Hidden Stabilizers, the Isogeny To Endomorphism Ring Problem and the Cryptanalysis of pSIDH

Mingjie Chen[1][0009−0003−8274−0685], Muhammad Imran[2][0000−0002−0776−2644], Gábor Ivanyos[3][0000−0003−3826−1735], Péter Kutas[1,6][0000−0002−2043−9542], Antonin Leroux[4,5], and Christophe Petit[1,7][0000−0003−3482−6743]

[1] University of Birmingham, UK
[2] Budapest University of Technology and Economics, Hungary
[3] Institute for Computer Science and Control, Hungarian Research Network, Hungary
[4] DGA-MI, Bruz, France
[5] IRMAR, UMR 6625, Université de Rennes, France
[6] Eötvös Loránd University, Hungary
[7] Université libre de Bruxelles, Belgium

Abstract. The *Isogeny to Endomorphism Ring Problem* (IsERP) asks to compute the endomorphism ring of the codomain of an isogeny between supersingular curves in characteristic p given only a *representation* for this isogeny, i.e. some data and an algorithm to evaluate this isogeny on any torsion point. This problem plays a central role in isogeny-based cryptography; it underlies the security of pSIDH protocol (ASIACRYPT 2022) and it is at the heart of the recent attacks that broke the SIDH key exchange. Prior to this work, no efficient algorithm was known to solve IsERP for a generic isogeny degree, the hardest case seemingly when the degree is prime.

In this paper, we introduce a new quantum polynomial-time algorithm to solve IsERP for isogenies whose degrees are odd and have $O(\log\log p)$ many prime factors. As main technical tools, our algorithm uses a quantum algorithm for computing hidden Borel subgroups, a group action on supersingular isogenies from EUROCRYPT 2021, various algorithms for the Deuring correspondence and a new algorithm to lift arbitrary quaternion order elements modulo an odd integer N with $O(\log\log p)$ many prime factors to powersmooth elements.

As a main consequence for cryptography, we obtain a quantum polynomial-time key recovery attack on pSIDH. The technical tools we use may also be of independent interest.

1 Introduction

The problem of computing an isogeny between two supersingular elliptic curves is believed to be hard, even for a quantum computer. The assumption that this statement is true led to the idea of using isogenies to build post-quantum cryptography.

However, building actual cryptography from this principle is not easy and the security of concrete isogeny-based protocols is based on weaker versions of the isogeny problem, where the attacker is given more information. The nature of this additional information differs from one proposal to another but the heart of the problem remains the same.

At the core of the cryptanalytic efforts to attack the isogeny problems lies another problem: the endomorphism ring problem, which requires to compute the endomorphism ring of a curve given in input. In fact, computing isogenies and computing endomorphism rings are computationally equivalent problem for supersingular curves [EHL+18, Wes21]. However, this equivalence result does not fully answer the following question : given a "reasonable" representation of an isogeny $\phi : E \to E'$ and the knowledge of the endomorphism ring of the starting curve E, can we always efficiently compute the endomorphism ring of the codomain E'? This question leads to the following problem, where the exact definition of *weak isogeny representation* will be given in Section 2.4.

Problem 1.1 (Isogeny to Endomorphism Ring Problem (IsERP)). Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ and let $\phi : E \to E_1$ be an isogeny of degree N for some integer N. Given $\mathrm{End}(E)$ and a weak isogeny representation for ϕ, compute $\mathrm{End}(E_1)$.

The answer to this question is known to be yes when the degree of ϕ is powersmooth (and this is what is used in the equivalence results mentioned above), but the question remains open for an arbitrary degree. For a prime degree, this problem can be seen as the generalization of the key recovery problem for the pSIDH scheme recently introduced by Leroux [Ler22a]. The best known algorithm has subexponential quantum complexity in N, and the generic endomorphism ring attack has complexity exponential in $\log p$.

Isogeny-based cryptography. Isogeny-based cryptography originates in Couveignes' seminal work [Cou99] where he proposed to use the natural class group action on ordinary elliptic curves to instantiate a potentially quantum-resistant version of the Diffie-Hellman key exchange. The reasoning for that is that the discrete logarithm problem has more structure than needed to instantiate a key exchange, and this structure is exploited in Shor's algorithm [Sho97]. Couveignes' ideas were rediscovered by Rostovtsev and Stolbunov [RS06] and thus the resulting scheme is referred to as the CRS key exchange. These ideas were far from practical and a major breakthrough came with the invention of CSIDH [CLM+18]. The idea is quite similar but one uses supersingular elliptic curves defined over $\mathbb{F}_p$ and the acting group is the class group of $\mathbb{Z}[\sqrt{-p}]$. In other words one considers supersingular curves defined over $\mathbb{F}_p$ together with isogenies defined over $\mathbb{F}_p$ as well.

The same idea does not apply to supersingular curves defined over $\mathbb{F}_{p^2}$ because the endomorphism rings are non-commutative (hence the natural class group action of left ideals modulo principal left ideals is a non-commutative group action). This means that providing codomains of secret isogenies (i.e., curves E_A, E_B) is not enough to arrive at a shared secret that both parties can compute. Thus in order to instantiate a Diffie-Hellman-like key exchange on the full set of supersingular curves parties must provide additional information. In 2011 De Feo and Jao proposed SIDH [JDF11] where both parties share the images of other person's torsion basis under their secret isogeny. This motivated the following problem:

Problem 1.2. Let E be a supersingular elliptic curve and let A, B be coprime smooth numbers. Let $\phi : E \to E_A$ be a secret isogeny of degree A. One is provided with the action of ϕ on $E[B]$. Compute ϕ.

In [Pet17] it was shown that this problem can be solved in polynomial time for certain parameter sets (where $B > p^2A^2$). In order to instantiate SIDH efficiently one usually uses parameters A, B, p such that AB divides $p + 1$ as then all computations can be carried out over $\mathbb{F}_{p^2}$ so in some sense these initial results seemed theoretical. Then the initial idea of Petit [Pet17] was improved in [QKL⁺21] to $B > \sqrt{p}A^2$ which already included parameter sets which could have been used in SIDH variants. Nevertheless none of these attacks directly impacted SIDH where A and B are roughly the same size. Then in 2022 Castryck and Decru [CD22] (and independently Maino and Martindale [MM22]) vastly improved these using ingenious techniques (utilizing superspecial abelian surfaces) which break SIDH with known endomorphism ring in polynomial time even if A and B are balanced. Finally, Robert proposed a polynomial-time attack on SIDH with unknown endomorphism ring (furthermore, he only needs $B^2 > A$ as opposed to $B > A$ in other attacks).

These attacks have shown that using smooth degree isogenies and providing torsion point information will potentially not lead to secure and efficient cryptographic constructions (in [FMP23] some countermeasures are proposed, but the ones that are not broken are much less efficient than the original SIDH construction). Thus, in order to navigate in the supersingular isogeny graph parties have to share some other kind of extra information.

Alternative isogeny representations. In the pSIDH protocol introduced by Leroux [Ler22a], one reveals *suborder representations* for isogenies of large prime degrees to build a key exchange. Suborder representations are a particular kind of weak isogeny representations, i.e. some data to represent isogenies together with an algorithm to efficiently evaluate these isogenies on any point up to a scalar. Prime degree isogenies were not really used before as one cannot write down the isogeny itself (but one can compute its codomain with non-trivial techniques). More recently, a similar type of secret isogeny was used in the SCALLOP scheme [DFFK⁺23]. In SCALLOP, a partial isogeny representation is revealed to the attacker.

From a cryptanalytic point of view, the unlimited amount of torsion information provided by the isogeny representation revealed in pSIDH (and more generally, any isogeny representation) is very interesting. However, when the kernel points are not defined over a small extension, the known algorithms do not apply and it is still unclear how to exploit the isogeny representation to recover the secret isogeny.

Leroux studied the case where a specific isogeny representation (the suborder representation) is revealed, but we can generalize this setting to any isogeny representation. He showed that computing the endomorphism ring of the codomain would make pSIDH insecure, therefore motivating Problem 1.1 in the prime case.

More recently, Robert introduced yet another isogeny representation based on torsion point images and the recent SIDH attacks [Rob22]. This representation could be used (for isogenies with large prime degrees) instead of the suborder representation to derive a key exchange protocol similar to pSIDH, and this protocol would be similarly affected by our new results.

A group action for SIDH and pSIDH In [KMPW21] the authors introduce a group action on a particular set of supersingular elliptic curves. Let E be a supersingular elliptic curve with endomorphism ring isomorphic to O. Then $(O/NO)^*$ acts naturally on the set of cyclic subgroups of E of order N. If there is a one-to-one correspondence between cyclic subgroups and N-isogenous curves, then one can look at this action as acting on a set of curves. This action was used to provide a subexponential quantum key recovery attack on overstretched SIDH parameters.

The reason the attack only works for overstretched parameter sets is that in general this group action is not easy to evaluate (thus substantial amount of extra information on the secret isogeny is needed). This motivates the following problem where the name Malleability Oracle Problem comes from the term introduced in [KMPW21].

Problem 1.3 (Malleability Oracle Problem). Let E be a supersingular elliptic curve and let $\phi : E \to E'$ be a secret isogeny with kernel generated by A. Let $\sigma \in \mathrm{End}(E)$. Find the j-invariant of $E/\langle\sigma(A)\rangle$.

Contributions. Our main result is the following theorem on the resolution of IsERP.

Theorem 1.4. *Let $N = \prod \ell_i^{e_i} \neq p$ be an odd integer that is of size polynomial in p and has $O(\log(\log p))$ divisors. Then there exists a quantum polynomial-time algorithm that solves the IsERP.*

We first provide a reduction from the IsERP to the Powersmooth Quaternion Lifting Problem (PQLP). The PQLP is the problem of finding a powersmooth representative for a given class in $\mathcal{O}/N\mathcal{O}$ for some integer N and maximal order $\mathcal{O}$ in the quaternion algebra $\mathcal{B}_{p,\infty}$.

Our reduction from the IsERP to PQLP is obtained through a quantum equivalence between the IsERP and a problem similar to Problem 1.3, which we call the Group Action Evaluation Problem. The most difficult direction of this equivalence (reducing the IsERP to the Group Action Evaluation Problem) is obtained with a quantum polynomial-time algorithm. The other reduction is classical and uses standard tools for the Deuring correspondence.

The quantum polynomial reduction relies on a special case of the well-known hidden subgroup problem (HSP), namely when the acting group is $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ and the hidden subgroup is a conjugate of the subgroup of upper triangular matrices. This problem was previously studied only for prime N [DMR10] and in this paper we provide a polynomial-time quantum algorithm for any N. Furthermore, whenever N is smooth we propose a classical polynomial-time algorithm which might be of independent interest.

We then propose a classical polynomial-time algorithm for the PQLP. The algorithm relies on several tools developed in KLPT [KLPT14]. Namely we decompose elements $\sigma \in \mathcal{O}$ as $\alpha_1\gamma\alpha_2\gamma\alpha_3$ where the α_i lie in a special subset of $\mathcal{O}$ (linear combinations of j, ij) that can be lifted efficiently to powersmooth elements, and γ is a fixed element of $\mathcal{O}$ of powersmooth norm. Finding γ and lifting the α_i are accomplished with slightly modified subroutines of KLPT, whereas the decomposition itself is inspired by similar decompositions in other contexts [PLQ08]. The lifting algorithm requires that N is odd and has $O(\log\log p)$ prime factors. We look at approaches to generalize this algorithm to arbitrary N (thus solve IsERP for arbitrary degrees) in [CII+23, Appendix D]. We have also implemented this algorithm for prime N in Sagemath [The22], available on GitHub [git23].

The rest of the paper is organized as follows: in Section 2, we introduce some necessary background. Then, in Section 3, we introduce a quantum algorithm to solve the Borel Hidden Subgroup Problem. In Section 4 we define the Group Action Evaluation Problem and the Powersmooth Quaternion Lift Problem (PQLP). We show various reductions between the two problems and the IsERP, most importantly reducing IsERP to the PQLP. In Section 5 we describe our polynomial-time algorithm for PQLP, which leads to a resolution of the IsERP through the reductions. Finally in Section 6, we discuss the impacts of our results on isogeny-based cryptography.

2 Preliminaries

Below, we give a brief introduction to some necessary mathematical background. More details on elliptic curves and isogenies can be found in [Sil09]. The book of John Voight [Voi18] is a good reference regarding quaternion algebras and the Deuring correspondence. In the remaining of this paper, we fix a prime $p > 2$.

2.1 Supersingular elliptic curves and isogenies

Let E_1, E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny is a non-constant rational map from E_1 to E_2 that is simultaneously a group homomorphism. Equivalently, it is a non-constant rational map that sends the point of infinity of E_1 to the point of infinity of E_2. An isogeny induces a field extension $K(E_1)/K(E_2)$ of function fields. An isogeny is called separable, inseparable or purely inseparable if the extension of function field is of the respective type.

The degree of the isogeny is the degree of the field extension $K(E_1)/K(E_2)$. The kernel of an isogeny $\phi : E_1 \to E_2$ is a finite subgroup of E_1. If the isogeny is separable, then the size of the kernel is equal to the degree of the isogeny (more generally, the size of the kernel equals the separable degree of the field extension $K(E_1)/K(E_2)$). For every isogeny $\phi : E_1 \to E_2$ there exists a dual isogeny $\hat{\phi} : E_2 \to E_1$ such that $\deg(\phi) = \deg(\hat{\phi}) = d$ and $\phi \circ \hat{\phi} = [d]_{E_2}$ (and $\hat{\phi} \circ \phi = [d]_{E_1}$). Isogenies (together with the zero map) from E to itself are called endomorphisms. Endomorphisms of an elliptic curve form a ring under addition and composition. An elliptic curve over a finite field is called ordinary if its endomorphism ring is commutative, and supersingular otherwise.

2.2 Quaternion algebras

The endomorphism rings of supersingular elliptic curves over $\mathbb{F}_{p^2}$ are isomorphic to maximal orders of $B_{p,\infty}$, the quaternion algebra ramified at p and ∞. We fix a basis $1, i, j, k$ of $B_{p,\infty}$, satisfying $i^2 = -q$, $j^2 = -p$ and $k = ij = -ji$ for some integer q. The canonical involution of conjugation sends an element $\alpha = a + ib + jc + kd$ to $\overline{\alpha} = a - (ib + jc + kd)$. A *fractional ideal* I in $B_{p,\infty}$ is a $\mathbb{Z}$-lattice of rank four. We denote by $n(I)$ the *norm* of I as the largest rational number such that $n(\alpha) \in n(I)\mathbb{Z}$ for any $\alpha \in I$. An order $\mathcal{O}$ is a subring of $B_{p,\infty}$ that is also a fractional ideal. An order is called *maximal* when it is not contained in any other larger order. The left order of a fractional ideal is defined as $\mathcal{O}_L(I) = \{\alpha \in B_{p,\infty} \mid \alpha I \subset I\}$ and similarly for the right order $\mathcal{O}_R(I)$. Then I is said to be a right $\mathcal{O}_R(I)$-ideal or a left $\mathcal{O}_L(I)$-ideal. A fractional ideal is *integral* if it is contained in its left order, or equivalently in its right order; we refer to integral ideals hereafter as ideals. Eichler orders are the intersection of two maximal orders. If I is an ideal, we can define the Eichler order associated to I as $\mathcal{O}_L(I) \cap \mathcal{O}_R(I)$. In that case, it can be shown that $\mathcal{O}_L(I) \cap \mathcal{O}_R(I) = \mathbb{Z} + I$ (see [DFKL+20]).

2.3 The Deuring correspondence

Fix a supersingular elliptic curve E_0, and an order $\mathcal{O}_0 \simeq \mathrm{End}(E_0)$. The curve/order correspondence allows one to associate to each outgoing isogeny $\varphi : E_0 \to E_1$ an integral left $\mathcal{O}_0$-ideal, and every such ideal arises in this way (see [Koh96] for instance). Through this correspondence, the ring $\mathrm{End}(E_1)$ is isomorphic to the right order of this ideal. This isogeny/ideal correspondence is defined in [Wat69], and in the separable case, it is explicitly given as follows.

Definition 2.1. *Given I an integral left $\mathcal{O}_0$-ideal coprime to p, we define the I-torsion $E_0[I] = \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0$ for all $\alpha \in I\}$. To I, we associate the separable isogeny φ_I of kernel $E_0[I]$. Conversely given a separable isogeny φ, the corresponding ideal is defined as $I_\varphi = \{\alpha \in \mathcal{O}_0 : \alpha(P) = 0$ for all $P \in \ker(\varphi)\}$.*

We summarize properties of the Deuring correspondence in Table 1, borrowed from [DFKL+20].

Supersingular j-invariants over $\mathbb{F}_{p^2}$ $j(E)$ (up to Galois conjugacy)	Maximal orders in $\mathcal{B}_{p,\infty}$ $\mathcal{O} \cong \mathrm{End}(E)$ (up to isomorphism)
(E_1, φ) with $\varphi : E \to E_1$	I_φ integral left $\mathcal{O}$-ideal and right $\mathcal{O}_1$-ideal
$\theta \in \mathrm{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$n(I_\varphi)$

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

2.4 Isogeny representation

In this subsection, we look at isogenies through a more algorithmic prism. Specifically, we consider the following question: what does it mean to "compute" an isogeny? A natural answer is a rational map representation of the isogeny. Other representations are however possible, and in [PL17, Sec 2.4] and [Ler22a] it is argued that any such representation should allow efficient evaluation at arbitrary points (for a more complete study, look at [Ler22b, chapter 4]). More formally, Leroux defines an isogeny representation as some data s_ϕ associated to an isogeny $\phi : E \to E'$ of degree N such that there are two algorithms: one to "verify" and one to "evaluate" ϕ.

The motivation to have a verification algorithm is found in a cryptographic context where an adversary might try to cheat by revealing something that is not a valid isogeny representation. But, in the more cryptanalatic point of view of this paper, we can assume that we work with a valid isogeny representation. This is why we take a relaxed definition of isogeny representation where we only require an evaluation algorithm (a verification algorithm can probably be derived from the evaluation algorithm anyway). Moreover, we assume that the representation is "efficient" meaning that is has polynomial size and the evaluation algorithm is polynomial-time in the log of the degree and the prime. We give a detailed version below. In our context, it is sufficient that the evaluation algorithm gives evaluation of points up to a (common) scalar which is why we qualify our isogeny representation as *weak*.

Definition 2.2. *A weak isogeny representation for the isogeny $\phi : E \to E'$ of degree N, is a data s_ϕ of size $O(polylog(p + N))$ (associated to a unique isogeny ϕ), such that there exists an algorithm $\mathcal{E}$ that takes s_ϕ and a point P of the curve E of order d in input and computes $\lambda(d)\phi(P)$ in $O(polylog(d + N + |P|))$ for any point P of E, where $|P|$ is the bitsize of the representation of P.*

The notion of isogeny representation is particularly relevant when the degree N is a big prime and the kernel points are defined over an $\mathbb{F}_p$-extension of big degree (this is exactly the setting of pSIDH [Ler22a]). Indeed, in that case, the standard ways to represent isogenies (with polynomials, or kernel points) are not compact or efficient enough to match our definition.

The Deuring correspondence gave us the tools to obtain efficient representations with a natural isogeny representation obtained by taking s_ϕ as the ideal I_ϕ corresponding to ϕ. This ideal representation matches Definition 2.2, however it also reveals the endomorphism ring of E'. One of the motivations of Leroux in [Ler22a] to introduce another isogeny representation (called the suborder representation) is to have an isogeny representation that does not directly reveal the endomorphism ring of the codomain. This suborder representation matches our notion of weak isogeny representation as defined in Definition 2.2. The main contribution of this paper implies that the suborder representation does not hide the endomorphism ring of the codomain to a quantum computer, even when the degree is prime.

Since then, Robert [Rob22] suggested to use the techniques introduced to attack SIDH in order to obtain another isogeny representation (this one not even requiring to reveal the endomorphism ring of the domain). Our analysis holds for any suborder representation, hence it also applies to Robert's one.

2.5 The pSIDH key exchange

As an application of the hardness of computing the endomorphism ring from the suborder representation, Leroux introduced a key exchange called pSIDH. The principle can be summarized as follows: use the evaluation algorithm for the suborder representation to perform an SIDH-like key exchange, but for isogenies of big prime degree. The SIDH and pSIDH key exchange both use the following commutative isogeny diagram:

$$\begin{tikzcd} E_B \arrow[r, "\psi_A"] & E_{AB} \\ E_0 \arrow[u, "\phi_B"] \arrow[r, "\phi_A"] & E_A \arrow[u, "\psi_B"'] \end{tikzcd}$$

In pSIDH, Alice and Bob's secret keys are ideal representations for the isogenies ϕ_A and ϕ_B (or equivalently the endomorphism ring of the two curves E_A and E_B), and their associated public keys are the suborder representations for ϕ_A and ϕ_B.

Leroux showed that the knowledge of $\mathrm{End}(E_A)$ (resp. $\mathrm{End}(E_B)$) and the suborder representation of ϕ_B (resp. ϕ_A) was enough to compute the end curve E_{AB} from which the common secret can be derived efficiently. The mechanism behind this computation is quite complicated and is not relevant for us since we target the key recovery problem. We refer to [Ler22a] for more details.

2.6 The hidden subgroup problem

The hidden subgroup problem (HSP for short) in a group G is defined as the problem of finding a subgroup $H \leq G$ given a function f on G satisfying that f is constant on the left cosets of H and takes different values on different cosets, i.e., $f(x) = f(y)$ if and only if $x^{-1}y \in H$. There is also a right version of the hidden subgroup problem where the level sets of the hiding function f are the right cosets of H. As taking inverses in G maps left cosets to right cosets and vice versa, the two versions of HSP are equivalent. (One just needs to replace the hiding function with its composition with taking inverses.) Although the equivalence is straightforward, it is useful as in certain cases it is easier to understand right cosets than left ones (or conversely).

The framework of HSP captures many computational problems including some problems which most cryptographic protocols used today rely on, e.g., factoring and the discrete logarithm problem. Shor's quantum algorithms [Sho97] can solve factoring and the discrete logarithm problem efficiently. Furthermore, quantum polynomial time algorithms for the finite abelian HSP generalizing Shor's algorithm are available, see [Kit95],[BL95].

It is well known that the graph isomorphism problem can be cast as HSP in symmetric. Also, a method solving the HSP in dihedral groups via the standard approach would also solve a special, though still presumably hard special case of the shortest vector problem. However, in contrast to the abelian case, there are only a few positive results known for HSP in finite non-commutative groups. As shown in [EH00], HSP in dihedral groups is related to another problem called the hidden shift problem. The hidden shift problem in a group G is the problem of finding an element $s \in G$ given two functions f_1 and f_2 on G satisfying that $f_1(g) = f_2(gs)$ for every $g \in G$. If f_1 and f_2 are injective then the hidden shift problem in an abelian group G is equivalent to a hidden subgroup problem in the semidirect product $G \rtimes \mathbb{Z}/2\mathbb{Z}$. This is of particular interest in isogeny contexts, as the key recovery problem in CSIDH can be reduced to the injective hidden shift problem in abelian groups in order to produce quantum subexponential-time attacks based on Kuperberg's algorithm [Kup05].

In this paper, we consider a restricted HSP in the general linear group. We use the term Borel hidden subgroup problem for it.

Problem 2.3. Let $N \in \mathbb{Z}_{\geq 1}$ and let $\mathbb{Z}/N\mathbb{Z}$ be the group of integers modulo N. The Borel HSP is the hidden subgroup problem in the general linear group $\mathrm{GL}_n(\mathbb{Z}/N\mathbb{Z})$ for $N \in \mathbb{Z}_{\geq 1}$, i.e., the group of invertible n by n matrices with entries from $\mathbb{Z}/N\mathbb{Z}$, where the hidden subgroup H is promised to be a conjugate of the subgroup consisting of the upper triangular matrices.

Restricting the possible hidden subgroups in non-abelian groups may lead to efficient algorithms to find them. Denney et al. in [DMR10] proposed a polynomial-time algorithm for the Borel HSP in $\mathrm{GL}_2(\mathbb{F}_p)$ for prime numbers p. A quantum algorithm for the more general case of $\mathrm{GL}_n(\mathbb{F}_q)$ over fields of size $q = p^k$, is provided by Ivanyos in [Iva12]. That algorithm runs in polynomial time if q is not much smaller than n.

In this paper, we consider the Borel HSP for $\mathrm{GL}_2(\mathbb{Z}_N)$ for any integer N greater than one, and we present both classical and quantum algorithms for different parameters N. Note that $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ acts as a permutation group on the set of the free cyclic $\mathbb{Z}/N\mathbb{Z}$-submodules of $(\mathbb{Z}/N\mathbb{Z})^2$ and each Borel subgroup H in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ is the stabilizer of a free cyclic $\mathbb{Z}/N\mathbb{Z}$-submodule S of $(\mathbb{Z}/N\mathbb{Z})^2$, thus finding the Borel subgroup is equivalent to finding the corresponding cyclic submodule. The main tool of the classical algorithm is a testing procedure to determine whether elements of $(\mathbb{Z}/N\mathbb{Z})^2$ are in S. The classical algorithm solves the

Borel HSP efficiently for any smooth number N, while the quantum algorithm efficiently solves the Borel HSP for arbitrary N. The main idea of the quantum algorithm is based on the observation that the problem can be reduced to another restricted hidden subgroup problem in the group $G = (\mathbb{Z}/N\mathbb{Z})^2 \rtimes (\mathbb{Z}/N\mathbb{Z})^*$ where the hidden subgroup is promised to be a complement of the normal subgroup $(\mathbb{Z}/N\mathbb{Z})^2$. The latter restricted HSP can be cast as an instance of the multiple shift problem considered in [IPS18], which can itself be seen as a generalization of the hidden shift problem.

Problem 2.4. The hidden multiple shift problem $\mathrm{HMS}(N, n, r)$ is parameterized by three positive integers N, n and r, where $N > 1$ and $2 \leq r \leq N - 1$. Assume that we have a set $H \subseteq \mathbb{Z}/N\mathbb{Z}$ of cardinality r and a function $f_s : (\mathbb{Z}/N\mathbb{Z})^n \times H \to \{0,1\}^l$, defined as $f_s(x, h) = f(x - hs)$ where $s \in (\mathbb{Z}/N\mathbb{Z})^n$ and $f : (\mathbb{Z}/N\mathbb{Z})^n \to \{0,1\}^l$ is an injective function. Given f_s by an oracle, the task is to find $s \bmod \frac{N}{\delta(H,N)}$, where $\delta(H, N)$ is defined as the largest divisor of N such that $h - h'$ is divisible by $\delta(H, N)$ for every $h, h' \in H$.

A special case of the HMS problem was first considered by Childs and van Dam [CVD05]. They presented a quantum polynomial time algorithm for the case when $n = 1$ and H is a contiguous interval of size $N^{\Omega(1)}$. For general n, an algorithm in [IPS18] solves $\mathrm{HMS}(N, n, r)$ in $O(\mathrm{poly}(n)(\frac{N}{r})^{n+O(1)})$. For a set H of small size, HMS is close to the hidden shift problem. Specifically, $\mathrm{HMS}(N, n, 2)$ is the standard hidden shift problem, though modulo a divisor of N depending on the difference of the two elements of H. On the other extreme, for $r = N$, HMS is an abelian hidden subgroup problem in the group $(\mathbb{Z}/N\mathbb{Z})^{n+1}$. Intuitively, the larger r is, the easier $\mathrm{HMS}(N, n, r)$ becomes. Below we restate the above mentioned result from [IPS18] for the special case $n = 1$.

Theorem 2.5. *There is a quantum algorithm that solves the HMS($N, 1, r$) in time $\left(\frac{N}{r}\right)^{O(1)}$ with high probability.*

2.7 The malleability oracle

In [KMPW21] the authors introduce a general framework dubbed the malleability oracle. Let G be a group acting on a set X and let $f : X \to I$ be an injective function where I is some set. The input of the malleability oracle is an element $g \in G$ and a value $f(x)$ (x is not provided) and the output is $f(g * x)$. It is shown in [KMPW21, Theorem 3.3] that if G is abelian and the action of G on X is free and transitive then inverting $f(x)$ can be reduced to an abelian hidden shift problem. The idea of the proof is as follows. One takes an arbitrary known x_0 and the corresponding $f(x_0)$. Then one can define two functions f_0, f_1 from G to I where $f_0(g) = f(g * x_0)$ and $f_1(g) = f(g * x)$. These functions are well-defined as f was injective. Now since the action of G is transitive there is an element s that takes x_0 to x. One can easily see that f and f_0 are shifts of each other and the shift is realized by that element s. Since the action is free, f and f_0 will be injective functions themselves hence one can apply Kuperberg's algorithm to find s and finally that is enough to compute x.

Remark 2.6. It follows from the proof that it is not strictly necessary for the action to be transitive. It is enough if we know any element in the orbit of the secret x. For instance it suffices if there are only a few orbits and we have a representative of each of them (as we can run Kuperberg's algorithm multiple times with different x_0s).

The way to interpret this result is as follows. If one has a way instantiating the malleability oracle, then one can utilize that to invert the function in subexponential time. For isogeny-based cryptography the natural function to be considered here is the one-way function sending a subgroup H to the elliptic curve E/H. In [KMPW21] it is shown that [KMPW21, Theorem 3.3] applies to two scenarios:

- In CSIDH when curves and isogenies are defined over $\mathbb{F}_p$. This result was not novel as it is the same as the original hidden shift attack
- In SIDH when one knows the image of the secret isogeny on a sufficiently large torsion group

We explain the second application a bit further. Let E be a supersingular elliptic curve with known endomorphism ring O. Here we assume that one can evaluate every element of O efficiently on points of E. Let N be any integer. Then O/NO is isomorphic to $M_2(\mathbb{Z}/N\mathbb{Z})$ [Voi18, Theorem 42.1.9]. This implies that $(O/NO)^*$ is isomorphic to $G = \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. Now it is clear that G acts on cyclic subgroups of order N of E by evaluation. When there is a one-to-one correspondence between cyclic subgroups of order N and N-isogenous curves to E, then this implies an action on N-isogenous curves to E. What would a malleability oracle look like in this framework? One is given a curve E' that is N-isogenous to E. Let A be the corresponding secret kernel. Now the input of the oracle is an endomorphism σ (whose degree is coprime to N) and then it returns $E/\sigma(A)$. [KMPW21, Theorem 3.3.] "almost" states that if one has access to such an oracle, then one can compute A via a hidden shift algorithm. The "almost" part comes from the fact that G here is not abelian and the group action is not free. In [KMPW21] it is shown that one can get around this issue by essentially just utilizing a subgroup of G that is abelian (and evoking some small technical conditions).

One can look at this result as a subexponential quantum reduction from finding a certain N-isogeny to being able to instantiate the malleability oracle, which is formulated as Problem 1.3. The results of Section 4 will be related to a generalization of 1.3.

In [KMPW21] the authors were able to solve Problem 1.3 when $\deg(\phi) = 2^k$ and the action of ϕ is known on a sufficiently large subgroup of E. In order to achieve this result one had to throw away most of the available information (by restricting G to a small abelian subgroup) in order to fit the malleability oracle framework. In this paper we show that utilizing the entire G-action improves on [KMPW21] significantly.

The second claim can be reinterpreted in the context of the IsERP problem. Namely when the isogeny degree is a power of 2 and the isogeny is provided with some isogeny representation, then one can compute the endomorphism ring of the codomain in quantum subexponential time (assuming the endomorphism ring of the domain curve was known).

3 The Borel hidden subgroup problem

In this section, we present both classical and quantum algorithms for the "two-dimensional" Borel hidden subgroup problem. The classical algorithm solves the Borel HSP efficiently in the group $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ for smooth number N, while the quantum algorithm solves it efficiently for any positive odd number N. For an even number N we can use a classical procedure applied to the 2-part of N with the quantum one for the odd part of N to obtain a quantum method for every N.

Let N be an integer greater than one. By fixing a basis, we have an explicit isomorphism $\mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2) \cong M_2(\mathbb{Z}/N\mathbb{Z})$ and $\mathrm{Aut}((\mathbb{Z}/N\mathbb{Z})^2) \cong \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. Note that $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ acts as a permutation group on the set of the free cyclic $\mathbb{Z}/N\mathbb{Z}$-submodules of $(\mathbb{Z}/N\mathbb{Z})^2$. Let H be the stabilizer of a secret free cyclic submodule S. In the matrix notation, H is a conjugate of the subgroup consisting of the upper triangular matrices in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$. That is, in an appropriate basis for $(\mathbb{Z}/N\mathbb{Z})^2$, the elements of H are of the form

$$\begin{pmatrix} * & * \\ 0 & * \end{pmatrix},$$

where the diagonal entries are units in $\mathbb{Z}/N\mathbb{Z}$. (Here the first basis element is a generator for S.) The Borel HSP in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ is the following: we are given a function on $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ (given by an oracle) that is constant on the left cosets of H and takes different values on distinct cosets, the task is to find H, or equivalently the submodule S.

Using Chinese remaindering, one can reduce the case when N is any number of known factorization to instances of the prime power case.

Lemma 3.1. *Let $N = N_1N_2$ be a known decomposition of N where $\gcd(N_1, N_2) = 1$. Then we have*

$$\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z}) \cong \mathrm{GL}_2(\mathbb{Z}/N_1\mathbb{Z}) \times \mathrm{GL}_2(\mathbb{Z}/N_2\mathbb{Z}).$$

Moreover, one can reduce the Borel HSP in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ to the Borel HSP in $\mathrm{GL}_2(\mathbb{Z}/N_i\mathbb{Z})$ for $i = 1, 2$.

Proof. By the Chinese Remainder Theorem, $\mathbb{Z}/N\mathbb{Z} \cong \mathbb{Z}/N_1\mathbb{Z} \oplus \mathbb{Z}/N_2\mathbb{Z}$, $(\mathbb{Z}/N\mathbb{Z})^2 \cong (\mathbb{Z}/N_1\mathbb{Z})^2 \oplus (\mathbb{Z}/N_2\mathbb{Z})^2$, $\mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2) \cong \mathrm{End}((\mathbb{Z}/N_1\mathbb{Z})^2) \oplus \mathrm{End}((\mathbb{Z}/N_2\mathbb{Z})^2)$. Furthermore, these isomorphisms can be efficiently computed using the extended Euclidean algorithm. The restriction of the third isomorphism also gives $\mathrm{Aut}((\mathbb{Z}/N\mathbb{Z})^2) \cong \mathrm{Aut}((\mathbb{Z}/N_1\mathbb{Z})^2) \times \mathrm{Aut}((\mathbb{Z}/N_2\mathbb{Z})^2)$. The stabilizer H of the free cyclic submodule S generated by $(A_1, A_2) \in (\mathbb{Z}/N_1\mathbb{Z})^2 \oplus (\mathbb{Z}/N_2\mathbb{Z})^2$ is the direct product of the stabilizers H_i of S_i, where S_i are the free cyclic submodules over $\mathbb{Z}/N_i\mathbb{Z}$ generated by A_i. Hiding functions for H_i can be obtained by restricting the hiding function for H to the component $\mathrm{Aut}((\mathbb{Z}/N_i\mathbb{Z})^2)$. □

3.1 A classical Borel HSP algorithm

Based on iterated applications of Lemma 3.1, we can focus on the prime power case. (Note that the factorization of N can be computed in deterministic time polynomial in $B \log N$ where B is an upper bound on the prime divisors of N.) Therefore, we assume $N = q^k$ for a prime number q.

An important subroutine in our algorithm is a procedure for testing whether an element $u \in (\mathbb{Z}/N\mathbb{Z})^2$ is in S based on the following observations. If $u \in S$ then for any $\varphi \in \mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2)$ such that $\varphi + \mathrm{Id} \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ and $\varphi((\mathbb{Z}/N\mathbb{Z})^2) \leq \mathbb{Z}/N\mathbb{Z}u$ we have $\varphi + \mathrm{Id} \in H$. This is because for $u \in S$ we have $\varphi(u) \in \mathbb{Z}/N\mathbb{Z}u \leq S$ and $\mathrm{Id}(u) = u \in S$. On the other hand, if $u \notin S$ then there exists an element $\varphi \in \mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2)$ with $\varphi(V) \leq \mathbb{Z}/N\mathbb{Z}u$ and $\varphi(S) \not\leq S$. Indeed, if $\{v, w\}$ is an $\mathbb{Z}/N\mathbb{Z}$-basis of $(\mathbb{Z}/N\mathbb{Z})^2$ such that v is a generator of S, then the map sending v to u and w to zero satisfies these properties.

Another ingredient of the testing procedure is the following.

Lemma 3.2. *If $\varphi + \mathrm{Id} \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ then $\varphi(S) \leq S$ if and only if $\varphi + \mathrm{Id} \in H$.*

Proof. If $\varphi(S) \leq S$ then $(\varphi + \mathrm{Id})S \leq \varphi(S) + S = S$. To see the reverse implication, assume that $\varphi(v) \notin S$ for some $v \in S$. Then $\varphi(v) + v$ is in the coset $\varphi(v) + S$ disjoint from S. □

Thus for φ with $\varphi + \mathrm{Id} \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ we can test whether $\varphi(S) \leq S$ by comparing the value of the hiding function taken on $\varphi + \mathrm{Id}$ with that on Id.

Testing procedure: Let w_1, w_2 be a fixed basis of $(\mathbb{Z}/N\mathbb{Z})^2$. Given $u \in (\mathbb{Z}/N\mathbb{Z})^2$ we define two maps φ_1, φ_2 by $\varphi_i(w_i) = u$ and $\varphi_i(w_{3-i}) = 0$. Note that φ_1 and φ_2 generate $E_u := \{\varphi \in \mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2) : \varphi((\mathbb{Z}/N\mathbb{Z})^2) \leq \mathbb{Z}/N\mathbb{Z}u\}$ as an $\mathbb{Z}/N\mathbb{Z}$-submodule of $\mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2)$. Therefore if $\varphi_i(S) \leq S$ $(i = 1, 2)$ then for every element $\varphi \in E_u$ we have $\varphi(S) \leq S$. If $u \in q(\mathbb{Z}/N\mathbb{Z})$ then $\varphi_i - \mathrm{Id} \in q\,\mathrm{End}((\mathbb{Z}/N\mathbb{Z})^2) - \mathrm{Id} \subseteq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ $(i = 1, 2)$, so we can test whether $u \in S$ by testing $\varphi_i(S) \leq S$ $(i = 1, 2)$ by comparing the value of the hiding function taken on $\varphi_i + \mathrm{Id}$ with that on Id. If $q \neq 2$ then either $\varphi_i - \mathrm{Id}$ or $-\varphi_i - \mathrm{Id}$ (or both) fall in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ (depending on the nonzero eigenvalue of φ_i modulo q), so the test above works with a minor modification for $u \notin q(\mathbb{Z}/N\mathbb{Z})$ as well. Finally, to cover the case $q = 2$ and $u \notin q(\mathbb{Z}/N\mathbb{Z})$ observe that $u \in S$ if and only if $S = \mathbb{Z}/N\mathbb{Z}u$. To test whether this is the case we compute generators for the subgroup $H_{\mathbb{Z}/N\mathbb{Z}u} = \{\varphi \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z}) : \varphi(u) \in \mathbb{Z}/N\mathbb{Z}u\}$ and test membership of these generators for membership in H again by comparing values of the hiding function.

Equipped with the testing procedure, we compute S from "bottom up" as follows. First we compute $S \cap q^{k-1}(\mathbb{Z}/N\mathbb{Z})^2$. Note that $q^{k-1}(\mathbb{Z}/N\mathbb{Z})^2 \cong (\mathbb{Z}/q\mathbb{Z})^2$ and there are $q + 1$ possibilities for $S_{k-1} = S \cap q^{k-1}(\mathbb{Z}/N\mathbb{Z})^2$. We can find $S \cap q^{k-1}(\mathbb{Z}/N\mathbb{Z})^2$ by brute force based on the testing procedure on all $q + 1$ submodules corresponding to each possibility in time $q\mathrm{poly}\log|N|$. Assume that we have computed $S_l = S \cap q^l(\mathbb{Z}/N\mathbb{Z})^2$ for some $l > 0$. Then we compute $V_l = \{v \in q^{l-1}(\mathbb{Z}/N\mathbb{Z})^2 : qv \in S_l\}$ and by an exhaustive search in the factor

V_l/S_l we find $S_{l-1} = S \cap q^{l-1}(\mathbb{Z}/N\mathbb{Z})^2$ using again the test in time $q\mathrm{poly}\log|N|$. Note that V_l/S_l is an elementary abelian group of rank at most two. (A factor of a subgroup of an abelian q-group generated by 2 elements is also 2-generated.) The total cost is $kq\mathrm{poly}\log|N|$.

We deduce the following result.

Theorem 3.3. *There is a classical algorithm that solves the Borel hidden subgroup problem in* $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ *in time* $\mathrm{poly}(B\log N)$ *where* B *is an upper bound for the largest prime factor of* N*.*

3.2 A quantum algorithm

Denney, Moore and Russell [DMR10] proposed a quantum polynomial time algorithm that solves the problem in the case when N is a prime. In this subsection we extend their method to arbitrary N as follows. Based on Lemma 3.1 and Theorem 3.3, it is sufficient to give a procedure that works modulo the odd part of N. Thus, in the rest of the discussion we can and do assume that N is odd.

We use the notation introduced at the beginning of the section. In particular, H is the stabilizer of a secret free cyclic $\mathbb{Z}/N\mathbb{Z}$-submodule S of $(\mathbb{Z}/N\mathbb{Z})^2$. Note that $(\mathbb{Z}/N\mathbb{Z})^2/S$ is again a free cyclic $\mathbb{Z}/N\mathbb{Z}$-module whence for $w = (1,0)^T$ or $w = (0,1)^T$ we have that S and w generate $(\mathbb{Z}/N\mathbb{Z})^2$. We describe an algorithm that works under the assumption that S and $w = (1,0)^T$ generate V. If that fails, we repeat it after an appropriate basis change.

From the assumption, it follows that there is a unique element $s \in \mathbb{Z}/N\mathbb{Z}$ such that S is generated by $v = (s,1)^T$. We restrict the hidden subgroup problem to the stabilizer K of the vector w. Note that K is the group consisting of invertible matrices of the form

$$\begin{pmatrix} 1 & * \\ 0 & * \end{pmatrix}.$$

Observe that K is isomorphic to the semidirect $\mathbb{Z}/N\mathbb{Z} \rtimes (\mathbb{Z}/N\mathbb{Z})^*$ where the action of $(\mathbb{Z}/N\mathbb{Z})^*$ on $\mathbb{Z}/N\mathbb{Z}$ is the multiplication by its elements and the hidden subgroup $H \cap K$ is the stabilizer of the free cyclic submodule S in K. Note that the stabilizer of the submodule $(\mathbb{Z}/N\mathbb{Z})v$ in K is the conjugate of the stabilizer of the submodule of $(\mathbb{Z}/N\mathbb{Z})^2$ generated by $(0,1)^T$ in K by the unitriangular matrix

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

transporting $(0,1)^T$ to $v = (s,1)^T$. Hence, by a straightforward calculation, the stabilizer $H \cap K$ of the submodule $\mathbb{Z}/N\mathbb{Z}v$ in K is the subgroup

$$K_s = \left\{ \begin{pmatrix} 1 & hs - s \\ 0 & h \end{pmatrix} : h \in (\mathbb{Z}/N\mathbb{Z})^* \right\}. \tag{1}$$

As N is odd, the images of the matrices of the form $M - \mathrm{Id}$ where $M \in H \cap K$:

$$\left\{ \begin{pmatrix} 0 & (h-1)s \\ 0 & h-1 \end{pmatrix} : h \in (\mathbb{Z}/N\mathbb{Z})^* \right\}$$

generate the submodule $\mathbb{Z}/N\mathbb{Z}v$. Indeed, $2 \in (\mathbb{Z}/N\mathbb{Z})^*$ and hence we have $M = \begin{pmatrix} 1 & s \\ 0 & 2 \end{pmatrix} \in H \cap K$ and so the image of $M - \mathrm{Id}$ is $\mathbb{Z}/N\mathbb{Z}v$.

As shown in the *preprint version* of [IPS18], the HSP in $K \cong \mathbb{Z}/N\mathbb{Z} \rtimes (\mathbb{Z}/N\mathbb{Z})^*$ where the hidden subgroup H is a conjugate of the complement $(\mathbb{Z}/N\mathbb{Z})^*$ can be cast as an instance of the hidden multiple shift problem $HMS(N, 1, r)$ with $r = \phi(N)$, the Euler's totient function of N. To see this, note that from (1) it follows that the *right* cosets of K_s are of the form

$$K_s \cdot \left\{ \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} : h \in (\mathbb{Z}/N\mathbb{Z})^* \right\} = \left\{ \begin{pmatrix} 1 & hs - s + a \\ a & h \end{pmatrix} : h \in (\mathbb{Z}/N\mathbb{Z})^* \right\}.$$

Therefore, when we encode the elements g of K by the second column of $g - \mathrm{Id}$, the right version of the HSP gives an instance of the hidden multiple shift problem on $H \times \{h-1 : h \in (\mathbb{Z}/N\mathbb{Z})^*\}$. As already noted in Section 2.6, the left version of the HSP is equivalent to the right one. Therefore, since $HMS(N, 1, \phi(N))$ can be solved efficiently by Theorem 2.5, we obtain the following result.

Theorem 3.4. *There is a quantum algorithm that efficiently finds the associated free cyclic submodule S for the hidden Borel subgroup H in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ in time $(\log N)^{O(1)}$.*

4 On the isogeny to endomorphism ring problem

In this section, we study the IsERP and its connection to other algorithmic problems. Our final result provides a reduction from the IsERP to a pure quaternion problem, the PQLP (Problem 4.10), but we obtain this reduction through a quantum equivalence between the IsERP and the Group Action Evaluation Problem, that can be seen as a generalization of Problem 1.3. Most of the work in this section is dedicated to this equivalence.

In Section 4.1, we formally introduce the group action we consider. Then, in Section 4.2, we prove the result. Finally, in Section 4.3 we give the link with the PQLP and study the hardness of this problem.

4.1 The group action of $\mathbf{GL_2}(\mathbb{Z}/\boldsymbol{N}\mathbb{Z})$ on $\boldsymbol{N}$-isogenies.

In this section, we cover all necessary results on the group action we will consider. For that, it is important to understand how 2×2 matrices $\bmod N$ appear naturally when you consider the action of endomorphisms on the N-torsion. This comes from the isomorphism $\mathrm{End}(E)/N\,\mathrm{End}(E) \cong M_2(\mathbb{Z}/N\mathbb{Z})$ which is a natural extension of the isomorphism $E[N] \cong \mathbb{Z}/N\mathbb{Z}^2$. We elaborate on that in the next paragraph.

The isomorphism. Let P, Q be a basis of $E[N]$. We identify any point $R = [x]P + [y]Q$ as the vector $v_R = (x, y)^T$. Then, an endomorphism $\sigma \in \mathrm{End}(E)$ can be seen as a matrix in $M_\sigma \in M_2(\mathbb{Z}/N\mathbb{Z})$ through its action on the basis P, Q. If we have $\sigma(P) = [a]P + [b]Q$ and $\sigma(Q) = [c]P + [d]Q$, then we can define M_σ as $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ and the representation of $\sigma(R)$ is given by $M_\sigma v_R$. In that case, it can be easily shown that $\det M_\sigma = \deg \sigma \bmod N$.

If one wants to compute an explicit isomorphism between $\mathrm{End}(E)/N\,\mathrm{End}(E)$ and $M_2(\mathbb{Z}/N\mathbb{Z})$ one can use the above method of evaluating a basis of $\mathrm{End}(E)$ on a basis of $E[N]$. However, when $E[N]$ is defined over a large extension field (e.g., N is large random prime), then this method is not efficient.

The issue can be circumvented by looking at the problem from a slightly different angle. Namely we have basis of $\mathrm{End}(E)/N\,\mathrm{End}(E)$ and we also have a multiplication table of the basis elements. Such a representation is called a structure constant representation. Rónyai [Rón90] proposed a polynomial-time algorithm for this problem when N is prime. The next lemma generalizes the algorithm to arbitrary N whose factorization is known.

Proposition 4.1. *Let A be a ring isomorphic to $M_2(\mathbb{Z}/N\mathbb{Z})$ given by a structure constant representation. Suppose that factorization of N is known. Then there exists a polynomial-time algorithm that computes an explicit isomorphism between A and $M_2(\mathbb{Z}/N\mathbb{Z})$.*

Proof. First we reduce the problem to the case where N is prime power. Let $N = ab$ where a and b are coprime. Then A/aA is isomorphic $M_2(\mathbb{Z}/a\mathbb{Z})$ and A/bA is isomorphic to $M_2(\mathbb{Z}/b\mathbb{Z})$. Since $M_2(\mathbb{Z}/a\mathbb{Z}) \times M_2(\mathbb{Z}/b\mathbb{Z})$ is isomorphic to $M_2(\mathbb{Z}/N\mathbb{Z})$, knowing an explicit isomorphism between A/aA and $M_2(\mathbb{Z}/a\mathbb{Z})$ and an explicit isomorphism between A/bA and $M_2(\mathbb{Z}/b\mathbb{Z})$ is enough to recover the isomorphism between A and $M_2(\mathbb{Z}/N\mathbb{Z})$. Using this procedure iteratively (using the fact that the factorization of N is known) one can reduce to the case where $N = q^k$ where q is some prime number.

Now suppose that A is isomorphic to $M_2(\mathbb{Z}/q^k\mathbb{Z})$. Observe that A/qA is isomorphic to $M_2(\mathbb{Z}/q\mathbb{Z})$. One can compute a non-trivial idempotent in A/qA using Rónyai's algorithm [Rón90], let that be e_0. Now observe that qA is the Jacobson radical of A. Indeed, since qA is clearly contained in the radical and A/qA is semisimple. Then [DK12, Corollary 3.1.2] says that e_0 can be lifted modulo qA to an idempotent of A. Now we know the existence, we show how one can do that algorithmically.

One has that $e_0^2 - e_0 \in qA$ and e_0 and $e_0 - 1$ are not in qA. Our goal is to find an element e which is an idempotent of A. We will perform an iteration which starts with e_0 and in the ith step we return an element e_i for which $e_i^2 - e_i \in q^{i+1}A$. Suppose we have an element e_{i-1} for which $e_{i-1}^2 - e_{i-1} \in q^i A$. Now we are looking for an $f \in A$ such that $(e_{i-1} + fq^i)^2 - (e_{i-1} + fq^i) \in q^{i+1}A$. Let $(e_{i-1}^2 - e_{i-1})/q = E_{i-1}$. Then this is equivalent to $e_{i-1}f + fe_{i-1} \equiv 1 - E_{i-1} \pmod{()q}$. This is a system of linear equations that has a solution by [DK12, Corollary 3.1.2] thus can be found efficiently. Now we have thus found an idempotent e of A.

Since A is a 2×2 matrix ring, $I = Ae$ is an irreducible A-module. Then the left action of A on I provides an explicit isomorphism between A and $M_2(\mathbb{Z}/q\mathbb{Z})$. We could not find a reference for this fact, so we present a quick simple proof.

Let $\begin{pmatrix} a & b \\ c & (1-a) \end{pmatrix}$ be an idempotent matrix. We can assume that it has the above form as e is not congruent to 0 or the identity matrix modulo qA. We also have that $a(1-a) = bc$. Now the following is a generating set (as an abelian group) of Ae:

$$\begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ a & b \end{pmatrix}, \begin{pmatrix} c & (1-a) \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ c & (1-a) \end{pmatrix}.$$

One has that either a or $1-a$ is invertible in $\mathbb{Z}/q^k\mathbb{Z}$, one may suppose that a is invertible (the calculation is the same in the other case). Then it is clear that every element of the form

$$\begin{pmatrix} \alpha & \alpha(b/a) \\ \beta & \beta(b/a) \end{pmatrix}$$

is in the left ideal for any $\alpha, \beta \in \mathbb{Z}/q^k\mathbb{Z}$. We show that every element of Ae is of this form. This follows from the fact that every element of the form

$$\begin{pmatrix} \gamma c & \gamma(1-a) \\ \delta c & \delta(1-a) \end{pmatrix}$$

can be written in this form (a linear combination of the second two basis elements) because if $\gamma c = \alpha$ and $\delta c = \beta$, then $\alpha(b/a) = \gamma(1-a)$ and $\beta(b/a) = \delta(1-a)$ (because $cb/a = (1-a)$). Finally, it is clear that the map $\begin{pmatrix} \alpha & \alpha(b/a) \\ \beta & \beta(b/a) \end{pmatrix} \mapsto (\alpha, \beta)$ is an isomorphism of A-modules. □

Remark 4.2. Once an idempotent e is found, one can finish the proof in an alternate way as well. Namely one can show that $\mathrm{Im}(e) = \ker(e)$ is a cyclic subgroup of $(\mathbb{Z}/q^k\mathbb{Z})^2$ of cardinality q^k. Then a generator of $\ker(e)$ and $\ker(1-e)$ will be a basis in which e is $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ which shows indeed that the left ideal generated by e is minimal.

The group action of invertible matrices on isogenies. Now, let us take a cyclic subgroup $G \subset E[N]$ of order N (this is a submodule of rank 1 inside $(\mathbb{Z}/N\mathbb{Z})^2$ with our isomorphism). If $\sigma \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$, then it is clear that $\sigma(G)$ is also a cyclic subgroup of order N. Thus, we have a natural action of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ on the cyclic subgroups of order N.

This group action on subgroups of order N can naturally be extended to a group action of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ on the set of N-isogenies from E through the bijection between cyclic subgroups of order N and N-isogenies given by $G \mapsto (\phi_G : E \to E/G)$ (and whose inverse is simply $\phi \mapsto \ker\phi$).

Composing this bijection with the group action we already have, we get the following group action

$$M_\sigma \star \phi_G \mapsto \phi_{\sigma(G)}. \tag{2}$$

This action is always well-defined. However, for computational purposes, we want ways to efficiently represent its elements and compute the action $\star$. These considerations motivate the remaining of this paper.

The problem we consider is the following:

Problem 4.3 (Group Action Evaluation Problem). Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ and let $\phi : E \to E_1$ be an isogeny of degree N for some integer N. Given $\mathrm{End}(E)$, an isogeny representation for ϕ, M in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$, find an isogeny representation of $M \star \phi$.

The stabilizer subgroups. One last thing that will be important to apply our results to this group action is to identify the stabilizer subgroup associated to a given isogeny ϕ. In fact, those are pretty easy to identify and are well-known objects. The answer is given by the following proposition.

Proposition 4.4. *Let $\phi : E \to E'$ be an isogeny of degree N. The stabilizer subgroup associated to ϕ through the group action defined in Equation (2) is made of the matrices M_σ such that σ is in the Eichler order $\mathbb{Z} + I_\phi$ where I_ϕ is the ideal associated to ϕ under the Deuring correspondence.*

Proof. By definition of the group action, the stabilizer subgroup is obtained with the matrices M_σ such that $\sigma(\ker\phi) = \ker\phi$. This means that σ acts as a scalar λ_σ on $\ker\phi$. Thus, $\ker\phi \subset \ker(\sigma - \lambda_\sigma)$ and by definition of I_ϕ, we have that $\sigma - \lambda_\sigma \in I_\phi$, hence $\sigma \in \mathbb{Z} + I_\phi$. Conversely, it is clear that any element in $\mathbb{Z} + I_\phi$ acts as a scalar on $\ker\phi$ and so is part of the stabilizer. For the proof that $\mathbb{Z}+I_\phi$ is an Eichler order, see [DFKL+20]. □

Remark 4.5. Writing the stabilizer subgroups as Eichler orders of the form $\mathbb{Z}+I_\phi$ will help us prove that computing the stabilizer subgroup is essentially equivalent to finding the endomorphism ring of the codomain of ϕ (which is isomorphic to the right order of I_ϕ).

Proposition 4.6. *The stabilizer subgroups are conjugates of the subgroup of upper triangular matrices (i.e., a Borel subgroup).*

Proof. Follows from [Voi18, 23.1.3]. For an elementary proof see [CII+23, Appendix B]. □

4.2 The main reductions.

In this section, we prove a quantum polynomial-time equivalence between the the Group Action Evaluation Problem and the IsERP.

Theorem 4.7. *The Group Action Evaluation Problem reduces to the IsERP in classical polynomial-time.*

Proof. Assume we have an efficient algorithm to solve the IsERP. Let us take an instance of the Group Action Evaluation Problem. So we have $N, E, \mathrm{End}(E)$, a representation for ϕ and a matrix M, and we want to compute a representation for $M * \phi$.

The first step of the reduction is to compute the ideal I_ϕ associated to ϕ. There are several ways to do that, but to keep this proof short, we will use some of the results proven in [Ler22a]. Thus, our first step is to build a suborder representation for the isogeny ϕ as in [Ler22a]. The suborder representation is made of endomorphisms of $\mathbb{Z} + N\,\mathrm{End}(E) \hookrightarrow \mathrm{End}(E')$ of powersmooth norm. Since we know $\mathrm{End}(E)$ and $\mathrm{End}(E')$, the algorithms of the Deuring correspondence can be used to compute their kernels in polynomial time (their norm being powersmooth implies that their kernels are defined over a small extension). Then, we can compute the suborder representation using Vélu's formulas. Once we have the suborder representation, we can apply the equivalence between the SOIP and the SOERP [Ler22a, Proposition 13] to find the ideal I_ϕ. Once I_ϕ has been computed, we need to compute the ideal $I_{M\star\phi}$. For that we are going to use a σ such that $M_\sigma = M$. We can build such a σ in polynomial time from $\mathrm{End}(E)$ using Proposition 4.1.

Once a good σ is known, we get the ideal $I_{M\star\phi}$ as $\sigma(I_\phi \cap \mathcal{O}\sigma)\sigma^{-1} + N\mathcal{O}$ (where we take $\mathcal{O} \cong \mathrm{End}(E)$). Since the ideal $I_{M_\sigma\star\phi}$ is a valid isogeny representation, this proves the result. □

Theorem 4.8. *The IsERP reduces to the Group Action Evaluation Problem in quantum polynomial time.*

Proof. Assume we can solve the Group Action Evaluation Problem.

Let us take an input of the IsERP, we have a curve E, its endomorphism ring $\mathrm{End}(E)$, an integer N and the isogeny representation associated to an isogeny ϕ of degree N.

The algorithm to solve the Group Action Evaluation Problem allows us to compute efficiently the group action introduced in Section 4.1. Using Proposition 4.6 and Theorem 3.4 one can compute the stabilizer subgroup associated to ϕ. As the stabilizer subgroups of ϕ give us matrices corresponding to some σ in the Eichler order $\mathbb{Z}+I_\phi$, we can compute the embedding of this order in $\mathcal{O} \cong \mathrm{End}(E)$ in polynomial time using the algorithm of Proposition 4.1. Then, we can extract the ideal I_ϕ and compute $O_R(I_\phi)$ which is isomorphic to $\mathrm{End}(E')$ and this gives the result.

□

When the degree N is smooth, we can modify the proof of Theorem 4.8 to get a classical reduction by using Theorem 3.3 instead of Theorem 3.4.

Theorem 4.9. *Suppose that degree of the secret isogeny is smooth. Then IsERP reduces to the Group Action Evaluation Problem in classical polynomial time.*

4.3 Reduction of the Group Action Evaluation Problem to the PQLP.

In this section, we reduce the Group Action Evaluation Problem to another problem that we call the Powersmooth Quaternion Lift Problem (PQLP). The PQLP can be stated as follows:

Problem 4.10. Let $\mathcal{O}$ be a maximal order in $\mathcal{B}_{p,\infty}$. Given an integer N and an element $\sigma_0 \in \mathcal{O}$ such that $(n(\sigma_0), N) = 1$, find $\sigma = \lambda\sigma_0 \bmod N\mathcal{O}$ of powersmooth norm with some λ coprime to N.

We use $\mathsf{PQLP}_{\mathcal{O}}(\sigma_0)$ to denote the set of $\sigma \in \mathcal{O}$ that satisfy the conditions in Problem 4.10 with respect to $\sigma_0 \in \mathcal{O}$. The high level idea of the reduction from the PQLP to the Group Action Evaluation Problem is close to the approach introduced in [KMPW21]. Given a matrix M, the goal is to find a good representative of the class of M, i.e. a $\sigma \in \mathrm{End}(E)$ of powersmooth norm, such that $M = M_\sigma$. Then, we can use Vélu's formulae to solve the Group Action Evaluation Problem.

Proposition 4.11. *The Group Action Evaluation Problem reduces to PQLP in classical polynomial time.*

Proof. Let us take an instance of our problem. We have $N, E, \mathrm{End}(E)$, an isogeny representation for $\phi : E \to E'$ of degree N and a matrix M.

We need to show that if we know a $\sigma \in \mathrm{End}(E)$ (represented as a quaternion element in a maximal order $\mathcal{O} \cong \mathrm{End}(E)$) of powersmooth norm such that $M_\sigma = M$, then we can compute a representation for $M \star \phi$ in polynomial time. For that, we will use the following commutative isogeny diagram

$$\begin{tikzcd}
E' \arrow[r, "\sigma'"] & E'/\sigma(\ker\phi) \\
E \arrow[u, "\phi"] \arrow[r, "\sigma"] & E \arrow[u, "M\star\phi"']
\end{tikzcd}$$

where σ' has the same degree as σ and is defined by $\ker \sigma' = \phi(\ker \sigma)$. Since the isogeny σ' has powersmooth degree, it can be computed in polynomial time once $\ker \sigma'$ has been computed. Since, we can evaluate ϕ, it suffices to compute $\ker \sigma$ and this can be done in polynomial-time since we known $\mathrm{End}(E)$.

Since the diagram is commutative, we have that $\sigma' \circ \phi = M \star \phi \circ \sigma$ and this gives us the way to evaluate efficiently $M \star \phi$ on almost all torsion (as soon as the order is coprime to $\deg \sigma$) as $M \star \phi = \hat{\sigma}' \circ \phi \circ \sigma / \deg \sigma$. This is sufficient to build a suborder representation of $M \star \phi$ (see the algorithm outlined in the proof of Theorem 4.8).

This proves that the main computational task is to find this σ of powersmooth norm. Thus, it suffices to apply an algorithm to solve the PQLP on input N, $\mathrm{End}(E)$ and a σ_0 such that $M_{\sigma_0} = M$ (that we can find using Proposition 4.1). $\square$

5 Resolution of the PQLP

In this section, we solve the PQLP (Problem 4.10) with conditions imposed on the factors of N, as detailed in the following theorem.

Theorem 5.1. *Let $N = \prod \ell_i^{e_i} \neq p$ be an odd integer that is of size polynomial in p and has $O(\log(\log p))$ factors, then there exists a randomized classical polynomial time algorithm that solves the PQLP.*

This theorem follows from the correctness of Algorithm 3 which is introduced and discussed in Section 5.2. The successive reductions from the IsERP to the Group Action Evaluation Problem, and subsequently to the PQLP, demonstrate the existence of a polynomial time algorithm that solves the IsERP, given N satisfies the conditions in Theorem 5.1. As a direct consequence, this breaks pSIDH quantumly since N is a large prime in pSIDH. As mentioned previously, the IsERP is easy when N is powersmooth. We briefly discuss some approaches to solve the general case in [CII+23, Appendix D].

In Section 5.1, we give a summary of useful known algorithms and provide variants for RepresentInteger, StrongApproximation and KLPT to better accommodate our specific application. Following this, we introduce our primary strategy for resolving the PQLP in Section 5.2. In Section 5.3, we deal with a critical technical point which we introduce as the Quaternion Decomposition problem. The crux of this problem, and indeed our main conceptual contribution, is the decomposition of σ_0 into elements that are easy to lift, and elements already possessing a powersmooth norm. Finally in Section 5.4, we provide a quantum algorithm that solves the PQLP.

5.1 Algorithmic building blocks

Our algorithm for the PQLP is founded on algorithmic building blocks initially introduced in [KLPT14] and later extended in other work, such as [DFKL+20]. We provide a brief recap of these algorithms here, along with several new variants tailored to suit our requirements. We fix $\log^c p$ to be our powersmooth bound for some constant c, and this bound is inherently implied whenever we reference the term 'powersmooth'.

As in [KLPT14], for each p, we fix a special p-extremal maximal order $\mathcal{O}_0$.

$$\mathcal{O}_0 = \begin{cases} \mathbb{Z}\langle i, \frac{1+j}{2}\rangle \text{ where } i^2 = -1, j^2 = -p, & \text{for } p \equiv 3 \bmod 4, \\ \mathbb{Z}\langle \frac{1+i}{2}, j, \frac{ci+k}{q}\rangle \text{ where } i^2 = -q, j^2 = -p, & \text{for } p \equiv 1 \bmod 4, \end{cases} \tag{3}$$

where c is any root of $x^2 + p \bmod q$. In the second case, q is required to satisfy that $q \equiv 3 \bmod 4$ is a prime and $\left(\frac{-p}{q}\right) = 1$. We add one extra condition that $(q, N) = 1$. Under the generalized Riemann hypothesis (GRH), the smallest q is of size $O(\log^2 p)$. For the ease of exposition, we define q to be 1 when we are in the first case (i.e., when $p \equiv 3 \bmod 4$).

For each $\mathcal{O}_0$, we identify a suborder of the form $R+Rj$ for $R = \mathbb{Z}[i]$ (note that we are making a slightly different choice here than the R in [KLPT14] where they always take R to the maximal order in $\mathbb{Q}(i)$). For an element $\alpha = a + bi \in R$, we use $\mathsf{Re}_R(\alpha)$ to denote a and $\mathsf{Im}_R(\alpha)$ to denote b. Let D denote the index $[\mathcal{O}_0 : R + Rj]$, then

$$D = \begin{cases} 4, & \text{for } p \equiv 3 \bmod 4, \\ 4q, & \text{for } p \equiv 1 \bmod 4. \end{cases} \tag{4}$$

We will now detail the algorithmic building blocks, sourced from [KLPT14] or [DFKL+20].

- $\mathsf{Cornacchia}(M)$: on an input M that is a prime integer not equal to q, outputs either $\perp$ if M cannot be represented by $x^2 + qy^2$, or a solution x, y to the equation $M = x^2 + qy^2$.
- $\mathsf{RepresentInteger}_{\mathcal{O}_0}(M)$: on an input $M > p$, outputs $\gamma \in \mathcal{O}_0$ such that $n(\gamma) = M$.
- $\mathsf{StrongApproximation}_F(N, \mu_0)$: on inputs an integer $F > pN^4$, a prime N and $\mu_0 \in Rj$, outputs $\lambda \notin N\mathbb{Z}$ and $\mu \in \mathcal{O}_0$ of norm dividing F such that $\mu = \lambda\mu_0 \bmod N\mathcal{O}_0$.
- $\mathsf{KLTP}_M(I)$: on inputs an integer $M > p^3$ and an ideal I, outputs an equivalent ideal J such that $n(J) = M$.

Let us denote the output γ of $\mathsf{RepresentInteger}_{\mathcal{O}_0}(M)$ as $C + Dj$ with $C, D \in R$. To fit our algorithm's specific use case, we require not only that $n(\gamma)$ is powersmooth, but also that C, D satisfy additional conditions relative to some inputs $A, B \in R$, which are determined by σ_0 from the PQLP. As a result, we introduce the following variant named $\mathsf{RepresentInteger'}_{R+Rj}(N, A, B)$. This variant necessitates more randomized steps to find the desired outputs $C, D \in R$.

Algorithm 1: $\mathsf{RepresentInteger'}_{R+Rj}(N, A, B)$

Input: An integer N and $A, B \in R$
Output: $C, D \in R$ such that: i) $(n(C + Dj), N) = 1$ and $n(C + Dj)$ is powersmooth; ii) $(n(C), N) = 1$; iii) $(n(D), N) = 1$; iv) $(N, \mathsf{Im}_R(A\bar{B}C\bar{D})) = 1$ and v) $q(4p^2n(ABCD) - (n(AC) - pn(AD))^2)$ is a square modulo N.

1 Let k be the smallest integer such that the number of prime factors of N is less than $k \log\log p$, randomly generate M of size $\lfloor p \log^{3k+4} p \log\log p \rfloor$ that is powersmooth and coprime to N.
2 Set $m = \lfloor \sqrt{\frac{M}{p(q+1)}} \rfloor$ and sample random integers $z, t \in [-m, m]^2$.
3 Set $M' = M - p(z^2 + qt^2)$.
4 **If** $(z^2 + qt^2, N) \neq 1$ *or* $(M', N) \neq 1$ *or* M' *is not a prime* **then**
5 Go back to Step 2.
6 **If** $\mathsf{Cornacchia}(M') = \perp$ **then**
7 Go back to Step 2.
8 $x, y = \mathsf{Cornacchia}(M')$.
9 $C \leftarrow x + yi, D \leftarrow z + ti$.
10 **If** $(N, \mathsf{Im}_R(A\bar{B}C\bar{D})) \neq 1$ **then**
11 Go back to Step 2.
12 **If** $q(4p^2n(ABCD) - (n(AC) - pn(AD))^2)$ *is not a square modulo* N **then**
13 Go back to Step 2.
14 **Return** C, D.

Heuristic 5.2. *We assume that M', z^2+qt^2, $\mathsf{Im}_R(A\bar{B}C\bar{D})$ and $q(4p^2n(ABCD)-(n(AC)-pn(AD))^2)$ appearing in Algorithm 1 behave like random integers of the same size.*

Lemma 5.3. *Let N be as in Theorem 5.1, assuming Heuristic 5.2, Algorithm 1 returns a solution in polynomial time.*

Proof. For Step 4, by assumption, M' is a prime with probability $1/O(\log p)$, and once we ensure that M' is a prime, $(M', N) = 1$ holds with a negligible failure rate. On the other hand, since the number of prime factors of N is less than $k \log\log p$, the probability that $(z^2 + qt^2, N) = 1$ holds is at least $1/O(\log^k p)$. Therefore, Step 4 succeeds with probability greater than $1/O(\log^{k+1} p)$. For Step 10, similarly it succeeds with probability at least $1/O(\log^k p)$. For Step 6, $x^2 + qy^2 = M'$ has a solution if and only if (M') is a product of two principal ideals in $\mathbb{Z}[i]$. M' is split or ramifies in $\mathbb{Z}[i]$ with probability $1/2$, and a random invertible ideal in $\mathbb{Z}[i]$ is principal with probability $1/\#\mathcal{C}\ell(\mathbb{Z}[i]) > 1/O(\sqrt{q}\log q)$ [Coh95, Section 5.10.1]. Since q is at most $O(\log^2 p)$, this step succeeds with probability greater than $1/O(\log p \log\log p)$. Finally, for an integer to be a square modulo N, it's equivalent to this integer being a square modulo each prime factor of N. Therefore, Step 12 succeeds with probability at least $1/O(\log^k p)$. In total, the success probability is greater than $1/O(\log^{2+3k} p \log\log p)$. In Step 2, there will be $O(\log^{3k+2} p \log\log p)$ pairs of (z, t), therefore Algorithm 1 will return a solution in polynomial time. □

We also provide a generalization of the $\mathsf{StrongApproximation}$ algorithm to allow for composite N (note that in this algorithm we don't have restriction on the number of factors of N). The subscript $_{ps}$ refers to powersmooth.

Heuristic 5.4. *We assume that M appearing in Algorithm 2 behaves like a random integer of the same size.*

Lemma 5.5. *Consider a linear equation $N_1x+N_2y = N_3 \bmod N$ where $\gcd(N, N_1) = d_1, \gcd(N, N_2) = d_2$ and $(d_1, d_2) = 1$. Then this equation has N solutions in $(\mathbb{Z}/N\mathbb{Z})^2$.*

Proof. This can be seen by checking how many solutions there are for equation $N_1x + N_2y = N_3 \bmod \ell_i^{e_i}$ with $\ell_i^{e_i}$ being a prime power divisor of N and then using Chinese Reminder Theorem. □

Lemma 5.6. *Let N be an odd integer, assuming Heuristic 5.4, Algorithm 2 returns a solution in polynomial time.*

Proof. The $\#S \times \#S$ linear equations behave like a random system of linear equations of the same dimension, therefore, repeating Step 3 constant number of times will give rise to a linear system over $\mathbf{F}_2$ that is solvable. We have ensured that F generated in Step 8 is such that $\frac{F}{n(\mu_0)}$ is a square modulo N, hence Step 9 makes sense. Similar to what is discussed in the proof of Lemma 5.3, Step 12 succeeds with probability at least $1/O(\log^2 p \log\log p)$. To make sure this

Algorithm 2: $\mathsf{StrongApproximation}_{ps}(N, \mu_0)$

Input: An odd integer N and $\mu_0 \in Rj$ such that $(n(\mu_0), N) = 1$.
Output: $\lambda \in \mathbb{Z}$ such that $(\lambda, N) = 1$ and $\mu \in R$ with powersmooth norm such that $\mu = \lambda\mu_0 \bmod N\mathcal{O}_0$.

1 Write μ_0 as $(t + si)j$ with $t, s \in \mathbb{Z}$.
2 Let $S = \{\ell$ such that $\ell \mid N$ and $\left(\frac{n(\mu_0)}{\ell}\right) = -1\}$.
3 Randomly generate $\#S$ many prime p_i's such that $p_i < \log^c p$ and let $\epsilon_{ij} \in \{0, 1\}$ be the exponent such that $\left(\frac{p_i}{\ell_j}\right) = (-1)^{\epsilon_{ij}}$.
4 Solve the system of $\#S \times \#S$ equations $\Sigma_{i=1}^{\#S}\epsilon_{ij}x_i = 1$ for $j = 1, \cdots, \#S$ in $\mathbf{F}_2$.
5 **If** *There is no solution found in Step 4* **then**
6 Go back to Step 3
7 $F = \prod_{i=1}^{\#S} p_i^{x_i}$.
8 Multiply a $\log^c p$-powersmooth square factor that is coprime to $n(\mu_0)$ to F to ensure $F > pN^4$.
9 Denote one of the square root of $\frac{F}{n(\mu_0)}$ modulo N by λ.
10 Randomly generate c, d such that $\lambda p(2tc + 2qsd) \equiv (F - \lambda^2 p(t^2 + qs^2))/N \bmod N$.
11 Set $M = (F - p((\lambda t + cN)^2 + q(\lambda s + dN)^2))/N^2$.
12 **If** *M is not a prime or* $\mathsf{Cornacchia}(M) = \perp$ **then**
13 Go back to Step 10
14 $a, b = \mathsf{Cornacchia}(M)$.
15 **Return** $\lambda\mu_0 + N(a + bi + (c + di)j), \lambda$.

algorithm gets passed to Step 14, we need $O(\log^2 p \log\log p)$ many solutions from Step 10. According to Lemma 5.5, this happens if $O(\log^2 p \log\log p) < N$, and this holds since N is assumed to be of size polynomial in p. □

Finally, another variant we introduce here is $\mathsf{KLTP}'_M(\mathcal{O}_0, \mathcal{O})$. Here M is an integer such that $M > p^3$. This algorithm first computes a connecting ideal I' from $\mathcal{O}_0$ to $\mathcal{O}$, then computes an equivalent left $\mathcal{O}_0$-ideal I of norm M whose right order is $\alpha^{-1}\mathcal{O}\alpha$ for some nonzero $\alpha \in \mathcal{B}_{p,\infty}$ using $\mathsf{KLPT}_M(I')$. This KLPT' algorithm outputs I and α.

5.2 The main algorithm

In this section, we first present a strategy that solves Problem 4.10 for special orders $\mathcal{O}_0$. Then, we expand this strategy to address more general orders $\mathcal{O}$. Let $\sigma_0 \in \mathcal{O}_0$ be as in Problem 4.10, the method proceeds as follows.

1. Find $\sigma_0' \in R + Rj$ such that $\sigma_0' = \sigma_0 \bmod N\mathcal{O}_0$. Since $[\mathcal{O}_0 : R + Rj] = D$, $D\sigma_0 \in R + Rj$. Let $D' \in \mathbb{Z}$ be such that $D'D \equiv 1 \bmod N$, such D' exists since $(D, N) = 1$, then $\sigma_0' = D'D\sigma_0 \in R + Rj$ and $\sigma_0' = \sigma_0 \bmod N\mathcal{O}_0$. By an abuse of notation, we will use σ_0 to denote σ_0' in what follows.
2. Write $\sigma_0 = A+Bj$ with $A, B \in R$, let γ be the output of $\mathsf{RepresentInteger}'_{R+Rj}(N, A, B)$. Intuitively, γ is an element in $R+Rj$ that has powermooth norm and satisfies extra properties to ensure the next step has a solution.
3. Find $\alpha_1, \alpha_2, \alpha_3 \in Rj$ such that $\sigma_0 = \alpha_1\gamma\alpha_2\gamma\alpha_3 \bmod N\mathcal{O}_0$. This is the main technical point in this method, we introduce it as Problem 5.7 and discuss it in detail in Section 5.3.
4. Find $\gamma_i \in \mathcal{O}_0$ such that $\gamma_i = \lambda_i\alpha_i \bmod N\mathcal{O}_0$, $n(\gamma_i)$ is powersmooth and $(\lambda_i, N) = 1$ for $i = 1, 2, 3$. These are exactly the outputs of $\mathsf{StrongApproximation}_{ps}(N, \alpha_i)$.
5. The element $\gamma_1\gamma\gamma_2\gamma\gamma_3 \in \mathcal{O}_0$ satisfies that $\sigma_0 = \lambda\gamma_1\gamma\gamma_2\gamma\gamma_3 \bmod N\mathcal{O}_0$ with $n(\gamma_1\gamma\gamma_2\gamma\gamma_3)$ powersmooth and λ coprime to N.

In general, let $\mathcal{O}$ be a maximal order in $\mathcal{B}_{p,\infty}$, and let $n_I > p^3$ be a random integer that is coprime to N. Let $n_I' \in \mathbb{Z}$ such that $n_I'n_I \equiv 1 \bmod N$. Let I, α be the outputs of $\mathsf{KLPT}_{n_I}(\mathcal{O}_0, \mathcal{O})$ such that I is a connecting ideal from $\mathcal{O}_0$ to $\mathcal{O}' := \alpha^{-1}\mathcal{O}\alpha$ and $n(I) = n_I$. We then have inclusions $n_I\mathcal{O}' \subseteq \mathcal{O}_0$ and $n_I\mathcal{O}_0 \subseteq \mathcal{O}'$, therefore $n_I\alpha^{-1}\sigma_0\alpha \in \mathcal{O}_0$. Let $\sigma \in \mathsf{PQLP}_{\mathcal{O}_0}(n_I\alpha^{-1}\sigma_0\alpha)$, then $\sigma = n_I\alpha^{-1}\sigma_0\alpha \bmod N\mathcal{O}_0$. Multiplying n_I with both sides of the equation yields that $n_I\sigma = n_I^2\alpha^{-1}\sigma_0\alpha \bmod N\mathcal{O}'$. Multiplying $n_I'^2$ on both sides gives that $n_I'\sigma = \alpha^{-1}\sigma_0\alpha \bmod N\mathcal{O}'$. Since n_I' is coprime to N, $\sigma \in \mathsf{PQLP}_{\mathcal{O}'}(\alpha^{-1}\sigma_0\alpha)$, therefore $\alpha\sigma\alpha^{-1} \in \mathsf{PQLP}_{\mathcal{O}}(\sigma_0)$.

We summarize the discussions above and present Algorithm 3.

5.3 Quaternion decomposition

In this section, we discuss how to perform Step 3 from the strategy outline. We start with introducing a new problem.

Algorithm 3: $\mathsf{PQLP}_{\mathcal{O}}(N, \sigma_0)$

Input: An odd integer N that is of size polynomial in p and has $O(\log\log p)$ distinct prime factors, a maximal order $\mathcal{O}$, and an element $\sigma_0 \in \mathcal{O}$ such that $(n(\sigma_0), N) = 1$.

Output: $\sigma \in \mathsf{PQLP}_{\mathcal{O}}(\sigma_0)$.

Compute D' such that $D'D \equiv 1 \bmod N$.
$\sigma_0 \leftarrow D'D\sigma_0$
Write σ_0 as $A + Bj$ with $A, B \in R$
$\gamma \leftarrow \mathsf{RepresentInteger'}_{R+Rj}(N, A, B)$
$\alpha_1, \alpha_2, \alpha_3 \leftarrow \mathsf{QuaternionDecomposition}(\sigma_0, \gamma, N)$(Algorithm 4)
$\lambda_i, \gamma_i \leftarrow \mathsf{StrongApproximation}_{ps}(N, \alpha_i)$ for $i = 1, 2, 3$
$\sigma \leftarrow \gamma_1\gamma\gamma_2\gamma\gamma_3$
Randomly generate $n_I > p^3$ that is coprime to N.
$I, \alpha \leftarrow \mathsf{KLPT}_{n_I}(\mathcal{O}_0, \mathcal{O})$
Return $\alpha\gamma_1\gamma\gamma_2\gamma\gamma_3\alpha^{-1}$.

Problem 5.7 (Quaternion Decomposition). Let N be an odd integer, and $\mathcal{O}_0, R$ be as defined in Section 5.1. Let $\sigma_0 = A+Bj, \gamma = C+Dj \in R+Rj$ be such that: i) $(n(\gamma), N) = 1$ and $n(\gamma)$ is powersmooth; ii) $(n(C), N) = 1$; iii) $(n(D), N) = 1$; iv) $(N, \mathsf{Im}_R(A\bar{B}C\bar{D}))$ and v) $q(4p^2 n(ABCD) - (n(AC) - pn(AD))^2)$ is a square modulo N. Find $\alpha_1, \alpha_2, \alpha_3 \in Rj$ such that $\sigma_0 = \alpha_1\gamma\alpha_2\gamma\alpha_3 \bmod N\mathcal{O}_0$.

Suppose one could find $\alpha_1, \alpha_2, \alpha_3 \in Rj$ such that

$$\sigma_0\bar{\alpha}_3\bar{\gamma} = \alpha_1\gamma\alpha_2 \bmod N\mathcal{O}_0, \tag{5}$$

and $(n(\alpha_3), N) = 1$, then

$$\sigma_0 = n'_{\alpha_3} n'_{\gamma} \alpha_1\gamma\alpha_2\gamma\alpha_3.$$

Here n'_{α_3} and n'_{γ} are integers such that $n'_{\alpha_3} n(\alpha_3) \equiv 1 \bmod N$ and $n'_{\gamma} n(\gamma) \equiv 1 \bmod N$ respectively. We then search for solutions $\alpha_1, \alpha_2, \alpha_3 \in Rj$ for Equation (5) with $(n(\alpha_3), 1) = 1$ instead.

Let us write α_i's as $x_i j$ with x_i's being unknowns that we wish to find in R, writing Equation (5) in terms of A, B, C, D, x_1, x_2 and x_3 gives rise to the following:

$$\begin{aligned}
\text{Equation (5)} &\iff (A + Bj)(-j)\bar{x}_3(\bar{C} - j\bar{D}) = x_1 j(C + Dj)x_2 j \bmod N\mathcal{O}_0 \\
&\iff (-Ax_3 j + pB\bar{x}_3)(\bar{C} - j\bar{D}) = (x_1\bar{C}j - px_1\bar{D})x_2 j \bmod N\mathcal{O}_0 \\
&\iff (-pA\bar{D}x_3 + pB\bar{C}\bar{x}_3) + (-ACx_3 - pBD\bar{x}_3)j = -p\bar{C}x_1\bar{x}_2 - p\bar{D}x_1x_2 j \bmod N\mathcal{O}_0.
\end{aligned}$$

Therefore, in order to solve the original equation, it suffices to find $x_1, x_2, x_3 \in R$ with $(n(x_3), N) = 1$ such that

$$\begin{cases} pA\bar{D}x_3 - pB\bar{C}\bar{x}_3 = p\bar{C}x_1\bar{x}_2 \bmod NR, \\ ACx_3 + pBD\bar{x}_3 = p\bar{D}x_1x_2 \bmod NR. \end{cases} \tag{6}$$

Note that the modulo condition in Equation (6) holds not just in $N\mathcal{O}_0$ but in NR since $[\mathcal{O}_K : R] = 1$ or 2 is coprime to N. By assumption, $n(C)$ and $n(D)$ are both coprime to N, let $n'_C, n'_D \in \mathbb{Z}$ be integers such that $n'_C n(C) \equiv 1 \bmod N$ and $n'_D n(D) \equiv 1 \bmod N$ respectively, and let $p' \in \mathbb{Z}$ be such that $p'p \equiv 1 \bmod N$, then Equation (6) is equivalent to

$$\begin{cases} (n'_C p')(pA\bar{D}x_3 - pB\bar{C}\bar{x}_3) = x_1\bar{x}_2 \bmod NR, \\ (n'_D p')(ACx_3 + pBD\bar{x}_3) = x_1 x_2 \bmod NR. \end{cases} \tag{7}$$

Lemma 5.8. *Equation (7) has a solution $(x_1, x_2, x_3) \in R^3$ if and only if there exists $x_3 \in R$ such that $x := (n'_C p')(pA\bar{D}x_3 - pB\bar{C}\bar{x}_3)$ and $y := (n'_D p')(ACx_3 + pBD\bar{x}_3)$ have same norm modulo N.*

Proof. One solution to Equation (7) clearly implies $n(x) = n(y) = n(x_1)n(x_2)$. For the other direction, we provide a simple proof here for the case when N is a prime that is inert in R, and refer to [CII+23, Appendix C] for the general case when N is an arbitrary odd integer. Since N is a prime that is inert in R, $R/(N) \cong \mathbb{F}_{N^2}$. Hilbert's Theorem 90 implies that if $a \in R/(N)$ has norm 1, then $a = b/\bar{b}$ for $b \in R/(N)$. Since $n(x) = n(y)$ and both $x, y \notin NR$, we have that $n(x/y) = 1$, therefore $x/y = z/\bar{z}$ for some nonzero $z \in R/(N)$. Consequently, x_1, x_2 can be chosen to be lifts of yz and $1/z$ to R respectively. □

Remark 5.9. The method we present for odd integers N is constructive, therefore leads to an algorithm that finds x_1, x_2 given x_3 such that $n(x) = n(y)$. We call this algorithm $\mathsf{EquivNormConjugationProduct}(x_3)$.

The condition $n(x) = n(y)$ is equivalent to $n(CDpx) = n(CDpy)$. And one could calculate explicitly that

$$n(CDpx) = p^2 n(A)n^2(D)n(x_3) + p^2 n(B)n(C)n(D)n(x_3) - 2p^2 n(D)\mathsf{Re}_R(A\bar{B}C\bar{D}x_3^2),$$
$$n(CDpy) = n(A)n^2(C)n(x_3) + p^2 n(B)n(C)n(D)n(x_3) + 2pn(C)\mathsf{Re}_R(A\bar{B}C\bar{D}x_3^2).$$

We now aim to find $x_3 = s + ti \in R$ with $(n(x_3), N) = 1$ such that $n(CDpx) - n(CDpy) = n(\gamma)\Big(n(A)(n(C) - pn(D)) + 2p\mathsf{Re}_R(A\bar{B}C\bar{D}x_3^2)\Big) = 0 \bmod N$. Plugging in $x_3 = s + ti$, finding x_3 is equivalent to finding $(s, t) \in \mathbb{Z}^2$ such that

$$f(s,t) := C_1 s^2 + C_2 st + C_3 t^2 = 0 \bmod N, \tag{8}$$

where $C_1 = \Big(n(A)\big(n(C) - pn(D)\big) + 2p\mathsf{Re}_R(A\bar{B}C\bar{D})\Big), C_2 = -4qp(\mathsf{Im}_R A\bar{B}C\bar{D})$ and $C_3 = \Big(qn(A)\big(n(C) - pn(D)\big) - 2qp\mathsf{Re}_R(A\bar{B}C\bar{D})\Big)$. Clearly, a solution (s, t) exists if and only if the discriminant

$$4q(4p^2 n(ABCD) - (n(AC) - pn(AD))^2) \tag{9}$$

of the quadratic equation is a square modulo N. By our assumption this is the case. Then (s, t) viewed in $(\mathbb{Z}/N\mathbb{Z})^2$ is defined up to a scalar, and we could simply choose $s = 1$ and let t_0 be one of the root of $f(1, t) \equiv 0 \bmod N$.

Finally, suppose $(n(x_3), N) = 1$ does not hold, this implies $(N, \mathsf{Im}_R(A\bar{B}C\bar{D}))$ which contradicts our assumption. Therefore, we have shown that we could find a solution $x_3 = s + ti$ where s, t satisfies Equation (8) and $(n(x_3), 1) = 1$.

We now summarize our algorithm for solving Problem 5.7 in Algorithm 4.

Algorithm 4: $\mathsf{QuaternionDecomposition}$(N,A,B,C,D)

Input: N, A, B, C, D as in Problem 5.7.
Output: $x_1, x_2, x_3 \in R$ such that
$A + Bj = \lambda x_1 j(C + Dj)x_2 j(C + Dj)x_3 j \bmod N\mathcal{O}_0$ where λ is some integer that is coprime to N.

$t_0 \leftarrow$ root of $\Big(n(A)\big(n(C) - pn(D)\big) + 2p\mathsf{Re}_R(A\bar{B}C\bar{D})\Big) - 4qp\mathsf{Im}_R(A\bar{B}C\bar{D})t + \Big(qn(A)\big(n(C) - pn(D)\big) - 2qp\mathsf{Re}_R(A\bar{B}C\bar{D})\Big)t^2 = 0 \bmod N.$

$x_3 \leftarrow 1 + t_0 i$

$x_1, x_2 \leftarrow \mathsf{EquivNormConjugationProduct}(x_3)$

Return x_1, x_2, x_3.

5.4 Quantum algorithm for the PQLP

As discussed earlier, Theorem 5.1 implies that we can solve the IsERP in quantum polynomial time. However, Theorem 5.1 only guarantees a randomized polynomial-time algorithm, and it might be advantageous to avoid that inside a quantum algorithm. So instead of lifting elements inside the quantum algorithm we lift $O(\log N)$ elements first and then utilize them to make the lifting procedure inside the quantum algorithm deterministic (and free of any heuristic after the precomputation has succeeded).

Theorem 5.10. *There is an algorithm that solves the PQLP in quantum polynomial time.*

Proof. We provide a proof for the case where N is a prime number. The proof for general N is in [CII+23, Appendix A]. For any matrix M in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$, M can be written as $PL \cdot D \cdot U$ where P is a permutation matrix, L is a lower unitriangular (it is lower triangular with 1-s in the diagonal), D is diagonal, and U is upper unitriangular (it is upper triangular with 1-s in the diagonal). This decomposition can be found in polynomial time using Gaussian elimination. Now one has to decompose L, D and U separately. Any lower unitriangular matrix can be written as a power of $A = \begin{pmatrix} 1 & 0 \\ g & 1 \end{pmatrix}$ where g is a generator of $(\mathbb{Z}/N\mathbb{Z})^*$. Similarly, every upper unitriangular matrix can be written as a power of $B = \begin{pmatrix} 1 & g \\ 0 & 1 \end{pmatrix}$. Any diagonal matrix can be written as $C^k D^l$ where

$$C = \begin{pmatrix} g & 0 \\ 0 & 1 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 \\ 0 & g \end{pmatrix}.$$

This shows that every element in $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ can be written as $PA^aB^bC^cD^d$. Thus instead of lifting elements inside the main quantum algorithm using Theorem 5.1 one can precompute lifts of P, A, B, C, D and then decompose a matrix $M \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ as $A^aB^bC^cD^d$ to obtain a lift of M.

This decomposition requires several instances of the discrete logarithm problem in $\mathbb{Z}/N\mathbb{Z}$ which can be computed in quantum polynomial time. The only issue is that if one computes a powersmooth lift of A (or B, C or D) then A^a is not going to be powersmooth if a is large. To circumvent this issue one also computes lifts of $A^{2^k}, B^{2^k}, C^{2^k}$ and D^{2^k} for every k between 1 and $\log_2(N)$. Furthermore, one computes lifts which are coprime (this can be ensured easily as StrongApproximation can lift an element in $\mathbb{Z}[i]j$ to any number that is bigger than $p^{O(1)}$).

This way $PA^aB^bC^cD^d$ will also be powersmooth as it is the product of $4\log N + 1$ powersmooth numbers. Lifting $4\log N + 1$ numbers can be done in classical polynomial time using Theorem 5.1.

□

6 Impact on isogeny-based cryptography

The most important application of Theorem 1.4 is that it breaks pSIDH in quantum polynomial time as N is a prime number in pSIDH. Another application is on SCALLOP [DFFK+23]. Even though Theorem 1.4 does not break SCALLOP, it shows that its security reduces to the problem of evaluating the secret prime degree isogeny (up to a scalar). In [DFFK+23] it is already discussed that one can deduce some information on the secret isogeny ϕ by utilizing the fact that one can evaluate $\phi \circ \theta \circ \hat{\phi}$ efficiently on any point where θ is some fixed endomorphism on a curve which has an endomorphism of low degree (typically that curve is j-1728 and θ is the non-trivial automorphism).

Our results mildly generalize to the following setting. Let E be a supersingular elliptic curve that does not possess a non-scalar endomorphism of degree N^2. In other words there is a one-to-one correspondence between cyclic subgroups of order N and N-isogenous curves. Our results imply that if given some curve E/A and an endomorphism σ one can compute $E/\sigma(A)$, then one can also compute the endomorphism ring of E/A in quantum polynomial time (and actually the corresponding isogeny itself its degree is smooth). The difference between our previous results here is that we do not need an isogeny representation for the isogeny corresponding to the subgroup generated by A as long as we can evaluate the above group action. At the moment we do not see any particular application for this observation but it might prove to be a useful cryptanalysis tool in the future.

Acknowledgements Gábor Ivanyos is supported in part by the Hungarian Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program. Péter Kutas is supported by the Hungarian Ministry of Innovation and Technology NRDI Office within the framework of the Quantum Information National Laboratory Program, the J ´anos Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the UNKP-22-5 New National Excellence Program. Mingjie Chen, Péter Kutas and Christophe Petit are partly supported by EPSRC through grant number EP/V011324/1.

Bibliography

[BL95] Dan Boneh and Richard J Lipton. Quantum cryptanalysis of hidden linear functions. In *Annual International Cryptology Conference*, pages 424–437. Springer, 1995.

[CD22] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh (preliminary version). *Cryptology ePrint Archive*, 2022.

[CII+23] Mingjie Chen, Muhammad Imran, Gábor Ivanyos, Péter Kutas, Antonin Leroux, and Christophe Petit. Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of psidh. Cryptology ePrint Archive, Paper 2023/779, 2023. `https://eprint.iacr.org/2023/779`.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology - ASIACRYPT 2018*, pages 395–427, 2018.

[Coh95] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[Cou99] Jean-Marc Couveignes. Hard homogeneous spaces. *Preprint at https://eprint. iacr. org/2006/291*, 1999.

[CVD05] Andrew M Childs and Wim Van Dam. Quantum algorithm for a generalized hidden shift problem. *arXiv preprint quant-ph/0507190*, 2005.

[DFFK+23] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. Scallop: scaling the csi-fish. *PKC*, 2023.

[DFKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: Compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology – ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I*, page 64–93, Berlin, Heidelberg, 2020. Springer-Verlag.

[DK12] Yurj A Drozd and Vladimir V Kirichenko. *Finite dimensional algebras*. Springer Science & Business Media, 2012.

[DMR10] Aaron Denney, Cristopher Moore, and Alexander Russell. Finding conjugate stabilizer subgroups in PSL and related groups. *Quantum Inf. Comput.*, 10:282–291, 2010.

[EH00] Mark Ettinger and Peter Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25(3):239–251, 2000.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, pages 329–368, 2018.

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

[git23] Pqlp-prime github repository, 2023. `https://github.com/pqcisogeny/PQLP_prime.git`.

[IPS18] Gábor Ivanyos, Anupam Prakash, and Miklos Santha. On Learning Linear Functions from Subset and Its Applications in Quantum Computing. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 66:1–66:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Iva12] Gábor Ivanyos. Finding hidden borel subgroups of the general linear group. *Quantum Inf. Comput.*, 12(7-8):661–669, 2012.

[JDF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

[Kit95] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.

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

[Koh96] David Russell Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[Ler22a] Antonin Leroux. A new isogeny representation and applications to cryptography. *ASIACRYPT*, 2022.

[Ler22b] Antonin Leroux. *Quaternion Algebra and isogeny-based cryptography.* PhD thesis, Ecole doctorale de l'Institut Polytechnique de Paris, 2022.

[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 *Advances in Cryptology - ASIACRYPT 2017*, pages 330–353, 2017.

[PL17] Christophe Petit and Kristin Lauter. Hard and easy problems for supersingular isogeny graphs. Cryptology ePrint Archive, Report 2017/962, 2017. `https://eprint.iacr.org/2017/962`.

[PLQ08] Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater. Full cryptanalysis of lps and morgenstern hash functions. In *Security and Cryptography for Networks: 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings 6*, pages 263–277. Springer, 2008.

[QKL+21] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E Stange. Improved torsion-point attacks on sidh variants. In *Annual International Cryptology Conference*, pages 432–470. Springer, 2021.

[Rob22] Damien Robert. Evaluating isogenies in polylogarithmic time. *Cryptology ePrint Archive*, 2022.

[Rón90] Lajos Rónyai. Computing the structure of finite algebras. *Journal of Symbolic Computation*, 9(3):355–373, 1990.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006:145, 2006.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[The22] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.4)*, 2022. `https://www.sagemath.org`.

[Voi18] John Voight. Quaternion algebras. *preprint*, 13:23–24, 2018.

[Wat69] William C Waterhouse. Abelian varieties over finite fields. In *Annales scientifiques de l'École Normale Supérieure*, volume 2, pages 521–560, 1969.

[Wes21] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. Cryptology ePrint Archive, Report 2021/919, 2021. `https://ia.cr/2021/919`.