{p,\infty}$ . **Output:** A bit indicating if  $x \in \mathcal{L}_{isog}$ . 1: Parse x as  $D, E_1, E_2$  and take  $\ell$  a small prime. 2: Compute n(I) and  $\mathcal{O}_L(I), \mathcal{O}_R(I)$ . 3: if  $n(I) \neq D$  or  $I \not\subset \mathcal{O}_L(I)$  then 4: Return 0. 5: **end if** Take a curve  $E_0$  defined over  $\mathbb{F}_p$  with  $\operatorname{End}(E_0) \cong \mathcal{O}_0$  and compute  $I_1 =$ 6: ConnectingIdeal( $\mathcal{O}_0, \mathcal{O}_L(I)$ ),  $I_2 = I_1 \cdot I$ . 7: for  $i \in [1, 2]$  do Compute  $J_i = \mathsf{KLPT}_{\ell^{\bullet}}(I_i)$  and  $\varphi_i : E_0 \to E'_i = \mathsf{IdealTolsogeny}_{\ell^{\bullet}}(\mathsf{E}_0,\mathsf{J}_i)$ . 8: 9: end for 10: if  $(j(E'_1), j(E'_2)) \notin \{(j(E_1), j(E_2)), (j(E_1)^p, j(E_2)^p)\}$  then 11: Return 0. 12: end if 13: return 1.

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

Conversely, for  $(D, E_1, E_2) \in \mathbb{N} \times S_p^2$ , if there exists an ideal I such that  $\mathsf{IdealVerification}((D, E_1, E_2), I) = 1$  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  $I_{\varphi}$ , we have  $n(I_{\varphi}) = D$  and  $I_{\varphi} \subset \mathcal{O}_L(I_{\varphi})$  so the first check passes. Then, the codomain of the two  $\varphi_{I_i}$  have endomorphism ring isomorphic to  $\mathcal{O}_R(I_i)$  so they might be either both  $E_i$  or both  $E_i^p$  (since  $I_2 = I_1 I_{\varphi}$ , 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  $\mathsf{IdealVerification}((D, E_1, E_2), I) = 1$ , 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  $E^p_1, E^p_2$ . Since  $\varphi : E^p_1 \to E^p_2$  of degree D imply the existence of  $\varphi^p : E_1 \to E_2$  of degree D, in both cases we have that  $(D, E_1, E_2) \in \mathcal{L}_{isog}$ .

**Proposition 6.** Under GRH, IdealVerification terminates in expected  $O(\text{poly}(\log(pD) + C))$  where C is the bit size of the representation of I.

Proposition 6 follows directly from Propositions 1, 2 and 5.

Isogeny Evaluation from ideals. It is also possible to evaluate an isogeny from its ideal representation. In the full version of the paper, we present an algorithm IdealEvaluation solving that task. The main idea is to apply KLPT and IdealTolsogeny to find an equivalent isogeny of powersmooth degree and making use of it to perform the computation. Note that an algorithm very similar to IdealEvaluation can be found in [FKMT22].

## 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 in the full version that they do not appear more interesting. We will briefly explain how to extend the suborder representation to composite degrees in the full version as well. 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  $E_1, E_2$  or  $E_1, E_2^p$  are *D*-isogenous and works only when  $\operatorname{End}(E_1) \cong \operatorname{End}(E_2)$ . Thus, we consider the alternate language  $\mathcal{L}_{p-isog}$  defined as follows:

$$\mathcal{L}_{\mathsf{p}-\mathsf{isog}} = \{ (D, E_1, E_2) \in \mathbb{P} \times \mathcal{S}_p^2 | E_1 \not\cong E_2, E_2^p \text{ and } (D, E_1, E_2) \in \mathcal{L}_{\mathsf{isog}} \text{ or } (D, E_1, E_2^p) \in \mathcal{L}_{\mathsf{isog}} \}$$

In Section 4.1, we introduce the mathematical results underlying our new method. The method to extract the new representation from the ideal representation is the goal of Section 4.2. Then, in Section 4.3, we explain how to perform a heuristic polynomial-time verification of this new witness.

#### 4.1 Brandt Invariant and relation with isogenies

The goal of this section is to prove Proposition 7 that links the *Brandt invariant* of some orders with isogenies through the Deuring correspondence.

**Proposition 7.** Let  $D \neq p$  be a prime number and  $E_1, E_2$  be two supersingular curves over  $\mathbb{F}_{p^2}$ ,  $\mathcal{O}_1 \subset B_{p,\infty}$  is a maximal order isomorphic to  $\operatorname{End}(E_1)$ . The order  $\mathbb{Z} + D\mathcal{O}_1$  is embedded inside  $\operatorname{End}(E_2)$  if and only if either  $j(E_2) \in \{j(E_1), j(E_1)^p\}$  or  $(D, E_1, E_2) \in \mathcal{L}_{p-isog}$ .

We will prove the backward direction of Proposition 7 with a simple argument using orders and ideals, but it is worth noting that the concrete embedding can be obtained with the map  $\alpha_0 \mapsto [d] + \varphi \circ \alpha_0 \circ \hat{\varphi}$  between  $\operatorname{End}(E_1)$  and  $\operatorname{End}(E_2)$ when there exists  $\varphi : E_1 \to E_2$  of degree D. This is not the first appearance of this map that was introduced by Waterhouse [Wat69, Section 3.1]. It is also at the heart of the attacks [Pet17,KMP<sup>+</sup>20] on the SIDH key exchange and underlies the decryption process of the Séta encryption scheme [DFFdSG<sup>+</sup>21]. In the proof of Proposition 7. The forward direction is more subtle and we use the preliminary Lemma 3.

**Lemma 3.** Let Let D be prime number different from p and  $\mathfrak{O} \subset B_{p,\infty}$  be a quaternion order such that  $\mathfrak{O} = \mathbb{Z} + D\mathfrak{O}_0$  for another order  $\mathfrak{O}_0 \subset B_{p,\infty}$ . When  $\mathfrak{O}$  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  $\mathfrak{D}_0$  is not contained in  $\mathcal{O}$ . We set  $I = \{x \in \mathcal{O}, x\mathfrak{D}_0 \subset \mathcal{O}\}$ . 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  $\mathfrak{D}_0 \notin \mathcal{O}$ . Then, with  $D\mathfrak{D}_0 \subset \mathcal{O}$  we have  $Dx\mathfrak{D}_0 = xD\mathfrak{D}_0 \subset \mathcal{O}$  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  $x_0 \in \mathfrak{D}_0$  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  $\mathfrak{D}_0$  is contained in  $\mathcal{O}_R(I)$ . This concludes the proof.

Proof. (Proposition 7) For the forward direction, let us take a maximal order  $\mathcal{O}_2 \cong \operatorname{End}(E_2)$  such that  $\mathbb{Z} + D\mathcal{O}_1 \subset \mathcal{O}_2$  (which is possible since  $\mathbb{Z} + D\mathcal{O}_1$  is embedded inside  $\operatorname{End}(E_2)$ ). Then, we apply Lemma 3 to  $\mathfrak{O}_0 = \mathcal{O}_1$ , and  $\mathcal{O} = \mathcal{O}_2$ , we obtain that either  $\mathcal{O}_2$  contains  $\mathcal{O}_1$  (in which case  $\mathcal{O}_2 = \mathcal{O}_1$  since  $\mathcal{O}_1$  is maximal) and so we have  $\operatorname{End}(E_1) \cong \operatorname{End}(E_2) \Rightarrow j(E_2) \in \{j(E_1), j(E_1)^p\}$ , or there must be an  $\mathcal{O}_2$ -integral ideal of norm D whose right order contains  $\mathcal{O}_1$ . Once again, since  $\mathcal{O}_1$  is maximal, we have in fact equality and so we have an ideal of norm D whose right order is  $\mathcal{O}_2$  and left order is  $\mathcal{O}_1$ . By the Deuring correspondence, this means that  $(D, E_1, E_2) \in \mathcal{L}_{p-isog}$ .

For the backward direction, let us consider the ideal I corresponding to the Disogeny  $\varphi : E_1 \to E_2$  (w.l.o.g, we can assume that  $D, E_1, E_2 \in \mathcal{L}_{isog}$ ). This ideal has norm D. Since  $\mathcal{O}_L(I)$  is maximal, the local order  $\mathcal{O}_L(I) \otimes \mathbb{Z}_D$  is a principal ideal domain (see [Voi18, Chapter 23]) and so the ideal I is locally principal. This proves that we can write  $I = \mathcal{O}_L(I)\alpha + \mathcal{O}_L(I)D$  for some element  $\alpha \in \mathcal{O}_L(I)$ and so  $D\mathcal{O}_L(I) \subset I \subset \mathcal{O}_R(I)$ . Thus, we obtain that  $\mathbb{Z} + D\mathcal{O}_L(I) \subset \mathcal{O}_R(I)$ , and the proof is concluded by  $\mathcal{O}_L(I) \cong \operatorname{End}(E_1)$  and  $\mathcal{O}_R(I) \cong \operatorname{End}(E_2)$ .

## 4.2 Deriving the suborder representation from the ideal representation

Proposition 7 suggests that the embedding  $\mathbb{Z} + D\operatorname{End}(E_1) \hookrightarrow \operatorname{End}(E_2)$  can be used to prove the existence of an isogeny of degree D between  $E_1$  and  $E_2$ . The goal of this section is to introduce an algorithm IdealToSuborder that takes a maximal order  $\mathcal{O} \cong \operatorname{End}(E_1)$  and an  $\mathcal{O}$ -ideal I of norm D and outputs a suborder representation for  $\varphi$  made of  $\mathcal{O}$  and the embedding  $\mathbb{Z} + D\mathcal{O} \hookrightarrow \operatorname{End}(E_2)$ . By a representation of the embedding, we actually mean the embeddings of a generating family for  $\mathbb{Z} + D\mathcal{O}$  (see Definition 3 below). We give the full definition for a suborder representation as Definition 4.

**Definition 3.** A generating family  $\theta_1, \dots, \theta_n$  for an order  $\mathcal{O}$  is a set of elements in  $\mathcal{O}$  such that any element  $\rho \in \mathcal{O}$  can be written as a linear combination of 1 and  $\prod_{i \in \mathcal{I}} \theta_i$  for all  $\mathcal{I} \subset \{1, \dots, n\}$ . In that case, we write  $\mathcal{O} = \mathsf{Order}(\theta_1, \dots, \theta_n)$ .

**Definition 4.** Let  $\varphi : E_1 \to E_2$  be an isogeny of degree D. A suborder representation  $\pi_{\varphi}$  for  $\varphi$  is made of an order  $\mathcal{O} \cong \text{End}(E_1)$  and of the default representations  $s_1, \ldots, s_n$  of n endomorphisms of  $E_2$  corresponding to a generating family of  $\mathbb{Z} + D\mathcal{O}$ .

Our algorithm IdealToSuborder (Algorithm 2) is built upon a SmoothGen<sub> $\mathcal{N}$ </sub> sub-algorithm that we will present in Section 5.3. This algorithm computes a generating family  $\theta_1, \ldots, \theta_n \in B_{p,\infty}$  for the order  $\mathbb{Z} + D\mathcal{O}$  on input  $D, \mathcal{O}$  where each  $\theta_i$  has norm in  $\mathcal{N}$ . For Proposition 8 and Proposition 10, we are going to assume several things about this SmoothGen algorithm. We summarize them in Assumption 1.

**Assumption 1** Let  $\mathcal{N} \subset \mathbb{N}$  be either  $\ell^{\bullet}$  for some prime  $\ell = O(\operatorname{poly}(\log(pD)))$ , or the set of divisors of T for some integer  $T > p^{7/2}D^6$  of size  $O(\operatorname{poly}(pD))$  and smoothness bound  $O(\operatorname{poly}(\log(pD)))$ . On input  $\mathcal{O}, D$ , the algorithm  $\operatorname{SmoothGen}_{\mathcal{N}}$ is deterministic, correct and terminates in  $O(\operatorname{poly}(\log(pD) + C))$  where  $\mathcal{O}$  is represented by C bits. It outputs n = O(1) quaternion elements whose norms are contained in  $\mathcal{N}$  for all  $1 \leq i \leq n$ .

*Remark 1.* We hide several heuristics and a conjecture 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  $\ell$  has been fixed and we write  $\ell^{\bullet}$  for the set  $\{\ell^e, e \in \mathbb{N}\}$ .

#### **Algorithm 2** IdealToSuborder(I)

**Input:** I an integral ideal of maximal orders inside  $B_{p,\infty}$  of norm D. **Output:** Endomorphisms  $\varphi_i : E_2 \to E_2$  such that  $\iota : \operatorname{End}(E_2) \xrightarrow{\sim} \mathcal{O}_R(I)$  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  $\theta_1, \ldots, \theta_n = \operatorname{SmoothGen}_{\ell^{\bullet}}(\mathcal{O}, D)$ . 3: for  $i \in [1, n]$  do 4: Compute  $\varphi_i : E_2 \to E_2 = \operatorname{IdealTolsogeny}_{\ell^{\bullet}}(\mathcal{O}'\theta_i)$ . 5: Compute the default representation  $s_i$  of  $\varphi_i$ . 6: end for 7: Choose  $\mathcal{O}_c$ , a maximal order isomorphic to  $\mathcal{O}$  of small representation. 8: return  $\pi = \mathcal{O}_c, (s_i)_{1 \leq i \leq n}$ .

**Proposition 8.** Under Assumption 1 and GRH, IdealToSuborder is correct and terminates in  $O(\text{poly}(\log(pD) + C))$  where C is the bitsize of I and the output has size  $O(\text{poly}(\log(pD)))$ .

*Proof.* Correctness follows from the correctness of IdealTolsogeny and Smooth-Gen. The left and right orders of I have representation of size smaller than C, and so termination follows from GRH, Assumption 1 and Proposition 5 (with n = O(1)). The degree and smoothness bound of all the deg  $\varphi_i$  is given by Assumption 1 and this implies that the default isogeny representation has size  $O(\operatorname{poly}(\log(pD)))$  as we explained in the beginning of Section 3. An  $\mathcal{O}_c$  with representation of size  $O(\log(p))$  can be found and this concludes the proof.

#### 4.3 Verification of the suborder representation

This section focuses on the verification of the representation computed with IdealToSuborder. From Proposition 7, we know that it suffices to convince the verifier that  $\mathbb{Z} + D\text{End}(E_1)$  is embedded inside  $\text{End}(E_2)$  and  $\text{End}(E_1) \ncong \text{End}(E_2)$ . 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  $\varphi_1, \ldots, \varphi_n$ . With Lemma 4, we show that it suffices to check some traces and norms of endomorphisms computed from the  $(\varphi_i)_{1 \le i \le n}$ . Due to the lack of space, the proof of Lemma 4 can be found in the full version.

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

As Lemma 4 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 19).

Lemma 5 below gives a bound above which equality will hold over  $\mathbb{Z}$  if it holds mod M. In the full version of the paper, we also explore the option of choosing a

value of M below the bound of Lemma 5, producing a tradeoff between efficiency and soundness.

**Lemma 5.** Given any  $\theta \in \text{End}(E_1)$ , if  $\text{tr}(\theta) = t \mod M$  for  $M > 4\sqrt{n(\theta)}$  and  $|t| \leq M/2$ , then  $\text{tr}(\theta) = t$ .

*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 tr:  $(x, y, z, w) \mapsto 2x$ , we can easily verify that tr $(\theta)^2 < 4n(\theta)$ . This gives a bound of  $2\sqrt{n(\theta)}$  on the absolute value of tr $(\theta)$ . The result follows.

**Algorithm 3** SuborderVerification<sub>M</sub> $(x, \pi)$ 

**Input:**  $M \in \mathbb{N}, x \in \mathbb{P} \times S_p^2$  and  $\pi$  a suborder representation. **Output:** A bit indicating if  $x \in \mathcal{L}_{p-isog}$ . 1: Parse x as  $D, E_1, E_2$  and  $\pi = \mathcal{O}, (s_i)_{1 \le i \le n}$ . 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  $J = \mathsf{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$  and  $L = \mathsf{KLPT}_{\ell^{\bullet}}(J)$ . 7: Compute  $\psi: E_0 \to E'_1 = \mathsf{IdealTolsogeny}_{\ell^{\bullet}}(L)$ . 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 Parse  $s_i$  as the default representation of an isogeny of degree  $n(\theta_i) \in \ell^{\bullet}$  and 12: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 CheckTrace<sub>M</sub>( $\varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n, E_2$ ).

**Proposition 9.** If  $M > \max_{1 \le j \le n} 2\sqrt{n(\theta_j)^n}$ , then for  $x \in \mathbb{P} \times S_p^2$ , there exists a suborder representation  $\pi$  such that  $\operatorname{VerifSuborder}[\operatorname{Proof}_M(x,\pi) = 1$  if and only if  $x \in \mathcal{L}_{p-isog}$ .

Proof. Assume that there exists a representation  $\pi$  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  $\operatorname{End}(E_1) \cong \mathcal{O}$ . Finally, the verification is Step 13 proves that the  $\varphi_i$  are endomorphisms of  $E_2$ . Then, if  $\operatorname{CheckTrace}_M(\varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n, E_2) = 1$ , the correctness of  $\operatorname{Smooth-}$ Gen, CheckTrace, Lemmas 4 and 5 imply that  $\mathbb{Z} + D\mathcal{O}$  is embedded inside  $\operatorname{End}(E_2)$ and Proposition 7 proves that  $x \in \mathcal{L}_{p-isog}$ . Now let us take  $(D, E_1, E_2) \in \mathcal{L}_{p-isog}$ . By definition there exists an ideal I of norm D and  $\mathcal{O}_L(I) \cong \operatorname{End}(E_1)$ ,  $\mathcal{O}_R(I) \cong \operatorname{End}(E_2)$ . We are going to show that if  $\pi = \operatorname{IdealToSuborder}(I)$ , then we have  $\operatorname{SuborderVerification}_M(x, \pi) = 1$ . 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  $\mathcal{O}_L(I) \cong \operatorname{End}(E_1)$ . Then, by the correctness of IdealToSuborder showed in Proposition 8, 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 9, we see that we can take  $M = #E(\mathbb{F}_{p^m})$  for the smallest  $m \in \mathbb{N}$  such that M is bigger than the bound in Proposition 9. We refer to Proposition 19 for correctness and complexity of the CheckTrace algorithm.

**Proposition 10.** Let m, M be as defined above. Under GRH and Assumption 1, SuborderVerification<sub>M</sub> terminates in probabilistic O(poly(log(p) + log(D))).

*Proof.* Since m = O(poly(log(pD))) by Proposition 9 and Assumption 1, the result follows from Assumption 1, Propositions 1, 2, 5 and 19

#### 4.4 Evaluating with the suborder representation

In this section, we show that we can evaluate an isogeny from the suborder representation. By Proposition 7, any suborder representation  $\pi$  defines a unique isogeny that we write  $\varphi_{\pi}$ . The algorithm SuborderEvaluation that we introduce below shows how to use  $\pi$  to evaluate  $\varphi_{\pi}$ . This algorithm is going to be one of the major building blocks behind the NIKE scheme of Section 6. For this application of our algorithm, we only need to compute the image of cyclic subgroups of the form  $E_1[J]$  for some ideal J. Thus, SuborderEvaluation take a suborder representation for  $\varphi$  and an ideal J as input and outputs  $\varphi(E_1[J])$ .

The SuborderEvaluation algorithm is built on a subprotocol IdealSuborder-NormEquation 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 Let  $\mathcal{N} \subset \mathbb{N}$  be either  $\ell^{\bullet}$  for some prime  $\ell = O(\operatorname{poly}(\log(pD)))$ , or the set of divisors of T for some integer T > B of size  $O(\operatorname{poly}(pD))$  and smoothness bound  $O(\operatorname{poly}(\log(pD)))$  and where  $B = p^2 D^6 n(I)^3 n(J)^2$ . The algorithm IdealSuborderNormEquation<sub> $\mathcal{N}$ </sub> takes in input an integer D, two ideals I, J and outputs an element  $\beta \in (\mathbb{Z} + DI) \cap J$  with  $n(\beta)/n(J) \in \mathcal{N}$ , it terminates in expected  $O(\operatorname{poly}(\log(pDn(I)n(J))))$  with overwhelming probability.

The principle of SuborderEvaluation is different from the one of IdealEvaluation we sketched in Section 3.2. Indeed, as we will 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  $\mathbb{Z} + D\text{End}(E_1)$  inside  $\text{End}(E_2)$  is obtained by push-forwards through  $\varphi_{\pi}$ . More precisely, this means that  $\ker \iota(\beta) = \varphi_{\pi}(\ker \beta)$  for any  $\beta \in \mathbb{Z} + D\operatorname{End}(E_1)$ where  $\iota : \mathbb{Z} + D\operatorname{End}(E_1) \hookrightarrow \operatorname{End}(E_2)$ . Thus, to find  $\varphi_{\pi}(E_1[J])$ , we want to find an endomorphism  $\beta \in \mathbb{Z} + D\operatorname{End}(E_1)$  such that  $\ker \beta \cap E_1[n(J)] = E_1[J]$ . By definition of  $E_1[J]$ , and Assumption 2, such a  $\beta$  is exactly found by IdealSuborderNormEquation. After that, it suffices to compute  $\ker \iota(\beta) \cap E_2[n(J)]$  and we are done. The integer m is taken as in Proposition 10.

**Algorithm 4** SuborderEvaluation $(E_1, E_2, \pi, D, J)$ 

**Input:** two curves  $E_1, E_2$ , a prime  $D, \pi$  a suborder representation for  $(D, E_1, E_2) \in$  $\mathcal{L}_{\mathsf{p-isog}}$  and an ideal J of norm coprime with D and  $\ell$ . **Output:**  $\perp$  or  $\varphi_{\pi}(E_1[J])$ . 1: Parse  $\pi$  as  $\mathcal{O}, s_1, \ldots, s_n$ . 2: if  $\mathcal{O}_L(J) \not\cong \mathcal{O}$  then 3: Return  $\perp$ . 4: end if 5: if SuborderVerification  $_{\#E_1(\mathbb{F}_pm)}((D, E_1, E_2), \pi) = 0$ . then 6: Return  $\perp$ . 7: end if 8: for  $i \in [1, n]$  do 9: Parse  $s_i$  as the default representation of an isogeny of degree  $n(\theta_i) \in \ell^{\bullet}$  and compute it as  $\varphi_i : E_2 \to E_2$ . 10: end for 11: Compute  $\theta_1, \ldots, \theta_n = \mathsf{SmoothGen}_{\ell^{\bullet}}(\mathcal{O}, D).$ 12: Compute  $L = \text{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$  and I = RandomEquivalentPrimeIdeal(L)with  $I = L\alpha$ . 13: Compute  $\beta =$ IdealSuborderNormEquation $_{\ell^{\bullet}}(D, I, \alpha^{-1}J\alpha)$ . 14: Express  $\alpha\beta\alpha^{-1} = \sum_{\mathcal{I}\subset\{1,\dots,n\}} c_{i,\mathcal{I}}(\prod_{j\in\mathcal{I}}\theta_j).$ 15: Compute P, Q, a basis of  $E_2[n(J)]$ . 16: Compute  $R, S = \sum_{\mathcal{I} \subset \{1, \dots, n\}} c_{i, \mathcal{I}} (\prod_{j \in \mathcal{I}} \varphi_j) (P, Q).$ 17: if S = 0 then 18:return  $\langle Q \rangle$ . 19: end if 20: Compute  $a = \mathsf{DLP}(R, S)$ . 21: return  $\langle P - [a]Q \rangle$ .

**Proposition 11.** Under GRH and Assumptions 1 and 2, SuborderEvaluation is correct when the output is not  $\perp$  and terminates in probabilistic  $O(\operatorname{poly}(\log(pD)) + C_{\mathsf{DLP}}(n(J)))$  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  $\perp$ ,  $\pi$  is a valid suborder representation. When L =**ConnectingIdeal**( $\mathcal{O}_0, \mathcal{O}$ ) and I = **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  $\pi$  gives a correct embedding of  $\mathbb{Z} + D\mathcal{O}$  inside  $\operatorname{End}(E_1)$  and so  $\sigma = \sum_{\mathcal{I} \subset \{1,...,n\}} c_{i,\mathcal{I}} \prod_{j \in \mathcal{I}} \varphi_j$  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  $\gamma \in \operatorname{End}(E_1)$ , 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. In  $\ell^{\bullet}$ , we can always select an element  $\ell^e = O(\operatorname{poly}(\log(pDn(I)n(J))))$  of norm bigger than the bound B from Assumption 2 so the complexity follows from Assumptions 1 and 2, Propositions 1 and 10 and the fact that  $n(I) = O(\operatorname{poly}(p))$  by Proposition 14.

## 4.5 Deducing the ideal representation from the suborder representation

We saw with Proposition 8 that our new suborder representation can be computed from the ideal representation in polynomial time. The goal of this section is to study the reverse problem of extracting an ideal representation from a suborder representation. We are going to try to argue that this problem is hard in general and describe some cases where it is easy. We also introduce several other problems and prove that they are equivalent.

Problem 1. (SubOrder to Ideal Problem, SOIP) Let  $x = (D, E_1, E_2) \in \mathcal{L}_{p-isog}$ , and  $\pi$  be a suborder representation such that SuborderVerification $(x, \pi) = 1$ . Compute *I*, an ideal such that IdealVerification(x, I) = 1 or IdealVerification $((D, E_1, E_2^p), I) = 1$ .

We will show in Proposition 12 the equivalence of Problem 1 with the problem of computing the endomorphism ring of the codomain from the suborder representation (Problem 2).

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

**Proposition 12.** Under Assumption 1 and GRH, The SOIP and SOERP are equivalent.

Due to lack of space, the full proof is given in the full version of the paper but we summarize the important elements below. One of the two reductions is trivial since the right order of a solution to the SOIP is exactly a solution to the SOERP. The other reduction is more complex, the idea is that with the knowledge of the endomorphism ring of  $E_2$ , the endomorphisms of the suborder representation can be translated into principal ideals over the quaternions and with that, it is possible to compute a generator of the desired ideal. Interestingly, we can show that the SOIP is also equivalent to another problem, the Torsion to Ideal Problem (TIP) that can be seen as a generalization of the CSSI problem introduced by De Feo and Jao for SIDH [JDF11]. Due to lack of space, the proof of Proposition 13 is given in the full version.

Problem 3. (T-Torsion to Ideal (T-TI)) Let T be an integer. Let  $x = (D, E_1, E_2) \in \mathcal{L}_{isog}$  where D is coprime with T and let  $\varphi : E_1 \to E_2$  be an element of  $lsog_D$ . Let P, Q be a basis of  $E_1[T]$ . Given  $End(E_1)$  and  $\varphi(P), \varphi(Q) \in E_2[T]$ , Compute I, an ideal such that ldealVerification(x, I) = 1.

**Proposition 13.** For every D, p, there exists a value of T, such that the SOIP is equivalent to the T-TIP.

A sub-exponential quantum attack against the SOIP. To the best of our knowledge, the attack we describe below is the most efficient against the generic SOIP. We use a 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 exists a group action  $\star : G \times \mathcal{E} \to \mathcal{E}$  for which there is a malleability oracle: i.e., 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 sub-exponential time. In our context, we consider the group action of  $(End(E_1)/DEnd(E_1))^*$  on the set of cyclic subgroups of order D. This set is in correspondence with cyclic ideals of norm D inside  $\operatorname{End}(E_1)$  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 through the secret isogeny was given. This is can done with our algorithm SuborderEvaluation. As a consequence, we can evaluate  $\varphi_I$ on any subgroup of powersmooth order 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  $\operatorname{Cl}(\mathbb{Z}[\sigma])$  on the set of  $\mathbb{Z}[\sigma]$ -orientations (i.e pairs  $E, \iota$  where  $\iota : \mathbb{Z}[\sigma] \hookrightarrow \operatorname{End}(E_1)$ , see [CK19,DFFdSG<sup>+</sup>21] for more on orientations). With the knowledge of  $\sigma$ , 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. The attack we just outlined is similar to the ones exposed in [Wes22,ACL<sup>+</sup>22]. In the remaining of this section, we will describe other attacks, analyze the cases in which they prove to be efficient and explain why they fail to solve the generic SOIP.

Torsion point attacks. With the terminology torsion point attack, we designate any attack that aims at recovering an isogeny representation of a secret isogeny  $\varphi: E \to F$  from the knowledge of  $\varphi(P), \varphi(Q)$  where P, Q is a basis of E[T]for some integer T. This definition covers attacks against the T-TIP and the CSSI problem of SIDH, including the recent attacks by Castryck and Decru [CD22], Maino and Martindale [MM22] and Robert [Rob22]. These new attacks against SIDH can be seen as a generalization to higher dimension of the original torsion point attack due to Petit [Pet17]. Here is how we can explain their common generic principle: use the torsion points  $\varphi(P), \varphi(Q)$  to compute  $\theta$ , a T-endomorphism/isogeny of abelian varieties in dimension g (for some constant q) whose expression depends on  $\varphi$ . When T is big enough with respect to deg  $\varphi$ , the computation of  $\theta$  can be made solely from  $\varphi(P), \varphi(Q)$ . Then,  $\theta$  can be evaluated on the deg  $\varphi$ -torsion to recover ker  $\varphi$ . The real advantage of the new attacks against the initial idea of Petit is that they reduce the constraint to  $T > \deg \varphi$ which mean they can be applied to SIDH. With our algorithm SuborderEvaluation, it is possible to get the image of any subgroup under the isogeny  $\varphi$  of degree D from a suborder representation  $\pi_{\varphi}$ . Thus, it is always possible to apply a torsion point attack to the setting of the SOIP. However, the complexity of this attack will not always be polynomial. The main obstacle seems to be the field of definition of the D-torsion. In general, for a random integer D, we can expect the D-torsion to be defined over  $\mathbb{F}_{p^k}$  where k = O(D). This is true in particular when D is prime. When the field of definition of the torsion point is too big, there does not seem to be any way to express the kernel of  $\varphi$  in a compact manner and thus the attack does not have a polynomial complexity.

On the other hand, when the degree k of the field extension is polynomial in D, there is a quantum polynomial attack against the SOIP. Indeed, in this case, the torsion point attacks allow us to compute the kernel of  $\varphi$  in polynomial time and this kernel admits a representation of polynomial size. Then, the only remaining task to solve the SOIP is to compute the ideal I corresponding to  $\varphi$ . Since the endomorphism ring of the domain  $E_1$  is known, this can be done in quantum polynomial time using the algorithm from Galbraith, Petit and Silva [GPS17, Algorithm 3]. It is only quantum polynomial time because the algorithm requires to solve some DLPs over the D-torsion, every other aspect of the algorithm can be executed in classical polynomial time.

To conclude, we need that the *D*-torsion is not defined over a small field extension to ensure hardness of the SOIP. Fortunately, this should happen with overwhelming probability when D is chosen at random, and it can be verified by computing the order of  $p \mod D$  (the degree k is equal to this order up to a factor 2).

Other attacks. 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  $\operatorname{End}(E_1)$  is part of the suborder representation, it suffices to enumerate through all  $\operatorname{End}(E_1)$  ideals of norm D until IdealVerification 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 provides a quadratic speedup over a brute-force search, and is considered to be the most efficient method to find an isogeny of smooth degree between two random supersingular curves.

Another way to solve the problem in a generic manner is by computing  $\operatorname{End}(E_2)$  (see Proposition 12). Without using the proof  $\pi$  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<sup>+</sup>20]).

Even after seeing the above 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<sup>+20]</sup> 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 behind this algorithm 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 solution. The authors from [EHL<sup>+</sup>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 representation are not random cycles. By design, the suborder they generate is not Bass and we know 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 representation, the algorithm described in [EHL<sup>+</sup>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  $\mathbb{Z} + D\text{End}(E_1)$  inside  $\text{End}(E_2)$  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 IdealTolsogeny). This supposed attack would have to work over orders of non-trivial Brandt invariant rather than maximal orders to exploit the suborder representation. 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 6 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 CheckTrace<sub>M</sub> in Sections 5.2 to 5.4 respectively. We recall that the first algorithm is used to evaluate isogenies from the suborder representation in SuborderEvaluation (Algorithm 4 of Section 4.4) and the last two are building blocks for SuborderVerification (Algorithm 3 of Section 4.3) for the verification of our new suborder representation. Note that IdealSuborderNormEquation and SmoothGen are only heuristic as for the algorithms from [KLPT14,DFKL<sup>+</sup>20]. We expand on this matter in Remark 3.

We use the basis 1, i, j, k for  $B_{p,\infty}$  where  $i^2 = -q, j^2 = -p$  and k = ij = -jifor some small integer q > 0 (see [KLPT14] for values of q for all p, when p = 3mod 4 we can take q = 1). Following the classical approach in the literature ([KLPT14,DFKL<sup>+</sup>20]), we take  $\mathcal{O}_0 \subset B_{p,\infty}$  as a special extremal order as defined in [KLPT14], i.e., a maximal 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.

- RandomEquivalentPrimeldeal(I), given a left  $\mathcal{O}_0$ -ideal I, finds an equivalent left  $\mathcal{O}_0$ -ideal of prime norm.
- IdealModConstraint $(I, \gamma)$ , 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  $\gamma \mu_0 \in I$ .
- 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  $\gamma \mu_0 \in \mathbb{Z} + I$ .
- StrongApproximation<sub> $\mathcal{N}$ </sub> $(N, C_0, D_0)$ , given a prime N and  $C_0, D_0 \in \mathbb{Z}$ , finds  $\mu = \lambda \mu_0 + N \mu' \in \mathcal{O}_0$  of norm in  $\mathcal{N}$ , with  $\mu_0 = j(C_0 + \omega D_0)$  and  $\mu' \in \mathcal{O}_0$ .

The following result on the size of the output of RandomEquivalentPrimeldeal will prove useful.

**Proposition 14.** Let I be an integral ideal of maximal orders. The output J = RandomEquivalentPrimeIdeal(I) has norm n(J) = O(poly(p)).

Remark 3. The algorithms that we have introduced above are all expected to terminate in polynomial-time under various plausible heuristic assumptions introduced in [KLPT14,DFKL<sup>+</sup>20]. By plausible, we mean that these assumptions were verified experimentally. These assumptions mostly concern the probability that some integers represented by specific quadratic forms are prime and satisfy some quadratic reduosity condition (as in Remark 4 for instance, see also [Wes22] for more details). Our new algorithms are based on the sub-algorithms from [KLPT14] and this is why our results will be subject to the same assumptions. However, these assumptions are only used to justify the termination and expected running time of the sub-algorithms, and so they do not appear directly in our proofs, and this is also why we do not state them clearly.

Remark 4. The StrongApproximation<sub> $\mathcal{N}$ </sub> $(N, \cdot)$  algorithm was originally introduced for a prime number N in [KLPT14]. The probability of success depends on some quadratic reduosity condition mod N. 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. In general, under the heuristic assumption that the integers we consider mod N behave like random integers of the same size, we can see that the success probability should be  $1/2^k$ 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. Each of our norm equation algorithm is parameterized by a set  $\mathcal{N} \subset \mathbb{N}$  that defines the possible norm of the outputs. This set  $\mathcal{N}$  can be either  $\ell^{\bullet}$  for some prime  $\ell$  or  $\mathcal{M}(T)$ , the divisors of T for some  $T \in \mathbb{N}$ .

The first algorithms targetting that task were introduced in [KLPT14] where  $\Lambda$  was either a special extremal maximal order like  $\mathcal{O}_0$  or an ideal of left (and right) maximal order. In [DFKL<sup>+</sup>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<sup>+</sup>20] that consists 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  $I = \text{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$ , 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<sup>+</sup>20] and ours. We will explain how the ideas from [KLPT14,DFKL<sup>+</sup>20] fit into a 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 parameterized by two integers  $N_1, N_2$ . We look for elements of norm contained in some set  $\mathcal{N} \subset \mathbb{N}$ . In practice  $\mathcal{N}$  is going to be either  $\ell^{\bullet}$  or the divisors of some powersmooth integer T. The algorithms can be decomposed as follows:

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

The goal of these "conditions" on  $\gamma$  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  $\gamma$ . 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 EquivalentRandomPrimeldeal 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  $\Lambda$ is an  $\mathcal{O}_0$ -ideal of norm N. The condition on  $\gamma$  is summarized by Lemma 6 that is a reformulation of some of the results from [KLPT14]. We have  $N_1 = N$  and  $N_2 = N$ .

**Lemma 6.** [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.

The goal of the authors of  $[DFKL^+20]$  was to obtain a generalization of the algorithm of [KLPT14] when  $\Lambda$  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  $\gamma$  can be summarized with Lemma 7. In that case,  $N_1 = 1$  and  $N_2 = N$ .

**Lemma 7.**  $[DFKL^+ 20]$  Let I be an  $\mathcal{O}_0$  ideal. 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<sup>+</sup>20, Section 5] is simply obtained by combining Lemmas 6 and 7 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  $\gamma$  to satisfy more conditions than a simple constraint on its norm. We will introduce the necessary condition in Proposition 15. 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  $\gamma$  while ensuring that the linear algebra step always has a solution.

**Proposition 15.** Let I be an integral left  $\mathcal{O}_0$ -ideal of norm N and let D be a prime number distinct from N. 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  $\gamma$  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  $\gamma$  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 7). 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  $C_1 = C_0 \mod N, D_1 = D_0 \mod N$ ,  $C_1 = C'_2, D_1 = D'_2 \mod D$  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 15, 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  $\gamma$ . When we apply these ideas to the framework described above, we obtain EichlerSuborderNormEquation.

We remind the reader that the heuristics in Proposition 16 are the same as the ones from [KLPT14] (see Remark 3). This goes for Propositions 17 and 18 as well.

**Proposition 16.** (Heuristic) When N, D are distinct primes, Algorithm 5 terminates in expected  $O(\operatorname{poly}(\log(DN)))$  and outputs an element of  $\mathbb{Z}+DI$  of norm in  $\mathcal{N}$  when  $\mathcal{N}$  contains an elements bigger than  $p^{7/2}D^6$ . The expected norm is in  $O(\operatorname{poly}(p, D, N))$ .

*Proof.* As mentioned in Remark 4, because D is prime, under plausible heuristics, the algorithm  $\mathsf{StrongApproximation}_{\mathcal{N}}(D, \cdot)$  finds a solution of norm in  $\mathcal{N}$  with probability at least 1/2 in polynomial time when  $\mathcal{N}$  contain a big enough element (we will look at the required size at the end of the proof). As a result

**Algorithm 5** EichlerSuborderNormEquation  $\mathcal{N}(D, I)$ 

- **Input:** I a left  $\mathcal{O}_0$ -ideal of norm N coprime with D.
- **Output:**  $\beta \in \mathbb{Z} + DI$  of norm dividing F.
- 1: Select a random class  $(C_2 : D_2) \in \mathbb{P}^1(\mathbb{Z}/D\mathbb{Z})$ .
- 2: Compute  $\mu_2 = \text{StrongApproximation}_{\mathcal{N}}(D, C_2, D_2)$ ). If the computation fails, go back to Step 1.
- 3: Compute  $(C_0: D_0) = \text{EichlerModConstraint}(\mu_2, I)$ .
- 4: Sample a random  $D'_2$  in  $\mathbb{Z}/D\mathbb{Z}$ , compute  $C'_2 = -D'_2C_2(D_2)^{-1} \mod D$ .
- 5: Compute  $C_1 = \mathsf{CRT}_{N,D}(C_0, C'_2), D_1 = \mathsf{CRT}_{N,D}(D_0, D'_2).$
- 6: Compute  $\mu_1 = \mathsf{StrongApproximation}_{\mathcal{N}}(ND, C_1, D_1)$ . If it fails, go back to step 1.
- 7: return  $\beta = \mu_2 \mu_1$ .

of Proposition 15, EichlerModConstraint always succeeds in finding a solution  $(C_0: D_0)$ . Then, the second 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 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  $\beta$  of norm in  $\mathcal{N}$  if  $\mathcal{N}$  contains elements of size  $\approx 2 \log_{\ell}(p) + 6 \log_{\ell}(D) + 3 \log_{\ell}(N)$ . This comes from the fact that a strong approximation mod N' can find solutions of norm approximately equal to  $pN'^3$ . Other estimates provided in  $[DFKL^+20]$  prove that we will have  $N \approx \sqrt{p}$  and this yields the final bound  $p^{7/2}D^6$ .

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 6 and 7 and Proposition 15 and take  $N_1 = N'$ ,  $N_2 = NN'D$ . This yields Algorithm 6.

#### **Proposition 17.** (Heuristic) Assumption 2 holds.

Proof. Due to Lemmas 6 and 7 and Proposition 15, 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  $O(\text{poly}(\log(DNN')))$  since it consists of linear algebra mod D, N, N'. The executions of Strong Approximations terminates in probabilistic polynomial time and output a value with constant probability. So the global computations terminates in probabilistic  $O(\text{poly}(\log(DNN')))$ . It is correct because StrongApproximation is correct. The computation succeeds as soon as the target set  $\mathcal{N}$  contains elements that have size bigger than  $2\log_{\ell}(p) + 6\log_{\ell}(D) + 3\log_{\ell}(N) + 2\log_{\ell}(N')$  and this is the value we can take for the bound B ( the first execution of StrongApproximation gives an element of size  $\sim pD^3/N'$  and the second  $p(DNN')^3$ ).

### Algorithm 6 IdealSuborderNormEquation<sub> $\mathcal{N}$ </sub>(D, I, J)

**Input:** An integer D, and I, J two left  $\mathcal{O}_0$ -ideals of norm N, N' with gcd(N, N', D) = 1.

**Output:**  $\beta \in (\mathbb{Z} + DI) \cap J$  of norm N'N'' where  $N'' \in \mathcal{N}$ .

- 1: Select a random class  $(C_2 : D_2) \in \mathbb{P}^1(\mathbb{Z}/D\mathbb{Z})$ .
- 2: Compute  $\mu_2 = \text{StrongApproximation}_{\mathcal{N}}(D, C_2, D_2)$ ). If the computation fails or if  $gcd(n(\mu_2), N') = 1$ , go back to Step 1.
- 3: Compute  $(C_0 : D_0) = \text{EichlerModConstraint}(\mu_2, I)$ .
- 4: Compute  $(C_3: D_3) = \mathsf{IdealModConstraint}(\mu_2, J)$ .
- 5: Sample a random  $D'_2$  in  $\mathbb{Z}/D\mathbb{Z}$ , compute  $C'_2 = -D'_2C_2(D_2)^{-1} \mod D$ .
- 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  $\mu_1 = \mathsf{StrongApproximation}_{\mathcal{N}}(NDN', C_1, D_1)$ . If it fails, go back to step 1.

#### 5.3 Computing a smooth generating family

In this section, we describe the SmoothGen algorithm that takes in input a maximal order  $\mathcal{O}$  and a prime D, outputs a generating family of  $\mathbb{Z} + D\mathcal{O}$  of elements whose norms are in  $\mathcal{N}$ . The idea behind this algorithm is quite straightforward: apply EichlerSuborderNormEquation on I, for various ideals I connecting  $\mathcal{O}_0$  and orders isomorphic to  $\mathcal{O}$ . This gives a way to sample elements in  $\mathbb{Z} + D\mathcal{O}$ , and we iterate this method until we obtain a generating family from this set. Experimental results show that after taking a few elements in that manner (for instance, no more than ten for parameters of cryptographic sizes, i.e., of a few hundred bits), we can extract a generating family of size three. We formulate this more precisely as Conjecture 1.

Conjecture 1. Let  $\mathcal{O}_1$  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 EichlerSuborderNormEquation(D, I<sub>i</sub>) for i = 1, 2, 3, then  $\mathbb{Z} + D\mathcal{O} = \text{Order}(\alpha_1\theta_1\alpha_1^{-1}, \alpha_2\theta_2\alpha_2^{-1}, \alpha_3\theta_3\alpha_3^{-1})$  with probability 1/c where  $c = O(\text{poly}(\log(pD)))$ .

Proposition 18. (Heuristic) Assuming Conjecture 1, Assumption 1 holds.

*Proof.* Proposition 1 proves the desired running time for ConnectingIdeal. The same holds for RandomEquivalentPrimeIdeal and the outputs of this algorithm have norms in O(poly(p)) by Proposition 14. 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 16 and the outputs have norm in O(poly(pD)). By the termination condition, the output is a generating family of  $\mathbb{Z} + D\mathcal{O}$ .

<sup>8:</sup> return  $\beta = \mu_2 \mu_1$ .

Algorithm 7 SmoothGen<sub>N</sub>( $\mathcal{O}, D$ ) Input: A maximal order  $\mathcal{O}$  and a prime D. Output: A generating family  $\theta_1, \theta_2, \theta_3$  for  $\mathbb{Z} + D\mathcal{O}$  where each  $\theta_j$  has norm in  $\mathcal{N}$ . 1: Set  $L = \emptyset$  and  $I_0 = \text{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$ . 2: while There does not exist  $\theta_1, \theta_2, \theta_3 \in L$  s.t  $\mathbb{Z} + D\mathcal{O} = \text{Order}(\theta_1, \theta_2, \theta_3)$  do 3:  $I = \text{RandomEquivalentPrimeIdeal}(I_0)$  and  $I = I_0 \alpha$ . 4: Compute  $\theta = \text{EichlerSuborderNormEquation}_{\mathcal{N}}(D, J)$ . 5:  $L = L \cup \{\alpha \theta \alpha^{-1}\}.$ 6: end while 7: return  $\theta_1, \theta_2, \theta_3$ .

#### 5.4 Checking traces

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

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  $\operatorname{tr}(\theta) = \theta + \hat{\theta}$ , it suffices to evaluate  $\theta$  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 want to check equality.

Algorithm 8 CheckTrace<sub>M</sub>(E,  $\varphi_1, \ldots, \varphi_n, \theta_1, \ldots, \theta_n$ ) Input:  $\theta_1, \ldots, \theta_n$ , *n* endomorphisms of E and *n* elements of  $B_{p,\infty} \ \omega_1, \ldots, \omega_n$ . Output: A bit *b* equal to 1 if and only if  $\operatorname{tr}(\theta_i) = \operatorname{tr}(\omega_i) \mod M$  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  $\varphi_{\mathcal{I}}(R) + \hat{\varphi}_{\mathcal{I}}(R) = [\operatorname{tr}(\theta_{\mathcal{I}})]R$  for  $R \in \{P, Q\}$ . If not, set b = 0. 5: end for 6: return b.

**Proposition 19.** When  $M = #E(\mathbb{F}_{p^m})$ , n = O(1) and  $\deg \varphi_i = O(\mathsf{poly}(p))$ and have smoothness bound in  $O(\mathsf{poly}(\log(p)))$  for all  $1 \le i \le n$ ,  $\mathsf{CheckTrace}_M$ terminates in  $O(\mathsf{poly}(m \log(p)))$ 

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

## 6 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) but with prime degrees. For secret keys we propose to use ideal representations and then take suborder representations as public keys. The key exchange will be made possible with SuborderEvaluation (Algorithm 4 of Section 4.4). The full description can be found in In terms of security, the pSIDH key recovery problem is exactly the SOIP and our NIKE is secure under the hardness of a decisional variant of the SOIP (Problem 1) in a similar manner to SIDH with the CSSI and SSDDH problems introduced in [JDF11].

As pSIDH is a NIKE and the best attack is quantum-subexponential (see Section 4.5), pSIDH has an application profile similar to CSIDH (there is even a group action involved). The SOIP in itself is closer to the key recovery problem of CSIDH than it is to the one of SIDH (in the sense that they can be both seen as isogeny problems with partial endomorphism ring information which is not really the case for SIDH). However, despite some similarities, the protocols relies on different assumptions. Moreover, the underlying structure in pSIDH is not the same as in CSIDH so it might open new possibilities.

We discuss a concrete instantiation and the efficiency of pSIDH in Section 6.2, where we also compare with the efficiency of CSIDH.

### 6.1 The description of pSIDH

The idea of SIDH is the following: the two participants Alice and Bob generate isogenies  $\varphi_A, \varphi_B$  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)). 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). For efficiency, the degrees  $N_A, N_B$  need to be smooth.

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. We propose to use the ideal representations as secret keys and the suborder representations as public keys. The computation of the common key j(E) can be done as follows. Given an ideal I of norm  $D_A$  and the suborder  $\mathbb{Z} + D_B \mathcal{O}_0$ , 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 6 in Section 5.2). The embedding  $\iota_B : \mathbb{Z} + D_B \mathcal{O}_0 \hookrightarrow \text{End}(E_B)$ , is obtained by pushing forward the embedding of  $\mathbb{Z} + D_B \mathcal{O}_0$  inside  $\text{End}(E_0)$  through  $\varphi_B$  and so we have  $\iota_B(\theta) =$  $\psi_A \circ [\varphi_B]_* \varphi_A$  where  $\psi_A$  has degree S. Thus, using  $\pi_B$ , the suborder representation of  $\varphi_B$ , 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. 1 and the full description of the key exchange mechanism is given as Algorithm 10. The key generation algorithm is also described in Algorithm 9. To be able to run this algorithm in polynomialtime, we need to be able to compute efficiently isogenies of degree  $\psi_A$  and to be able to manipulate the full deg  $\psi_A$  torsion. This is why we take the degree of  $\psi_A$  as a divisor of a powersmooth integer T. To be able to apply SuborderEvaluation, we also need that T is coprime with the degree of the endomorphisms of the suborder representation (so we take T coprime with  $\ell$ ). We write  $\mathcal{M}(T)$  for the set of divisors of T. The integer m is taken to be as in Proposition 10.

The public parameters of pSIDH should include a prime p and a starting curve  $E_0$  together with a description of  $\operatorname{End}(E_0)$ . For simplicity, we may assume that  $\operatorname{End}(E_0) \cong \mathcal{O}_0$ , where  $\mathcal{O}_0$  is the special extremal order introduced in the beginning of Section 5.



Fig. 1. SIDH/pSIDH-isogeny diagram.

#### **Algorithm 9** KeyGeneration(D)

**Input:** A prime number  $D \neq p$ .

- **Output:** The pSIDH public key  $\mathsf{pk} = E, \pi$  and the pSIDH secret key  $\mathsf{sk} = I$  where  $\pi$  is a suborder representation and I an ideal representation 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  $\pi$ .
- 3: return  $pk, sk = (E, \pi), I.$

**Proposition 20.** Under GRH, Assumption 1, Assumption 2, KeyExchange terminates in expected poly(log(pD'D)).

#### Algorithm 10 KeyExchange $(I, D', E', \pi)$

**Input:** I an ideal of degree D and a prime  $D' \neq D, p$ . A curve E' and a suborder representation  $\pi$ .

**Output:** A *j*-invariant or  $\perp$ .

1: Parse  $\pi = (\mathcal{O}_0, \varphi_1, \dots, \varphi_n).$ 

- 2: Compute  $\theta_1, \dots, \theta_n = \mathsf{SmoothGen}_{\ell^{\bullet}}(\mathcal{O}_0, D').$
- 3: if !SuborderVerification $_{\#E'(\mathbb{F}_{p^m})}((D', E_0, E'), \pi)$  then
- 4: Return  $\perp$ .
- 5: **end if**
- 6: Take a powersmooth integer T coprime with  $\ell$  with B < T < 2B where B is the bound in Assumption 2 and T has the smallest possible smoothness bound.
- 7: Set  $J = O_0 1$ .
- 8: Compute  $\theta =$ IdealSuborderNormEquation<sub> $\mathcal{M}(T)$ </sub>(D', J, I).
- 9: Factorize  $T = \prod_{i=1}^{r} \ell_i^{e_i}$ .
- 10: Set  $G = \langle 0_{E'} \rangle$ .
- 11: for  $i\in [1,r]$  do
- 12: Compute  $J_i = \mathcal{O}_0 \overline{\alpha^{-1} \theta \alpha} + \mathcal{O}_0 \ell_i^{e_i}$ .
- 13:  $G = G + \text{SuborderEvaluation}(E_0, E', \pi, D', J_i).$
- 14: **end for**
- 15: Compute  $\psi: E' \to E'/G$ .
- 16: **return** j(E'/G).

*Proof.* Since  $B = O(\operatorname{poly}(\log(pDD')))$ , we can choose a value of T with a smoothness bound equal in  $O(\operatorname{poly}(\log(pDD')))$ . Thus, all operations over the T-torsion and the final computation of  $\psi$  can be done in  $O(\operatorname{poly}(\log(pD'D)))$ . The remaining computations terminate in expected  $O(\operatorname{poly}(\log(pD'D)))$  due to Assumptions 1 and 2 and Propositions 1, 10, 11 and 14.

**Proposition 21.** Let  $D_A, D_B \neq p$  be two distinct prime numbers. If  $E_A, \pi_A, I_A = \text{KeyGen}(D_A)$  and  $E_B, \pi_B, I_B = \text{KeyGen}(D_B)$ , 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  $\varphi_A, \varphi_B$  the isogenies corresponding to the two ideals  $I_A, I_B$ . Let us write  $\theta_A$  the quaternion element defined in Step 8 of KeyExchange $(I_A, D_B, E_B, \pi_B)$ . Then, the quaternion element  $\overline{\alpha_A^{-1}}\theta_A\alpha_A \in (\mathbb{Z} + D_B\mathcal{O}_0) \cap I_A$  corresponds to an endomorphism  $\psi_{A,0} \circ \varphi_A \in \text{End}(E_0)$  for some isogeny  $\psi_{A,0} : E_A \to E_0$ . Since it is contained in  $(\mathbb{Z} + D_B\mathcal{O}_0) \cap I_A$ , we can embed it inside the endomorphism ring of  $E_B$  by Proposition 7 and we obtain in that manner the endomorphism  $\psi_A \circ [\varphi_B]_* \varphi_A$  where  $\hat{\psi}_A = [\varphi_B]_* \hat{\psi}_{A,0}$ . In particular, the codomain of  $\hat{\psi}_A$  is isomorphic to the codomain of  $[\varphi_B]_* \varphi_A$ . We can make the same reasoning by swapping A and B and by definition of push-forward isogenies and Proposition 11, the two j-invariants obtained at the end of the two executions of KeyExchange are equal.

Remark 5. The purpose of Algorithm 10 is to present a simple version of the protocol for the key exchange. However, as it is written, our solution is not very optimized. For instance, a lot of redundant computations are made through the

call to SuborderEvaluation. In an optimized implementation of this key exchange, one would want to skip all the first steps which are already executed in KeyExchange to focus on the important steps.

We analyze the security of pSIDH in the full version of the paper, it is very similar to the security proof of SIDH.

#### 6.2 About efficiency and concrete instantiations

*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. We did not make a full implementation but we can obtain a good idea of the efficiency by comparison with the SQISign signature  $[DFKL^+20]$ . 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. It is unclear what should be the size of D but we can expect it to be bigger than p. This gap between pand 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 IdealTolsogeny sub-algorithm. The method introduced in [DFKL<sup>+</sup>20] and the improvement in [DFLW22] for IdealTolsogeny both 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 degree  $2^e$ where e is linear in the security parameter. For pSIDH, the size estimates from Section 5.2 show that we may expect element of degree whose logarithm is in  $6 \log(D)$  (and some linear dependency on  $\log(p)$ ).

On a concrete instantiation. We believe that finding a parameter D to reach a NIST-1 level of security for pSIDH is a problem on its own. However, we can easily find parameters that reaches the same security level as CSIDH-512. For that, we can take p of at least 256-bits (one of the SQISign primes should be good) and we can take  $E_0$  to be any starting curve of known endomorphism ring. For instance, if  $p = 3 \mod 4$ , we can take the curve of j-invariant 1728 with endomorphism ring isomorphic to  $\langle 1, i, \frac{1+k}{2}, \frac{i+j}{2} \rangle$  where 1, i, j, k is the canonical basis of the quaternion algebra ramified at p and  $\infty$ .

We need D of at least 256-bits as well (so that the set of subgroups of order D has the same size as the class number inf CSIDH-512). We remind the reader that, for the hardness of the SOIP, the D-torsion of supersingular curves in

characteristic p needs to be defined over an extension of big degree (roughly equal to  $2^{256}$  to have the best possible security). This condition can be checked by computing the order of  $p \mod D$ . If (D-1)/2 is prime, then the computation of the order will have polynomial time (because D-1 is easy to factor) and the order of  $p \mod D$  is going to be bigger than (D-1)/2 with overwhelming probability. Such a prime D can be found after trying roughly log D primes. Apart from that, there is no constraint on the choice of D.

Even though we did not make an implementation, it is clear, looking at the latest performances of SQISign [DFLW22], that an implementation of pSIDH with the parameters we propose, is going to be a lot slower than CSIDH-512.

However, we want to stress that the asymptotic behaviour is rather on the side of pSIDH. Indeed, as we said, the complexity of pSIDH is linear in  $\log(D)$  whereas the complexity of CSIDH is worst than linear in  $\log(p)$  (and the quantum attack is sub-exponential in  $\log(p)$  for CSIDH).

#### Acknowledgements

We are very grateful to Steven Galbraith for a very thorough review of the paper and numerous comments to help improve the current write-up. We would also like to thank anonymous reviewers for their insight on our work. Finally, we thank Luca De Feo for useful remarks regarding the best way to define an isogeny representation.

## References

$ACL^+22.$	Sarah Arpin, Mingjie Chen, Kristin E Lauter, Renate Scheidler, Kather-
	ine E Stange, and Ha TN Tran. Orienteering with one endomorphism.
	arXiv preprint arXiv:2201.11079, 2022.
BDF21.	Jeffrey Burdges and Luca De Feo. Delay encryption. In EUROCRYPT,
	pages 302–326. Springer, 2021.
BdFLS20.	Daniel J. Bernstein, Luca de Feo, Antonin Leroux, and Benjamin Smith.
	Faster computation of isogenies of large prime degree. In Steven Gal-
	braith, editor, ANTS-XIV - 14th Algorithmic Number Theory Sympo-
	sium, pages 39–55, Auckland, New Zealand, June 2020.
BJS14.	Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum al-
	gorithm for computing isogenies between supersingular elliptic curves.
	In International Conference on Cryptology in India, pages 428–442.
	Springer, 2014.
BKV19.	Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. Csi-fish:
	Efficient isogeny based signatures through class group computations. In
	ASIACRYPT, pages 227–247. Springer, 2019.
BKW20.	Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudoran-
	dom functions from isogenies. In ASIACRYPT, pages 520–550. Springer,
	2020.
CD22.	Wouter Castryck and Thomas Decru. An efficient key recovery attack
	on SIDH (preliminary version). Cryptology ePrint Archive, 2022.

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.

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

- CSRHT22. Jorge Chavez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum vdf. In *Selected Areas in Cryptography*, pages 441–460. Springer, 2022.
- DFFdSG<sup>+</sup>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 *EUROCRYPT*, pages 759–789. Springer, 2019.
- DFKL<sup>+</sup>20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In *ASIACRYPT*, pages 64–93. Springer, 2020.
- DFLW22. Luca De Feo, Antonin Leroux, and Benjamin Wesolowski. New algorithms for the deuring correspondence: SQISign twice as fast. Cryptology ePrint Archive, 2022.
- DFMPS19. Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In ASIACRYPT, pages 248–277. Springer, 2019.
- EHL<sup>+</sup>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<sup>+</sup>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.
- FKMT22. Tako Boris Fouotsa, Péter Kutas, Simon-Philipp Merz, and Yan Bo Ti. On the isogeny problem with torsion point information. In *PKC*, pages 142–161. Springer, 2022.
- FP22. Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on sidh. In Cryptographers' Track at the RSA Conference, pages 322–344. Springer, 2022.
- GPS17. Steven D Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In ASIACRYPT, pages 3–33. Springer, 2017.
- GPST16. Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In ASIACRYPT, pages 63–91. Springer, 2016.
- 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 Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion-isogeny path problem. LMS Journal of Computation and Mathematics, 17(A):418–432, 2014.

- KMP<sup>+</sup>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 *EUROCRYPT*, pages 242–271. Springer, 2021.
- Koh96. D. Kohel. Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California at Berkeley, 1996.

MM22. Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. *Cryptology ePrint Archive*, 2022.

- Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *ASIACRYPT*, pages 330–353. Springer, 2017.
- Rob22. Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, 2022.
- Sch95. René Schoof. Counting points on elliptic curves over finite fields. Journal de théorie des nombres de Bordeaux, 7(1):219–254, 1995.
- UXT<sup>+</sup>22. Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/em analysis on post-quantum kems. *IACR Transactions on Cryptographic Hard*ware and Embedded Systems, pages 296–322, 2022.
- 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<sup>+</sup>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.