Computing the Endomorphism Ring of a Supersingular Elliptic Curve from a Full Rank Suborder

Mingjie Chen[1](✉) and Christophe Petit[2,3]

[1] KU Leuven, Leuven, Belgium
mjchennn555@gmail.com
[2] Université libre de Bruxelles, Brussels, Belgium
[3] University of Birmingham, Birmingham, UK

Abstract. In this paper, we study the problem of computing the endomorphism ring of a supersingular elliptic curve given the knowledge of a full rank suborder. We provide a polynomial time quantum algorithm to solve this problem in full generality. This result enhances our understanding of the endomorphism ring problem, which is at the core of isogeny-based cryptography.

As part of our approach, we also present a polynomial time quantum algorithm to solve the problem of computing the endomorphism ring of the codomain curve of an isogeny from a curve with known endomorphism ring. This extends the work of [CII+23a] by lifting their restrictions on the number of factors of the isogeny degree.

As an application, we present quantum reductions between key hard problems in isogeny-based cryptography. We show that some of our quantum reductions are tighter than the classical ones, while all reductions are of polynomial time complexity. In particular, we improve the query complexity of the reduction of the EndRing problem to the OneEnd problem from $\mathrm{poly}(\log p)$ (classically) to $O(1)$ (quantumly), strengthening the hardness assumption of the OneEnd problem in the post-quantum setting. This reduction underlies the 2-special soundness proof of SQIsign identification protocols.

1 Introduction

The problem of computing the endomorphism ring $\mathrm{End}(E)$ of a supersingular elliptic curve E defined over a finite field of characteristic p, known as the EndRing problem, is a central problem in isogeny-based cryptography. Its hardness was shown to underpin the security of nearly all isogeny-based constructions, including the CGL hash function [CGL09], the SQIsign digital signature [DFKL+20], the CSIDH [CLM+18] and SCALLOP [FFK+23] key exchange protocols, among others.

The earliest work on computing $\mathrm{End}(E)$ dates back to Kohel in 1996, who gave an approach for generating a suborder of finite index of the endomorphism ring $\mathrm{End}(E)$ [Koh96, Theorem 75]. The algorithm is based on finding cycles

S. Fehr and P.-A. Fouque (Eds.): EUROCRYPT 2025, LNCS 15606, pp. 446–474, 2025.
https://doi.org/10.1007/978-3-031-91095-1_16

in the ℓ-isogeny graph of supersingular elliptic curves for a small prime ℓ and runs in time $O(p^{1+\epsilon})$. Specifically, since $\mathrm{End}(E)$ is a $\mathbb{Z}$-lattice of rank 4, any suborder of finite index (or equivalently, full rank) are generated by two non-commuting endomorphisms on E, which are cycles on the isogeny graph in this case. However, Kohel did not discuss how to compute $\mathrm{End}(E)$ from this suborder and left it as future work.

In 2020, [EHL+20] completed Kohel's algorithm by presenting a subexponential time algorithm to recover the full endomorphism ring $\mathrm{End}(E)$ when the given suborder is a Bass order. Under their heuristic assumption, their input suborder, which is obtained from cycle-finding in the supersingular ℓ-isogeny graph, is a Bass order with constant probability. In 2024, the more general problem of computing $\mathrm{End}(E)$ from a generic suborder of finite index d is considered in [ES24], and they present an algorithm for this problem that runs in polynomial time in $\log p$, $\log d$ and linear time in the largest prime dividing d. This algorithm is only polynomial when d is smooth w.r.t. a smoothness bound of size $\mathrm{poly}(\log p)$. This is a rather strong assumption as suborders of index d exist for any given integer d, in particular, d can be a prime of size linear in p, in which case the runtime becomes exponential in $\log p$.

In fact, the presumed hardness of computing $\mathrm{End}(E)$ from a suborder of an index containing a large prime factor was used by Leroux in 2022 [Ler22a] to build a key exchange protocol called pSIDH, where such a suborder is provided to enable the computation of the shared key. In 2023, this protocol was broken by a polynomial time quantum algorithm presented in [CII+23a]. Naturally, this result weakens the hardness of the problem of computing $\mathrm{End}(E)$ from a suborder. However, this quantum algorithm only targets the special class of suborders used in pSIDH, namely the suborders in $\mathrm{End}(E)$ that are embedding of order of the form $\mathbb{Z} + N\mathcal{O}_0$ where $\mathcal{O}_0$ is a special maximal order defined in Sect. 2.2, and $N \neq p$ is a prime. Clearly, this special class only covers a small portion of suborders in $\mathrm{End}(E)$.

Despite the limited use cases, the isogeny community has increasingly come to believe that computing $\mathrm{End}(E)$ from a suborder can generally be achieved in quantum polynomial time after the attack in [CII+23a]. This impacts both cryptographic constructions and cryptanalysis in the field. For instance, it suggests that finding one more endomorphism that does not commute with the existing orientation in orientation-based construction such as CSIDH [CLM+18] and SCALLOP [FFK+23] suffices to break them. In this paper, we substantiate this belief by presenting a polynomial time quantum algorithm that solves the problem of computing $\mathrm{End}(E)$ from a suborder in full generality.

1.1 Contributions

In this paper, we prove the following theorems. Note that both of the theorems will be labeled with "heuristic", which suggests that they rely on plausible heuristics. We do not record the heuristics here but they can be found in Remark 27, Remark 31 and [CP25, Appendix A]. In Sect. 7, we briefly discuss how we expect to remove some of them.

Theorem 1 (heuristic). *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Given a full rank suborder $\mathcal{R}_E \subseteq \mathrm{End}(E)$ of discriminant $\Delta_{\mathcal{R}}$, there exists a quantum algorithm that computes $\mathrm{End}(E)$ in expected polynomial time in $\log p$ and $\log(\sqrt{\Delta_{\mathcal{R}}}/p)$.*

Remark 2. The suborder $\mathcal{R}_E \subseteq \mathrm{End}(E)$ will be given by an efficient representation as defined in Sect. 2.3. In this paper, we work under the assumption that the degrees of the basis elements of $\mathcal{R}_E$ are bounded above by $\mathrm{poly}(\Delta_{\mathcal{R}})$, this can be realized by applying lattice reduction algorithms to the given basis to obtain a Minkowski reduced basis of $\mathcal{R}_E$, which takes time polynomial in $\log p$ and log of the maximum degree of the given basis elements to compute.

In order to prove Theorem 1, we reduce the problem of computing $\mathrm{End}(E)$ given $\mathcal{R}_E$ to a related problem. Specifically, to the problem of computing $\mathrm{End}(E)$ given a weak isogeny representation (see Sect. 2.3) of $\varphi : \widetilde{E} \to E$ of degree N and $\mathrm{End}(\widetilde{E})$. This is the Isogeny to Endomorphism Ring Problem (IsERP) introduced in [CII+23a], which is partially solved using quantum algorithms in the same work when (N, p) is an odd integer of $O(\log \log p)$ many factors under certain heuristics assumptions. As a crucial component of the proof of Theorem 1, we solve the IsERP in its full generality, as stated in Theorem 3.

Theorem 3 (heuristic). *Let $\widetilde{E}, E$ be two supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\varphi : \widetilde{E} \to E$ be an isogeny of degree N for some integer N. Given $\mathrm{End}(\widetilde{E})$ and a weak isogeny representation for φ, there exists a quantum algorithm that computes $\mathrm{End}(E)$ in expected polynomial time in $\log p$ and $\log N$.*

In the case when the integer N is an ℓ-power for a small prime ℓ, the IsERP is then the well-known problem of reducing EndRing to the path-finding problem on the ℓ-isogeny graph, and a polynomial time classical reduction exists [EHL+18, Wes21], which can be easily generalized to a polynomial time classical algorithm that solves the IsERP for a smooth integer N w.r.t. to a smoothness bound of size $\mathrm{poly}(\log p)$. We stress here that the IsERP remains hard classically for a generic integer N. On the other hand, Theorem 3 shows that it is easy for quantum computers.

Remark 4. There can be a natural hybrid approach for the IsERP that combines the classical algorithm (to remove small factors) and the quantum algorithms (to remove large factors, but the number of factors cannot be so many). However, there are integers N that contain, loosely speaking, *many sufficiently large factors* such that the hybrid approach does not work for such integers N. For instance, let k be some positive integer and $B = (\log p)^k$ be the smoothness bound, let $N = \prod_{i=1}^{r} p_i$ for primes p_i, then as long as $r < \frac{\lfloor \log N \rfloor}{k \log \log p}$, we can have each factor p_i exceeding B since in this case

$$N \approx 2^{\lfloor \log N \rfloor} > 2^{rk(\log \log p)} = (2^{\log \log p})^{kr} = B^r.$$

As applications of our results to isogeny-based cryptography, we use these quantum algorithms to build quantum reductions between important hard problems in isogeny-based cryptography including the EndRing, OneEnd, ℓ-Isogeny and Isogeny problems (see Sect. 6 for their definitions). Even though polynomial time classical reductions have been established between these problems, we show that our quantum reductions are often *tighter*, meaning that they have *lower query complexity*, which makes such reductions more practical in the post-quantum setting. Specifically, we show that the query complexity of the reduction from EndRing to OneEnd can be improved from $\mathrm{poly}(\log p)$ (classically [PW24]) to $O(1)$ (quantumly), and the query complexity of the reduction from EndRing to Isogeny can be improved from $\mathrm{poly}(\log p)$ (classically [PW24]) to only 1 (quantumly). The hardness of the OneEnd problem lies under the 2-special soundness property of SQIsign digital signatures [DFKL+20, DLRW24, BFD+24, NO24, DF24]; our tighter reduction from EndRing to OneEnd greatly strengthen the security reduction of SQIsign. For a detailed account of the applications, see Sect. 6.

1.2 Technical Overview

Recall that the quantum algorithm from [CII+23a] works for suborders $\mathcal{R}_E \subseteq \mathrm{End}(E)$ when $\mathcal{R}_E$ is the image of an embedding of $\mathbb{Z} + N\mathcal{O}_0$ into $\mathrm{End}(E)$ where $N \neq p$ is a prime. In this section, for the convenience of discussion, we refer to the problem of computing $\mathrm{End}(E)$ given such a suborder as the SpecialSubOrderEndoRing problem, and we refer to the problem of computing $\mathrm{End}(E)$ given a full rank suborder as the SuborderEndoRing problem. Leroux proved in [Ler22a] that this special suborder (as in the SpecialSubOrderEndoRing problem) gives rise to an isogeny from a special curve E_0 w.r.t. $\mathcal{O}_0$ (see Remark 15) to E together with a weak isogeny representation for this isogeny. Now recovering $\mathrm{End}(E)$ becomes a special instance of the Isogeny to Endomorphism Ring Problem (IsERP) that is the main object of [CII+23a], whose definition we recall here.

Problem 5 (Isogeny to Endomorphism Ring Problem (IsERP)). Let $\widetilde{E}, E$ be two supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\varphi : \widetilde{E} \to E$ be an isogeny of degree N for some integer N. Given $\mathrm{End}(\widetilde{E})$ and a weak isogeny representation for ϕ, compute $\mathrm{End}(E)$.

Note that given an isogeny $\varphi : \widetilde{E} \to E$ of degree N coprime to p and a matrix $g \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$, one can define an action $g \star \varphi$ to be the isogeny with kernel $\theta_g(\ker \varphi)$, where $\theta_g \in (\mathrm{End}(\widetilde{E})/N\,\mathrm{End}(\widetilde{E}))^\times$ is obtained through an isomorphism between $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ and $(\mathrm{End}(\widetilde{E})/N\,\mathrm{End}(\widetilde{E}))^\times$, which can be computed in polynomial time as explained in [CII+23a]. We do not give a notation for this isomorphism but use θ_g to indicate the relation. Note that $\theta_g(\ker \varphi)$ is still a point of order N. In [CII+23a], the IsERP is then reduced to the following Group Action Evaluation Problem (GAEP) in quantum polynomial time.

Problem 6 (Group Action Evaluation Problem (GAEP)). Let $\widetilde{E}, E$ be two supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\varphi : \widetilde{E} \to E$ be an isogeny of degree N for some integer N coprime to p. Given $\mathrm{End}(\widetilde{E})$, a weak isogeny representation for φ, $g \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$, find a weak isogeny representation of $g \star \varphi$.

The GAEP is further reduced to a pure quaternion algebra problem called the Powersmooth Quaternion Lift Problem (PQLP). This reduction runs in classical polynomial time.

Problem 7 (Powersmooth Quaternion Lift Problem (PQLP)). Let $\widetilde{\mathcal{O}}$ be a maximal order in $\mathcal{B}_{p,\infty}$. Given an integer $(N, p) = 1$ and an element $\sigma_0 \in \widetilde{\mathcal{O}}$ such that $(n(\sigma_0), N) = 1$, find $\sigma \in \widetilde{\mathcal{O}}$ and $\lambda \in (\mathbb{Z}/N\mathbb{Z})^\times$ such that $\sigma = \lambda\sigma_0 \bmod N\widetilde{\mathcal{O}}$ and that $n(\sigma)$ is powersmooth.

Finally, in [CII+23a], they develop a polynomial time classical algorithm that resolves the PQLP when N is an odd integer with at most $O(\log\log p)$ many factors. These conditions are certainly satisfied by the isogeny defined by $\mathbb{Z} + N\mathcal{O}_0$ where the degree N is a prime different from p.

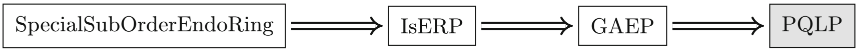


Fig. 1. Reductions established in [CII+23a] and [Ler22a] in order to solve the SpecialSubOrderEndoRing problem. Here " $\Longrightarrow$ " means that the problem on the left reduces to the problem on the right. The last box is shaded to indicate that the PQLP is only resolved when N is odd and has at most $O(\log\log p)$ many factors. Among the reductions, the one from IsERP to GAEP is a quantum reduction and the rest are classical reductions.

We summarize in Fig. 1 the approach for computing $\mathrm{End}(E)$ with ingredients from [Ler22a] and [CII+23a]. The reductions shown in the figure inspired the following strategy for solving the general problem of computing $\mathrm{End}(E)$ from a full rank suborder $\mathcal{R}_E$:

A. Reduce the SubOrderEndoRing problem to an instance of the IsERP (note that the literature so far only provided such a reduction when the suborder is of a special shape). This step can be also divided into three substeps as follows:
   - A1. Derive an embedding of $\mathbb{Z} + N\mathcal{O}_0$ into $\mathrm{End}(E)$ for some N, based on the knowledge of $\mathcal{R}_E$.
   - A2. Generalize the theory for the existence of an isogeny $\varphi : E_0 \to E$ from the embedding $\mathbb{Z}+N\mathcal{O}_0$ to the broader case, where N is no longer necessarily a prime coprime to p.
   - A3. Extend the method for obtaining a weak isogeny representation for φ to arbitrary integers N.

B. Solve the PQLP in full generality.

We solve task A as mentioned above in Sect. 3. Note that task A2 was briefly discussed in [Ler22a, Appendix A, Proposition 24], but we add more details to the proof. Then we present a new algorithm to solve the general PQLP in Sect. 4, only requiring N being odd. We conclude the proof of our main theorems by reducing the general IsERP to the one where we restrict N to be an odd integer coprime to p in Sect. 5. The roadmap is summarized in Fig. 2.

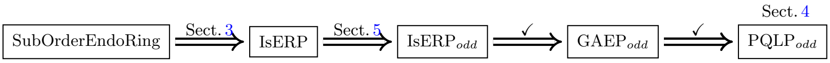


Fig. 2. Outline of our strategy for solving the SubOrderEndoRing problem.Here " $\Longrightarrow$ " means that the problem on the left reduces to the problem on the right. By the subscript $_{odd}$, we refer to the case when N is odd where N is either the degree of an isogeny or the modulus depending on the concrete problem. Among the reductions, the one from IsERP to GAEP is a quantum reduction and the rest are classical reductions.

1.3 Outline

This paper is organized as follows: in Sect. 2, we introduce necessary background. In Sect. 3, we discuss how to obtain an isogeny with a weak isogeny representation for it from the knowledge of a suborder $\mathcal{R}_E \subseteq \mathrm{End}(E)$. In Sect. 4, we present a new algorithm that resolves the PQLP when N is odd. In Sect. 5, we present the proofs for Theorem 1 and Theorem 3. Finally, in Sect. 6, we discuss the applications of our algorithm to isogeny-based cryptography.

2 Preliminaries

In this section we recall some basic notions on supersingular elliptic curves, isogenies and quaternion algebras, concluding with some commonly used algorithms in isogeny-based cryptography.

2.1 Notations

Throughout, the letter p will denote a prime integer greater than 3. A function $g(n)$ is $O(f(n))$ if there exist constants $c > 0$ and n_0 such that for all $n \geq n_0$, $g(n) \leq c \cdot f(n)$; A function $g(n)$ is $\Omega(f(n))$ if there exist constants $c > 0$ and n_0 such that for all $n \geq n_0$, $g(n) \geq c \cdot f(n)$. Denote by $\mathrm{poly}(f(n))$ the set of functions that are polynomial in $f(n)$. The "soft-O" notation $\widetilde{O}$ is shorthand for $O(\mathrm{poly}(\log f(n)) f(n))$, which hides polylogarithmic factors in the asymptotic complexity. An integer n is said to be B-*powersmooth* if none of its prime power factors exceeds B, and is said to be *powersmooth* if $B = O(\mathrm{poly}(\log p))$. For $n \in \mathbb{Z}$, the notation $\omega(n)$ counts the number of distinct prime factors of n.

2.2 Quaternion Algebras

An algebra $\mathcal{B}$ is a *quaternion algebra over* $\mathbb{Q}$ if there exist $a, b \in \mathbb{Q}^\times$ and $\mathbf{i}, \mathbf{j} \in B$ such that $(1, \mathbf{i}, \mathbf{j}, \mathbf{ij})$ is a $\mathbb{Q}$-basis for $\mathcal{B}$ and

$$\mathbf{i}^2 = a, \quad \mathbf{j}^2 = b, \quad \text{and} \quad \mathbf{ji} = -\mathbf{ij}.$$

Given a and b, the corresponding algebra is denoted by $\left(\frac{a,b}{\mathbb{Q}}\right)$. Write an arbitrary element of B as $\alpha = x_1 + x_2\mathbf{i} + x_3\mathbf{j} + x_4\mathbf{ij}$ with $x_i \in \mathbb{Q}$. The quaternion algebra B has a canonical involution $\alpha \mapsto \overline{\alpha} = x_1 - x_2 i - x_3 j - x_4 ij$. It induces the *reduced trace* and the *reduced norm*

$$\operatorname{tr}(\alpha) = \alpha + \overline{\alpha} = 2x_1, \quad \operatorname{n}(\alpha) = \alpha\overline{\alpha} = x_1^2 - ax_2^2 - bx_3^2 + abx_4^2.$$

A *fractional ideal* I in $\mathcal{B}$ is a $\mathbb{Z}$-lattice of rank 4. We denote by $n(I)$ the *reduced 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 $\mathcal{B}$ that is also a fractional ideal. An order is called *maximal* when it is not contained in any other larger order. A fractional ideal is *integral* if it is contained in its *left order* $\mathcal{O}_L(I) = \{\alpha \in \mathcal{B}_{p,\infty} \mid \alpha I \subset I\}$, or equivalently in its *right order* $\mathcal{O}_R(I) = \{\alpha \in \mathcal{B}_{p,\infty} \mid I\alpha \subset I\}$.

To any prime number p, one associates a quaternion algebra $\mathcal{B}_{p,\infty}$, defined as the unique quaternion algebra over $\mathbb{Q}$ ramified exactly at p and ∞. Explicitly, it is given by the following lemma.

Lemma 8. *Let $p > 2$ be a prime. Then, $\mathcal{B}_{p,\infty} = \left(\frac{-q,-p}{\mathbb{Q}}\right)$, where*

$$q = \begin{cases} 1 & \textit{if } p \equiv 3 \pmod 4, \\ 2 & \textit{if } p \equiv 5 \pmod 8, \\ q_p & \textit{if } p \equiv 1 \pmod 8, \end{cases}$$

where q_p is the smallest prime such that $q_p \equiv 3 \pmod 4$ and $\left(\frac{p}{q_p}\right) = -1$. Assuming GRH, we have $q_p = O((\log p)^2)$, which can thus be computed in polynomial time in $\log p$.

Remark 9. For any given integer N, in the case when $p \equiv 1 \bmod 8$, we can choose q such that $(N, q) = 1$ at the cost of increasing the size of q, while keeping it in $O(\text{poly}(\log p))$ under GRH.

To each $\mathcal{B}_{p,\infty}$ defined as above, we can identify a maximal order which is called a *special extremal order* $\mathcal{O}_0$ (first introduced in [KLPT14]) containing a suborder admitting an orthogonal decomposition $R + \mathbf{j}R$, where $R = \mathbb{Z}[\omega] \subset \mathbb{Q}[\mathbf{i}]$ is a quadratic order of minimal discriminant (or equivalently such that ω has smallest norm in $\mathcal{O}_0$). By orthogonal decomposition, we mean that $R \subset (\mathbf{j}R)^\perp$.

Lemma 10. *For any prime $p > 2$, the quaternion algebra $\mathcal{B}_{p,\infty}$ contains the maximal order*

$$\mathcal{O}_0 = \begin{cases} \left\langle 1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{ij}}{2}, \frac{1+\mathbf{j}}{2} \right\rangle & \textit{if } p \equiv 3 \pmod 4, \\ \left\langle 1, \mathbf{i}, \frac{2-\mathbf{i}+\mathbf{ij}}{4}, \frac{-1+\mathbf{i}+\mathbf{j}}{2} \right\rangle & \textit{if } p \equiv 5 \pmod 8, \\ \left\langle \frac{1+\mathbf{i}}{2}, \frac{\mathbf{j}+\mathbf{ij}}{2} \frac{1+c\mathbf{ij}}{q}, \mathbf{ij} \right\rangle & \textit{if } p \equiv 1 \pmod 8, \end{cases}$$

where in the last case c is an integer such that $q \mid c^2p+1$. Assuming GRH, the maximal order $\mathcal{O}_0$ contains the suborder $R + R\mathbf{j}$ with index $O((\log p)^2)$, where R is the ring of integers of $\mathbb{Q}(\mathbf{i})$. If ω is a reduced generator of R, then

$$n(x + y\mathbf{i} + \mathbf{j}(z + \mathbf{i}w)) = f(x, y) + pf(z, w),$$

where $f(x, y)$ is a principal, primitive, positive definite, integral binary quadratic form ([Cox89, Section 2]) of discriminant $\operatorname{Disc}\mathbb{Q}(\mathbf{i}) = O((\log p)^2)$.

Remark 11. The integer c can be chosen such that its absolute value is smaller than q. In particular, in the case when $p \equiv 1 \bmod 8$, we can choose c such that $c \bmod 8$ is a square root of $-p^{-1} \mod q$, and $-p^{-1}$ is always a square modulo q by the restrictions on q as in Lemma 8.

Remark 12. Let Δ_R denote the discriminant of R, which is in fact equal to $\operatorname{Disc}\mathbb{Q}(\mathbf{i})$ as R is the maximal order of $\mathbb{Q}(\mathbf{i})$. We can easily deduce that

$$\Delta_R = \begin{cases} -4, & \text{for } p \equiv 3 \bmod 4, \\ -8, & \text{for } p \equiv 5 \bmod 8, \\ -q, & \text{for } p \equiv 1 \bmod 8. \end{cases}$$

Computing the discriminant of the suborder $R + R\mathbf{j}$ implies that the index $[\mathcal{O}_0 : R + R\mathbf{j}] = \Delta_R$.

2.3 Elliptic Curves and Isogenies

Let E, E_1, E_2 be elliptic curves defined over a finite field of characteristic p. An isogeny from E_1 to E_2 is a non-constant rational map that is simultaneously a group homomorphism. An isogeny from a curve E to itself is an *endomorphism.* The set $\operatorname{End}(E)$ of all endomorphisms of E forms a ring under addition and composition. $\operatorname{End}(E)$ is either an order in an imaginary quadratic field and E is called *ordinary*, or a maximal order in $\mathcal{B}_{p,\infty}$, in which case E is called *supersingular.*

Let $\varphi : E \to E_1$, $\psi : E \to E_2$, and $\psi' : E_1 \to E'$ be isogenies of coprime degrees. If $\ker\psi' = \varphi(\ker\psi)$ holds, we say that ψ' is the push-forward of ψ by φ and denote it by $\psi' = [\varphi]_*\psi$. Under the same situation, we say that ψ is the pull-back of ψ' by φ and denote it by $\psi = [\varphi]^*\psi$.

By an efficient representation for an endomorphism α, we mean that there is an algorithm to evaluate $\alpha(P)$ for any $P \in E(\mathbb{F}_{p^k})$ in time polynomial in the length of the representation of α and in $k\log(p)$. We also assume that an efficient representation of α has length $O(\log(\deg(\alpha)))$. By an efficient representation of a suborder $\mathcal{R}_E \subseteq \mathrm{End}(E)$, we mean efficient representations of basis elements $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of $\mathcal{R}_E$.

Definition 13. *A weak isogeny representation for the isogeny $\varphi : E \to E'$, is a data s_φ of size $O(\mathrm{poly}\log(p + \deg\varphi))$ 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 a generator of $\langle\varphi(P)\rangle$ in time polynomial in $\log d, \log\deg\varphi$ and $|P|$ where $|P|$ is the bitsize of the representation of P.*

2.4 The Deuring Correspondence

Given a supersingular elliptic curve E over $\mathbb{F}_{p^2}$, its endomorphism ring $\mathrm{End}(E)$ is isomorphic to a maximal order in $B_{p,\infty}$. This in fact leads to a bijection called *the Deuring correspondence.*

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

Definition 14. *Given I an integral left $\mathcal{O}$-ideal coprime to p, we define the I-torsion $E[I] = \{P \in E(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0$ for all $\alpha \in I\}$. To I, we associate the separable isogeny φ_I of kernel $E[I]$. Conversely given a separable isogeny φ, the corresponding ideal is defined as $I_\varphi = \{\alpha \in \mathcal{O} : \alpha(P) = 0$ for all $P \in \ker(\varphi)\}$.*

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

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

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

Remark 15. Via the Deuring correspondence, we can associate each special maximal order $\mathcal{O}_0$ defined in Sect. 2.2 to a supersingular elliptic curve E_0. These curves E_0 are defined over $\mathbb{F}_p$ since $\mathcal{O}_0$ contains an element whose square is $-p$, and for a fixed $\mathcal{O}_0$, the curve E_0 is unique up to isomorphism over $\overline{\mathbb{F}}_p$. We call these elliptic curves *special curves w.r.t. $\mathcal{O}_0$.*

2.5 Algorithmic Building Blocks

We recall algorithmic building blocks that are commonly used in isogeny-based cryptography. These algorithms are taken from [Cor08,KLPT14,DFKL+20] unless otherwise specified.

- $\mathsf{Cornacchia}(M)$, given 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 $M = x^2 + qy^2$.
- $\mathsf{RepresentInteger}_{\mathcal{O}_0}(M)$, given an $M > p$, outputs $\gamma \in \mathcal{O}_0$ such that $n(\gamma) = M$.
- $\mathsf{IdealModConstraint}(I, \gamma)$, given an ideal I of norm N, and $\gamma \in \mathcal{O}_0$ such that $\gcd(n(\gamma), N^2) = N$, finds $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $\mu_0 = \gamma\mathbf{j}(C + \omega D)$ satisfies $\mu_0\gamma \in I$.
- $\mathsf{EichlerModConstraint}(I, \gamma)$, given an ideal I of norm N, and $\gamma \in \mathcal{O}_0$ such that $(n(\gamma), N) = 1$, finds $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $\mu_0 = \gamma\mathbf{j}(C + \omega D)$ satisfies $\mu_0\gamma \in \mathbb{Z} + I$.
- $\mathsf{GenStrongApproximation}(N, C, D, \mathsf{ps})$ ([CP25, Algorithm 7]), given an integer N, and $(C : D) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $(pf(C, D), N) = 1$, and a boolean value ps, finds $\lambda \in (\mathbb{Z}/N\mathbb{Z})^\times$ and $\mu \in \mathcal{O}_0$ such that $(n(\mu), N) = 1$ and $\mu = \lambda(C + \omega D)\mathbf{j} \bmod N\mathcal{O}_0$. In addition, μ is powersmooth when $\mathsf{ps} = \text{True}$.

Remark 16. Even though not directly documented in the literature, this algorithm $\mathsf{GenStrongApproximation}(N, C, D, \mathsf{ps})$ can be adapted easily from [CII+23a, Algorithm 2]. For completeness, we describe the algorithm in [CP25, Appendix A]. We use GenStrongApproximation instead of StrongApproximation to name this algorithm to stress the difference from such algorithms originated from [KLPT14] where N is required to be a prime.

3 A Full Rank Suborder Defines an Isogeny with a Weak Isogeny Representation

In this section, we explain how to derive an isogeny and its weak isogeny representation from a full rank suborder.

3.1 Recovering an Isogeny

Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ and $\mathcal{R}_E$ be a quaternion suborder that is contained in $\mathrm{End}(E)$. Let $\Delta_{\mathcal{R}}$ denote the discriminant of $\mathcal{R}_E$, which can be efficiently computed from the Gram matrix of a basis of $\mathcal{R}_E$. Let $D_{\mathcal{R}}$ denote the index $[\mathrm{End}(E) : \mathcal{R}_E]$, then $\Delta_{\mathcal{R}} = D_{\mathcal{R}}^2 p^2$. Since our main algorithm is a quantum algorithm, we may as well assume that the factorization of $D_{\mathcal{R}}$ is given by Shor's algorithm [Sho97]. Despite the lack of information on the full endomorphism ring $\mathrm{End}(E)$, the knowledge of a full rank suborder $\mathcal{R}_E$ suffices to define an isomorphism:

$$\iota : \mathcal{B}_{p,\infty} \to \mathrm{End}^0(E) := \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}.$$

Remark 17. Explicitly, this isomorphism can be computed following [EHL+18, Algorithm 6]. We start from step 2 of Algorithm 6 and replace the basis $\{1, \alpha, \beta, \gamma\}$ there with a basis of $\mathcal{R}_E$. [EHL+18, Proposition 6] states that under plausible heuristics, this algorithm runs in polynomial time in the input size. Specifically to our case, it runs in polynomial time in $\log \Delta_{\mathcal{R}}$. Note that the heuristics come from not knowing the factorization of several integers, which can be removed given our quantum setting.

Let $\mathcal{R}$ denote $\iota^{-1}(\mathcal{R}_E)$, and $\mathcal{O}$ denote the image $\iota^{-1}(\mathrm{End}(E))$. It is known that $\mathcal{O}$ is a maximal quaternion order in $\mathcal{B}_{p,\infty}$ from the Deuring correspondence.

Let $\mathcal{O}_0 \subseteq \mathcal{B}_{p,\infty}$ be a maximal order. From [Ler22a, Lemma 3] one learns that when N is prime, an inclusion $\mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{O}$ implies that there is an ideal I of norm N such that $\mathcal{O}_L(I) = \mathcal{O}$ and $\mathcal{O}_R(I) = \mathcal{O}_0$ when $\mathcal{O} \neq \mathcal{O}_0$. Furthermore, when N is composite, if this embedding is *primitive* in the sense of [Ler22a, Appendix A], then this embedding also defines such an ideal I (this is sketched as the proof of one direction of [Ler22a, Appendix A, Proposition 24]).

Motivated by these results, we first present Algorithm 1 and Algorithm 2 to obtain a primitive embedding of a special maximal order $\mathcal{O}_0$ into $\mathrm{End}(E)$. After having done so, we provide a more detailed proof of [Ler22a, Appendix A, Proposition 24] with Proposition 21, Lemma 22 and Lemma 23. Finally, we derive the isogeny in Proposition 24. We highlight our unique contributions in comparison to [Ler22a] at relevant points throughout the discussion.

Algorithm 1. SpecialSuborder($\mathcal{R}$)

```
Input: A quaternion order $\mathcal{R}$ and a maximal order $\mathcal{O}_0$ inside $\mathcal{B}_{p,\infty}$
Output: The smallest integer $N$ such that $\mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{R}$
$\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} \leftarrow$ a $\mathbb{Z}$-basis of $\mathcal{O}_0$
$\boldsymbol{\beta} = \{\beta_1, \beta_2, \beta_3, \beta_4\} \leftarrow$ a $\mathbb{Z}$-basis of $\mathcal{R}$
Compute the unique matrix $\mathrm{M} \in \mathrm{M}_4(\mathbb{Q})$ such that $\boldsymbol{\beta} M = \boldsymbol{\alpha}$
$m_{ij} \leftarrow (i, j)$-th entry of $M$ for $i, j = 1, \ldots, 4$
Write $m_{ij} \in \mathbb{Q}$ in reduced from
$N \leftarrow$ the least common multiple of the denominators of nonzero $m_{ij}$ for $i, j = 1, \ldots, 4$
return N
```

Lemma 18. *Algorithm 1 (SpecialSuborder) is correct and runs in polynomial time in the input size.*

Proof. By the choice of N, $N \cdot \mathrm{M} \in \mathrm{M}_4(\mathbb{Z})$. Therefore,

$$\mathbb{Z} + N\mathcal{O}_0 = \mathbb{Z} + \mathsf{Span}_{\mathbb{Z}}(N\boldsymbol{\alpha}) = \mathbb{Z} + \mathsf{Span}_{\mathbb{Z}}(\boldsymbol{\beta}(N \cdot \mathrm{M})) \subseteq \mathcal{R}.$$

N is smallest by design.

Suppose the coefficients of a basis of $\mathcal{R}$ in terms of $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ are bounded by B (meaning their numerators and denominators are bounded by B when written in reduced forms). Since the operations involved in this algorithm are a constant number of multiplications, divisions and gcd computations of integers bounded by B and q, it runs in polynomial time in $\log(B)$ and $\log q$. □

Algorithm 1 gives us the desired inclusions $\mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{R} \subseteq \mathcal{O}$. Since we almost certainly deal with composite integers N, we need to make sure that this embedding is primitive. We rephrase the definition of *primitive* embedding from [Ler22a] here.

Definition 19. *The inclusion $\mathbb{Z}+N\mathcal{O}_0 \subseteq \mathcal{O}$ is primitive if it cannot be extended to any super order $\mathbb{Z} + N'\mathcal{O}_0$ for an integer N' that divides N.*

Recall that we do not have the knowledge of $\mathcal{O}$, therefore this is a priori not a trivial task to find a primitive inclusion, and in fact we need to work with the curve E. With the aid from the recent division algorithm first introduced by [Rob22], generalized in [MW23, Algorithm 1] and finally improved by [CLP24, Remark 13], it is now practical to divide an endomorphism by an arbitrary integer with computations involving isogenies between abelian surfaces. This is the essential ingredient of Algorithm 2. There, DIVIDE is an algorithm that on input an endomorphism θ and an integer ℓ, returns θ/ℓ if θ is divisible by ℓ, and $\perp$ otherwise.

Algorithm 2. PrimitiveSpecialSuborder(N)

```
Input: A maximal order $\mathcal{O}_0$ such that $\mathbb{Z}+N\mathcal{O}_0 \subseteq \mathcal{O} \subseteq \mathcal{B}_{p,\infty}$, an elliptic curve $E$, and
    an embedding $\iota : \mathrm{End}^0(E) \to \mathcal{B}_{p,\infty}$ such that $\iota(\mathrm{End}(E)) = \mathcal{O}$
Output: $N'$ such that the inclusion $\mathbb{Z} + N'\mathcal{O}_0 \subseteq \mathcal{O}$ is primitive
$\alpha_1, \alpha_2, \alpha_3, \alpha_4 \leftarrow$ a $\mathbb{Z}$-basis of $\mathcal{O}_0$
$\alpha_{1,E}, \alpha_{2,E}, \alpha_{3,E}, \alpha_{4,E} \leftarrow \iota(N\alpha_1), \iota(N\alpha_2), \iota(N\alpha_3), \iota(N\alpha_4)$
$N = \prod_{i=1}^{r} \ell_i^{e_i}$
$N' = N$
for $i = 1, \cdots, r$ do
    while none of $\alpha_{i,E} = \perp$ for $i = 1, 2, 3, 4$ do
        for $i = 1, 2, 3, 4$ do
            $\alpha_{i,E} \leftarrow$ DIVIDE($\alpha_{i,E}, \ell_i$) ([MW23, Algorithm 1])
        end for
        $N' = N/\ell_i$
    end while
end for
return $N'$
```

Lemma 20. *Algorithm 2 is correct and runs in polynomial time in the size of the input.*

Proof. The correctness is clear since the inclusion $\mathbb{Z} + N'\mathcal{O}_0 \subseteq \mathcal{O}$ is primitive if and only if the embedding $\iota : \mathbb{Z} + N'\mathcal{O}_0 \hookrightarrow \mathrm{End}(E)$ can not be extended to any super order of the form $\mathbb{Z} + N''\mathcal{O}_0$ with $N'' \mid N'$, and such an extension cannot be made if and only if there exists an element $\alpha \in \mathcal{O}_0$ such that $\iota(N'\alpha)$ is not divisible by any $N'' \mid N'$. There are at most $\log(N)$ iterations and each loop takes time polynomial in input size. Therefore overall this algorithm runs in polynomial time in the size of the input. □

Given the embedding of $\mathbb{Z} + N\mathcal{O}_0$ to $\mathcal{O}$, we now show the existence of an ideal I that is a left $\mathcal{O}_0$-ideal and a right $\mathcal{O}$-ideal. While the main ideas in the proof of Proposition 21 are from the proof of [Ler22a, Appendix A, Proposition 24], Lemma 22 and Lemma 23 are the details we add for our use cases.

Proposition 21. *Let the inclusion $\mathbb{Z}+N\mathcal{O}_0 \subseteq \mathcal{O}$ be primitive, then there exists a cyclic ideal I that is a left $\mathcal{O}_0$-ideal and a right $\mathcal{O}$-ideal of norm N.*

Proof. By Lemma 22, $p \nmid N$, otherwise the inclusion is not primitive. For each prime factor ℓ of N, the inclusion can be rewritten as

$$\mathbb{Z} + \ell(\mathbb{Z} + (N/\ell)\mathcal{O}_0) \subseteq \mathcal{O}.$$

By [Ler22a, Lemma 3], since $\ell \neq p$ and $\mathbb{Z} + (N/\ell)\mathcal{O}_0 \not\subseteq \mathcal{O}$ because of the primitivity of the inclusion, there exists an ideal I of norm ℓ such that $\mathcal{O}_L(I) = \mathcal{O}$, $\mathcal{O}_R(I) = \mathcal{O}'$ and $\mathcal{O}'$ contains $\mathbb{Z} + (N/\ell)\mathcal{O}_0$. According to Lemma 23, this is again a primitive inclusion. Therefore, we can apply [Ler22a, Lemma 3] on this inclusion again. In the end, we will have a sequence of ideals

$$I_1, I_2, \cdots, I_k$$

of norms corresponding to each prime factors of N. These ideals are compatible (meaning that $\mathcal{O}_R(I_i) = \mathcal{O}_L(I_{i+1})$ for $i = 1, \ldots, k-1$) and their multiplication is an ideal of norm N whose left order is $\mathcal{O}$ and right order is $\mathcal{O}_0$, which we still denote by I by an abuse of notation. This ideal I is cyclic since otherwise the inclusion $\mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{O}$ will not be primitive. The conjugate ideal $\bar{I}$ is then a cyclic ideal that is a left $\mathcal{O}_0$-ideal and a right $\mathcal{O}$-ideal of norm N as in the statement of the proposition.

□

Lemma 22. *Suppose $\mathbb{Z}+p^kM\mathcal{O}_0 \subseteq \mathcal{O}$ for some positive integers k and M such that $p \nmid M$, then $\mathbb{Z} + M\mathcal{O}_0 \subseteq \mathcal{O}$.*

Proof. By the local to global principle of $\mathbb{Z}$-lattices [Voi21, Theorem 9.1.1], it suffices to show that $(\mathbb{Z}+M\mathcal{O}_0) \otimes \mathbb{Z}_v \subseteq \mathcal{O} \otimes \mathbb{Z}_v$ holds for all places v. Let $v \neq p$, then

$$(\mathbb{Z} + p^k M\mathcal{O}_0) \otimes \mathbb{Z}_v = (\mathbb{Z} + M\mathcal{O}_0) \otimes \mathbb{Z}_v \subseteq \mathcal{O} \otimes \mathbb{Z}_v.$$

When $v = p$, since $\mathcal{B}_{p,\infty}$ ramifies at p, $\mathcal{B}_{p,\infty} \otimes \mathbb{Q}_p$ is a division ring over $\mathbb{Q}_p$ and it has only one unique maximal order. Therefore, $(\mathbb{Z} + M\mathcal{O}_0) \otimes \mathbb{Z}_p \subseteq \mathcal{O} \otimes \mathbb{Z}_p$. □

Lemma 23. *Let $\mathcal{O}', N$ and ℓ be as in Proposition 21 and its proof, then the inclusion $\mathbb{Z} + (N/\ell)\mathcal{O}_0 \subseteq \mathcal{O}'$ is primitive.*

Proof. By construction from the proof of [Ler22a, Lemma 3],

$$I = \{x \in \mathcal{O} \mid x(\mathbb{Z} + (N/\ell)\mathcal{O}_0) \subseteq \mathcal{O}\}.$$

Assume by contradiction that $\mathbb{Z}+(N/\ell\ell')\mathcal{O}_0 \subseteq \mathcal{O}'$ for some prime $\ell' \mid N/\ell$. Since I has norm ℓ, the integer $\ell \in I$. Therefore by the definition of I,

$$\ell(\mathbb{Z} + (N/\ell\ell')\mathcal{O}_0) \subseteq \mathcal{O},$$

which implies that

$$\mathbb{Z} + (N/\ell')\mathcal{O}_0 \subseteq \mathcal{O}.$$

This contradicts the assumption that the inclusion of $\mathbb{Z}+N\mathcal{O}_0$ into $\mathcal{O}$ is primitive. □

Finally, in Proposition 24 we show the existence of an isogeny to the curve E with the properties we desire. Although we believe that this result is implicitly used in [Ler22a], we were unable to find an explicit statement or proof.

Proposition 24. *Let $\mathcal{O}_0$ be a special extremal order as defined in Sect. 2.2 and $E_0/\mathbb{F}_p$ a special curve w.r.t. $\mathcal{O}_0$ (Remark 15) such that* $\mathrm{End}(E_0) \cong \mathcal{O}_0$ *via isomorphism ι_0 (i.e., $\iota_0(\mathcal{O}_0) = \mathrm{End}(E_0)$). Let the inclusion $\mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{O}$ be primitive. Then there exists a cyclic degree N isogeny φ from E_0 to E such that*

$$[\varphi]_\star\iota_0(\beta) = \iota(\beta) \text{ or } [\varphi]_\star{\iota_0}^{(p)}(\beta) = \iota(\beta), \text{ for any } \beta \in \mathbb{Z} + N\mathcal{O}_0 \text{ that satisfies } (n(\beta), N) = 1,$$

where $\iota_0^{(p)}$ is the Frobenius conjugate of ι_0.

Note that it still holds that $\iota_0^{(p)}(\mathcal{O}_0) = \mathrm{End}(E_0)$ since E_0 is defined over $\mathbb{F}_p$.

Proof. Let I be the ideal from Proposition 21. Let us define $E' := E/E[I]$ where $E[I] = \cap_{\alpha \in I \subset \mathcal{O}} \ker \iota(\alpha)$, and let $\varphi_I : E \to E'$ be the induced isogeny. Then φ_I induces an isomorphism ι_I between $\mathrm{End}^0(E') := \mathrm{End}(E') \otimes_{\mathbb{Z}} \mathbb{Q}$ and $\mathrm{End}^0(E) := \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$ via the map $\theta \mapsto \frac{\varphi_I \circ \theta \circ \varphi^{-1}}{N}$. We define $\iota' = \iota \circ \iota_I$ as in the following diagram:

$$\begin{tikzcd}
\mathrm{End}^0(E) \arrow[rr, "\iota_I"] & & \mathrm{End}^0(E') \\
& \mathcal{B}_{p,\infty} \arrow[ul, "\iota"] \arrow[ur, "\iota'"'] &
\end{tikzcd}$$

By [Wat69, Proposition 3.9], $\iota'^{-1}(\mathrm{End}(E'))$ equals the right order of I, which is $\mathcal{O}_0$. Since $\mathrm{End}(E') \cong \mathrm{End}(E_0) \cong \mathcal{O}_0$ and $\mathcal{O}_0$ is special extremal, by the Deuring correspondence, there exists an isomorphism $\Phi : E' \to E_0$, and this induces an isomorphism $\iota_\Phi : \mathrm{End}^0(E') \to \mathrm{End}^0(E_0)$ such that $\iota_\Phi^{-1}(\mathrm{End}(E_0)) = \mathrm{End}(E')$, and an embedding $\iota'_0 = \iota' \circ \iota_\Phi : \mathrm{End}^0(E_0) \to \mathcal{B}_{p,\infty}$, as in the following diagram:

$$\begin{tikzcd}
\mathrm{End}^0(E) \arrow[rr, "\iota_I"] & & \mathrm{End}^0(E') \arrow[rr, "\iota_\Phi"] & & \mathrm{End}^0(E_0) \\
& \mathcal{B}_{p,\infty} \arrow[ul, "\iota"] \arrow[ur, "\iota'"'] \arrow[urrr, "\iota'_0"'] & & &
\end{tikzcd}$$

It can be verified that $\iota'_0(\mathcal{O}_0) = \iota' \circ (\iota_\Phi(\mathcal{O}_0) = \iota'(\mathcal{O}_0) = \mathrm{End}(E_0)$. Therefore, $\iota'_0 = \iota_0$ or $\iota_0^{(p)}$.

Let us define an isogeny $\varphi = \hat{\varphi}_I \circ \Phi^{-1} : E_0 \to E$. Clearly $\deg \varphi = N$. Let $\beta \in \mathbb{Z} + N\mathcal{O}_0$ such that $(n(\beta), N) = 1$, then

$$[\iota'_0(\beta)]_* \varphi = \varphi$$

since $\iota'_0(\beta)(\ker \varphi) = \ker \varphi$. This is because $N = \deg \varphi$ and $\iota'_0(\beta)(\ker \varphi)$ is actually a scalar multiplication on $(\ker \varphi)$ by some scalar coprime to N. This implies that

$$[\varphi]_\star \iota'_0(\beta) = \frac{\varphi \circ \iota'(\beta) \circ \hat{\varphi}}{N} = \iota(\beta)$$

according to the diagram above. Finally, $\iota'_0 = \iota_0$ or $\iota_0^{(p)}$ implies the statement in the proposition. □

3.2 Deriving a Weak Isogeny Representation

In this section, we show that the information we have on φ suffices to develop a weak isogeny representation on it: we make this precise in Proposition 25. The main ideas in the proof of Proposition 25 are from [Ler22a]. While the proof in [Ler22a] contains the necessary elements, its presentation lacks a unified structure. For the ease of reading, we provide a more systematic proof with more details added. Moreover, [Ler22a] considers only the case when the degree N is a prime and $N \neq p$, and we generalize it to an arbitrary degree N not divisible by p.

Proposition 25. *Let $E, \mathcal{R}_E, \iota, \mathcal{R}, \mathcal{O}$ be as in Sect. 3.1. Let $\mathcal{O}_0, E_0, \iota_0$ and $\varphi : E_0 \to E$ be of degree N as in Proposition 24 where $p \nmid N$ (Lemma 22), WLOG, we may assume it holds that $[\varphi]_\star \iota_0(\beta) = \iota(\beta)$ for $\beta \in \mathbb{Z} + N\mathcal{O}_0$ and $(n(\beta), N) = 1$ (otherwise, we replace ι_0 with $\iota_0^{(p)}$). There is a quantum algorithm that on input P, a point on E_0 such that $(\mathrm{ord}(P), N) = 1$, computes $\langle \varphi(P) \rangle$ in expected polynomial time in $\log \mathrm{ord}(P)$, $\log N$, and $|P|$ where $|P|$ is the bitsize of the representation of P.*

Proof. Let J be a left $\mathcal{O}_0$ ideal such that $E_0[J] = \langle P \rangle$. This ideal can be computed in quantum polynomial time in the size of inputs (i.e., in $\log \mathrm{ord}(P)$ and $|P|$) by [Ler22b, Algorithm 20]. Let us use $\bar{N}$ to denote $\mathrm{ord}(P)$, then $n(J) = \bar{N}$. Suppose one can find an element $\beta \in (\mathbb{Z} + N\mathcal{O}_0) \cap J$ with $n(\beta) = \bar{N}S$ for some integer S such that $(N\bar{N}, S) = 1$, this implies that $\ker \iota_0(\beta) \cap E_0[\bar{N}] = E_0[J]$. Therefore

$$\varphi(\langle P\rangle) = \varphi(E_0[J]) = \varphi(\ker\iota_0(\beta)\cap E_0[\bar{N}]) = \varphi(\ker\iota_0(\beta))\cap E[\bar{N}] = \ker([\varphi]_*\iota_0(\beta))\cap E[\bar{N}],$$

where the second last equality holds since $\ker\varphi\cap\ker\beta = \emptyset$ as our conditions on $\bar{N}, S$ imply that $(N, \bar{N}S) = 1$. Then we have

$$\ker([\varphi]_*\iota_0(\beta))\cap E[\bar{N}] = \ker\iota(\beta)\cap E[\bar{N}]$$

by our condition on φ. Since $\beta \in \mathbb{Z} + N\mathcal{O}_0 \subseteq \mathcal{R}$, we know an efficient representation of $\iota(\beta)$ as we know a $\mathbb{Z}$-basis (given as efficient representations of its elements) of $\iota(\mathcal{R}) = \mathcal{R}_E$ which is the given full rank suborder in $\mathrm{End}(E)$. Therefore, the subgroup $\ker\iota(\beta)\cap E[\bar{N}]$ can be computed in polynomial time in the input sizes (i.e., in $\log n(\beta)$ and $|P|$ since the bitsize of the representation of $E[\bar{N}]$ is in $O(|P|)$). The only missing piece to complete the proof is to show that there is an efficient algorithm that finds $\beta \in (\mathbb{Z} + N\mathcal{O}_0)\cap J$ with $n(\beta) = \bar{N}S$ such that $(N\bar{N}, S) = 1$. This is done in Proposition 26.

Note that here we have $n(\beta) = \mathrm{poly}(p)$ and $n(J) = \mathrm{ord}(P)$. Then the claimed runtime follows from that of Proposition 26 and the time complexity of evaluating $\iota(\beta)$ on $E[\bar{N}]$. □

Solving Norm Equation in $(\mathbb{Z}+N\mathcal{O}_0)\cap J$*.* We now talk about the missing piece in the proof of Proposition 25. [Ler22a] introduced an algorithm (IdealSuborderNormEquation [Ler22a, Algorithm 7]) for solving norm equations in $(\mathbb{Z}+DI)\cap J$ for a prime $D \neq p$. For us, we consider an easier lattice $(\mathbb{Z}+N\mathcal{O}_0)\cap J$ which allows us to simplify the algorithm a little bit, but relax N to be an arbitrary integer different from p. The main obstacle in obtaining the generalization in terms of N lies in the inputs requirements in an algorithm called StrongApproximation. Fortunately, a generalized version was introduced in [CII+23a, Algorithm 2] and that leads to GenStrongApproximation as explained in Sect. 2.5.

Proposition 26 (heuristic). *Algorithm 3 takes in input an integer N and an ideal J such that $(N, n(J)) = 1$, and outputs an element $\beta \in (\mathbb{Z} + N\mathcal{O}_0)\cap J$ with $(n(\beta)/n(J), n(J)N) = 1$ and $n(\beta)/n(J)$ in $O(\mathrm{poly}(pNn(J)))$, in expected polynomial time in $\log p$, $\log N$ and $\log n(J)$ with overwhelming probability.*

Proof. See [Ler22a, Algorithm 7] and [CII+23a, Lemma 5.6]. □

Remark 27. Most of the heuristics involved in Proposition 26 are the "common" KLPT-type heuristics that first appeared in [KLPT14], specifically, they are from the subroutines RepresentInteger and GenStrongApproximation. For the first one, we do not state the heuristic here as it is widely used subroutine. For the second algorithm, we give more discussions of the heuristics involved in [CP25, Appendix A].

Algorithm 3. IdealSuborderNormEquation(N, J)

```
Input: An integer $N$, and a left $\mathcal{O}_0$-ideal $J$ such that $(N, n(J)) = 1$.
Output: $\beta \in (\mathbb{Z} + N\mathcal{O}_0) \cap J$ of norm $n(J)S$ for some integer $S$ such that $(n(J)N, S) =$
    1.
 1: $\bar{N} = n(J)$
 2: Select a random integer $M$ such that $M > p$, $\gcd(M, \bar{N}^2) = \bar{N}$, $(M, N) = 1$.
 3: $\mu = \mathsf{RepresentInteger}_{\mathcal{O}_0}(M)$
 4: $(C_1 : D_1) = \mathsf{EichlerModConstraint}(\mu, \mathcal{O}_0 \mathsf{N} \mathcal{O}_0)$
 5: $(C_2 : D_2) = \mathsf{IdealModConstraint}(\mu, J)$
 6: $C = \mathsf{CRT}_{N,\bar{N}}(C_1, C_2)$, $D = \mathsf{CRT}_{N,\bar{N}}(D_1, D_2)$
 7: if $(pf(C, D), \bar{N}N) \neq 1$ then
 8:     Go to Step 2
 9: end if
10: $\tilde{\mu} = \mathsf{GenStrongApproximation}(N\bar{N}, C, D, \mathrm{False})$
11: return $\mu\tilde{\mu}$
```

4 Resolution of the Powersmooth Quaternion Lifting Problem

In this section, we present a new algorithm (Algorithm 4) that solves the Powersmooth Quaternion Lifting Problem (Problem 7) for any odd integer N. A solution to this problem was previously given in [CII+23a, Section 5], under the assumption that N is odd and has $O(\log \log p)$ many distinct prime factors. Our algorithm extends the approach from [CII+23a, Section 5]. Before we proceed, we point out that as done in [CII+23a], WLOG we may assume that in Problem 7:

1. $\widetilde{\mathcal{O}}$ is one of the special extremal orders $\mathcal{O}_0$ defined in Sect. 2.2. This is because we can find a connecting isogeny from $\widetilde{\mathcal{O}}$ to $\mathcal{O}_0$ of degree coprime to N. We refer the readers to [CII+23a, Section 5.2] for more details.
2. σ_0 is in the suborder $R + \mathbf{j}R \subseteq \mathcal{O}_0$. This is because $\Delta_R \sigma_0 = \Delta_R \sigma_0 \bmod N\mathcal{O}_0$ satisfies that $\Delta_R \sigma_0 \in R + R\mathbf{j}$ (due to the fact that $[\mathcal{O}_0 : R + R\mathbf{j}] = \Delta_R$) and $(\Delta_R, N) = 1$ (by our choices of q Remark 9 and the fact that N is odd), hence it suffices to lift $\Delta_R \sigma_0$.

The obstacle in extending the results in [CII+23a, Section 5] to arbitrary odd integers N is that the chance of a random integer being a square modulo N is exponentially low when N has $O(\log p)$ many factors. We resolve this issue by designing a recursive algorithm so that we can deal with "nice" factors of N at each iteration with the factors increasing in size. We note that a recursive approach to solving Problem 7 in all generality was also sketched in [CII+23b, Appendix D.3]; while the high level idea of our algorithm is inspired by this one, we modify several steps to either make it work or simplify it, and we provide algorithms for the necessary subroutines.

4.1 The Main Recursive Algorithm

The main ideas in Algorithm 4 are the following: Let N be an arbitrary odd integer with known factorization (say obtained by running Shor's quantum algorithm [Sho97]). Starting with $s = 0$, $N_0 = 1$ and continuing with increasing values of s, our algorithm will recursively compute an integer $N_s > N_{s-1}$, $N_{s-1}|N_s|N$ and a "lift" σ_s of σ_0 such that $\sigma_s = \lambda_s \sigma_0 \bmod N_s\mathcal{O}$, where $\lambda_s \in (\mathbb{Z}/N_s\mathbb{Z})^\times$ and $n(\sigma_s)$ is powersmooth. It is clear that one must call Algorithm 4 at most $O(\log N)$ times to solve Problem 7; in fact we argue in the proof of Theorem 30 that at most $O(\log \omega(N))$ calls will be needed on average.

Algorithm 4. RecursivePowerSmoothQuaternionLift$(\sigma_0, N, N_s, \lambda_s, \sigma_s)$

```
Input: $\sigma_0$, $N$, $\sigma_s$, $N_s$ and $\lambda_s \in \mathbb{Z}$ such that $\sigma_s = \lambda_s\sigma_0 \bmod N_s\mathcal{O}_0$ and $n(\sigma_s)$ powersmooth and $(n(\sigma_s), N_s) = 1$.
Output: $N_{s+1} > N_s$ such that $N_s|N_{s+1}|N$; $\lambda_{s+1} \in (\mathbb{Z}/N_{s+1}\mathbb{Z})^\times$ and $\sigma_{s+1}$, $N_{s+1}$ such that $\sigma_{s+1} = \lambda_{s+1}\sigma_0 \bmod N_{s+1}\mathcal{O}_0$ and $n(\sigma_{s+1})$ powersmooth, and $(n(\sigma_{s+1}), N_{s+1}) = 1$.
1: $i = 1$
2: Let $\sigma_0' = \sigma_0\bar{\sigma}_s(n(\sigma_0))^{-1} \bmod N\mathcal{O}_0$
3: Generate $\gamma \in \mathbb{Z} + N_s\mathcal{O}_0$ with powersmooth norm $S_{\gamma,s,i}$ coprime to $N$
4: Find $N_{s+1}|N$ maximal and $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{Z} + N_s\omega\mathbb{Z}$ such that
       $$\sigma_0' = \alpha_1\gamma\alpha_2\bar{\gamma}\alpha_3 \bmod N_{s+1}\mathcal{O}_0$$
5: for $r = 1, 2, 3$ do
6:     $\alpha_r = C_r + D_r\omega$
7:     $\beta_r =$ GenStrongApproximation$_{\mathcal{N}}(N_{s+1}, C_r, D_r, \mathrm{True})$
8: end for
9: Compute $\sigma_{s+1} = \sigma_0'\beta_1\mathbf{j}\gamma\beta_2\mathbf{j}\bar{\gamma}\beta_3\mathbf{j}$
10: if $i = 1$ then
11:     Let $i = 2$ ; let $N_s = N_{s+1}$ ; let $\sigma_s = \sigma_{s+1}$
12:     Go to Step 2
13: else
14:     Let $\sigma_{s+1} = p^{-6}\sigma_{s+1}$
15:     Compute $\lambda_{s+1}$ such that $\sigma_{s+1} = \lambda_{s+1}\sigma_0 \bmod N_{s+1}\mathcal{O}_0$
16: end if
17: return $N_{s+1}$, $\lambda_{s+1}, \sigma_{s+1}$
```

Remark 28. Algorithm 4 is strongly influenced by [CII+23a], which includes a decomposition similar to Step 4 and also uses a lifting algorithm of elements in $\mathbf{j}(\mathbb{Z} + \omega\mathbb{Z})$ from [CII+23a]. Our decomposition is however slightly different: we choose the α_r elements in $\mathbb{Z} + N_s\omega\mathbb{Z}$ instead of $\mathbf{j}(\mathbb{Z} + \omega\mathbb{Z})$ and we use $\bar{\gamma}$ instead of γ as the fourth factor in the decomposition; the element γ generated in Step 3 is also chosen from the suborder $\mathbb{Z}+N_s\mathcal{O}_0$ instead of $\mathcal{O}_0$. These changes facilitate a recursive procedure and analysis, as the choice $\alpha_1 = \alpha_2 = \alpha_3 = 1$

now provides a solution to the decomposition modulo N_s, so that we are only left with considering the remaining factors of N_{s+1}/N_s. Because α_r are now chosen in $\mathbb{Z} + N_s\omega\mathbb{Z}$ instead of $\mathbf{j}(\mathbb{Z} + \omega\mathbb{Z})$, in Step 7 we lift the elements $\alpha_r\mathbf{j}$ instead of α_r, and compensate for the extra factors $\mathbf{j}$ in Step 9. Note that the quaternion σ_{s+1} produced in Step 9 can always be re-written as a product of a powersmooth norm element and a power of $\mathbf{j}$; we repeat Steps 2 to 9 of the algorithm twice to make sure that this power is even, hence can be brought into the constant factor λ_{s+1}.

4.2 Subroutine Algorithms

Algorithm 4 explicitly or implicitly uses several subroutines to solve Steps 3, 4 and 7. For Step 7 one can use the GenStrongApproximation ([CP25, Algorithm 7]). We now provide subroutines for Steps 3 and 4, as well as another subroutine to generate powersmooth numbers.

In the decomposition used in [CII+23a], γ is a powersmooth element of $\mathcal{O}_0$, and they could simply use a subroutine from the KLPT algorithm. Here in Step 3, γ is restricted to the suborder $\mathbb{Z} + N_s\mathcal{O}_0$, and we introduce Algorithm 5 for this purpose. The algorithm restricts the search to an element γ of the form $\gamma = w + yN_s\mathbf{i} + zN_s\mathbf{k}$, such that $n(\gamma) = w^2 + N_sp(y^2 + z^2)$ determines $\pm w \bmod N_sp$. We then fix a suitably large and powersmooth norm S, and try various options for w until $S - w^2$ can be decomposed as a sum of two squares (using Cornacchia's algorithm [Cor08]). Assuming the proportion of such numbers is as would be expected for random numbers of the same size, one can expect the algorithm to correctly terminate with an element of norm only slightly larger than $N_s^2p^2$.

Algorithm 5. SubOrderRepresentInteger(N_s)

```
Input: $N_s$
Output: $\gamma \in \mathbb{Z} + N_s\mathcal{O}_0$ with powersmooth norm with $n(\gamma)$ coprime with $Np$.
1: Fix $S$ powersmooth, large enough, coprime with $Np$ and square modulo $N_sp$
2: Fix $w$ such that $w^2 = S \bmod N_sp$
3: Find small $k$ and $y, z$ such that $y^2 + z^2 = (S - (w + kN_sp)^2)/N_sp$
4: return $\gamma = (w + kN_sp) + yN_s\mathbf{i} + zN_s\mathbf{k}$
```

The decomposition to perform in Step 4 is similar (though not identical, as discussed in Remark 28) to the one performed by Algorithm 4 of [CII+23a], and it can be solved in a similar way. Recall that $R := \mathbb{Z} + \omega\mathbb{Z}$.

Lemma 29. *Let $\sigma_0' = A + B\mathbf{j}$ and $\gamma = C + D\mathbf{j}$ with $A, B, C, D \in R$, and let $N_s|N_{s+1}|N$ be integers, with* $\gcd(n(\gamma), N) = 1$. *Assume that*

$$\Delta_R[(n(B))^2(n(C) - pn(D))^2 - 4n(ABCD)]$$

is a quadratic residue modulo all primes dividing N_{s+1}/N_s. Then the equation

$$\sigma_0' = \alpha_1\gamma\alpha_2\bar{\gamma}\alpha_3 \bmod N_{s+1}\mathcal{O}_0$$

has a solution $\alpha_1, \alpha_2, \alpha_3 \in (\mathbb{Z} + N_s\omega\mathbb{Z})$ *with* $\gcd(n(\alpha_3), N_{s+1}) = 1$*. Moreover, this solution can be computed in polynomial time.*

Proof. We follow the approach of [CII+23a]. The equation has solutions iff $\sigma_0'\alpha_4\gamma = \alpha_1\gamma\alpha_2$ has solutions where $\alpha_4 = \bar{\alpha}_3(n(\alpha_3\gamma))^{-1} \bmod N_{s+1}$. The later equation translates into

$$(A + B\mathbf{j})\alpha_4(C + D\mathbf{j}) = \alpha_1(C + D\mathbf{j})\alpha_2$$

or

$$\begin{cases} AC\alpha_4 - B\bar{D}p\bar{\alpha}_4 = C\alpha_1\alpha_2 \\ AD\alpha_4 + B\bar{C}\bar{\alpha}_4 = D\alpha_1\bar{\alpha}_2, \end{cases} \tag{1}$$

where here and often below we omit to explicitly write $\bmod (N_{s+1}\mathcal{O}_0)$ and mod $(N_{s+1}R)$ in the equations for the sake of conciseness.

Taking the norms on both sides of both equations, we see that the system has solutions only if there exists α_4 such that

$$n(D)n(AC\alpha_4 - B\bar{D}p\bar{\alpha}_4) = n(C)n(AD\alpha_4 + B\bar{C}\bar{\alpha}_4).$$

The converse is also true (see Lemma 5.8 in [CII+23a]): when α_4 satisfies the above equation then we can compute $\alpha_5 \in R$ such that $\alpha_5/\bar{\alpha}_5 = \frac{D(AC\alpha_4 - B\bar{D}p\bar{\alpha}_4)}{C(AD\alpha_4 + B\bar{C}\bar{\alpha}_4)} \bmod N_sR$, then let $\alpha_1 = \frac{AC\alpha_4 - B\bar{D}p\bar{\alpha}_4}{C}\alpha_5 \bmod N_sR$ and $\alpha_2 = \frac{1}{D\alpha_5} \bmod N_sR$.

The above condition can be rewritten as

$$n(B)(n(D)^2p^2 - n(C)^2)n(\alpha_4) - \operatorname{tr}(A\bar{B}CD\alpha_4^2)(n(C) + pn(D)) = 0.$$

which further simplifies as

$$n(B)(n(C) - pn(D))n(\alpha_4) + \operatorname{tr}(A\bar{B}CD\alpha_4^2) = 0$$

since $\gcd(n(\gamma), N) = 1$ by assumption.

Let $\alpha_4 = \alpha_{4,0} + \alpha_{4,1}N_s\omega$. We note that the condition $\gcd(n(\alpha_4), N_{s+1}) = 1$ implies $\gcd(\alpha_{4,0}, N_s) = 1$, and that (potentially loosing some solutions) we can restrict to the existence of a solution with $\alpha_{4,0} = 1$.

Let $E := n(B)(n(C) - pn(D))$ and $F := A\bar{B}CD = F_0 + F_1\omega$. The condition above then translates into the quadratic equation

$$C_2\alpha_{4,1}^2 + C_1\alpha_{4,1} + C_0 = 0$$

where

$$\begin{cases} C_2 := N_s^2\left((\operatorname{tr}(\omega))^3F_1 + (\operatorname{tr}(\omega))^2F_0 - 3\operatorname{tr}(\omega)n(\omega)F_1 + n(\omega)E - 2n(\omega)F_0\right), \\ C_1 := N_s\left(2(\operatorname{tr}(\omega))^2F_1 + \operatorname{tr}(\omega)E + 2\operatorname{tr}(\omega)F_0 - 4n(\omega)F_1\right), \\ C_0 := \operatorname{tr}(\omega)F_1 + E + 2F_0. \end{cases}$$

This equation has solutions if and only if its discriminant

$$\Delta := \Delta_R N_s^2(E^2 - n(F))$$

is a square, or equivalent if Δ is a square modulo all prime numbers dividing N. Finally, we note that the condition is clearly satisfied for all primes dividing N_s: indeed modulo these primes we can simply choose $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

Powersmooth Number Generation. Our algorithm requires powersmooth numbers in its subroutines Algorithm 5. These numbers must additionally be coprime with Np, and satisfy additional quadratic residuosity assumptions. For our purposes, it is sufficient that the powersmooth bound is polynomial in $\log p$ and $\log N$.

These numbers can easily be generated as follows. Fix a number $r \in O(1)$, and initiate a value $\ell_{min} = 3$, representing the smallest prime dividing the next number. At each call for a new powersmooth number, select the $\lceil r \log p \rceil$ smallest primes $\{\ell_1 = \ell_{min}, \ell_2, \ldots, \ell_R\}$ larger or equal to ℓ_{min} which are also coprime to Np; set ℓ_{min} to the immediately next prime coprime to Np; choose exponents $e_1, \ldots, e_R$ such that $S = \prod \ell_i^{e_i}$ satisfies all constraints. In Algorithm 5, one simply choose all exponents even to make sure the number is a square integer.

4.3 Proof of the Resolution of the PQLP

Theorem 30 (heuristic). *There exists a probabilistic classical algorithm of expected polynomial time in* $\log p$ *and* $\log N$*, returning solutions to Problem 7 with norm* S *such that both* $\log S$ *and the largest power of prime factor of* S *are polynomial in* $\log p$ *and* $\log N$*.*

Proof. The algorithm repeats Algorithm 4 until $N_{s+1} = N$. We argue that (under plausible assumptions) each subroutine runs in polynomial time, and that only $O(\log \log N)$ calls to this algorithm will be needed. Throughout, we assume that the factorization of N is known.

Step 3 of Algorithm 4 simply calls Algorithm 5. That algorithm is a variant of the RepresentInteger algorithm from KLPT [KLPT14]. For simplicity, Step 3 of Algorithm 5 can restrict the search to polynomially small values k such that $M := (S - (w + kN_s p)^2)/N_s p$ is a prime. The algorithm's correctness and polynomial time complexity rely on M being a prime congruent to 1 modulo 4 with a non negligible probability. In practice, we can heuristically expect this to happen with the same probability as random integers of the same size, hence to only need to test polynomially many values for k. The powersmooth number S in Step 1 may be chosen as a square integer of value about $N_s^2 p^2$ times polylogarithmic factors to ensure existence of w in Step 2 and a positive value of M in Step 3.

By construction, the decomposition in Step 4 of Algorithm 4 has a solution $\alpha_1 = \alpha_2 = \alpha_3 = 1$ modulo N_s, so this step is really about finding a decomposition that also holds modulo other factors of N. This step can be implemented as follows: first compute the discriminant of Lemma 29. Next, find the largest factor $N_{s+1} | N$ such that the discriminant is a square modulo N_{s+1}. Finally, compute the solution $\alpha_1, \alpha_2, \alpha_3$ as in the proof of Lemma 29. For any fixed prime, a random number is a square modulo that prime with probability one half, so heuristically we can expect the discriminant to be a square modulo about half of the remaining factors of N.

As N has at most $\log N$ factors, this means we can heuristically expect that Algorithm 4 will be called at most $O(\log \log N)$ times.

Step 7 of Algorithm 4 can be implemented with [CP25, Algorithm 7]. Its correctness and polynomial time complexity is implied by a corresponding result on [CP25, Algorithm 7] there, which is essentially [CII+23a, Lemma 5.6].

By Step 10 in Algorithm 4, we have

$$n(\sigma_{s+1}) = n(\sigma_s)(n(\gamma))^2 n(\beta_1)n(\beta_2)n(\beta_3).$$

Since $n(\gamma) \approx p^2N^2 \cdot \mathrm{poly}\log(N,p)$ and $n(\beta_i) \approx pN^4$, these imply that

$$\log n(\sigma_{s+1}) \leq \log n(\sigma_s) + 7\log p + 16\log N + \mathrm{polylog}(N,p)$$

and the last σ_s output by our algorithm is expected to have norm such that

$$\log n(\sigma_s) \leq (7\log p + 16\log N + \mathrm{polylog}(N,p))\log\log N$$

that is polynomial in $\log p$ and $\log N$. □

Remark 31. We summarize the heuristics we made in the proof of Theorem 30 here. Firstly, we have a KLPT-type heuristic which says that the right hand side of Step 3 of Algorithm 5 is a random integer of the same size. The second heuristic is implicitly in the statement of Lemma 29, we state it more clearly in [CP25, Appendix A]. Finally, since we make use of the algorithm GenStrongApproximation in [CP25, Appendix A], this theorem also assumes the heuristics used for this algorithm. We refer the readers to [CP25, Appendix A] for more discussions on these heuristics.

5 Proofs of the Main Theorems

Lemma 32. *Problem 5 reduces to the case when the endomorphism ring of $\widetilde{E}$ is isomorphic to a special maximal order defined in Sect. 2.2 and when $(N, 2p) = 1$.*

Proof. If $p \mid N$, then φ factors through the Frobenius isogeny $\pi_p : \widetilde{E} \to \widetilde{E}^{(p)}$. I.e., there exists $\varphi' : \widetilde{E}^{(p)} \to \widetilde{E}$ such that $\varphi = \varphi' \circ \pi_p$. One can deduce a weak isogeny representation of φ' from this composition, and an efficient representation of $\mathrm{End}(\widetilde{E}^{(p)})$ can be obtained from that of $\mathrm{End}(\widetilde{E})$. Therefore, WLOG, we may assume that $(N, p) = 1$.

If $2 \mid N$, given the weak isogeny representation of φ, we can find the unique order 2 point P (up to sign) such that $\varphi(P) = O_E$. Let ϕ_2 be the isogeny defined by the kernel $\langle P\rangle \subseteq \widetilde{E}[2]$ with codomain $\widetilde{E}'$. Then similarly as before, there exists $\varphi' : \widetilde{E}' \to E$ such that $\varphi = \varphi' \circ \phi_2$. We can compute a weak isogeny representation of φ' from the composition, and an efficient representation of $\mathrm{End}(\widetilde{E}')$ from that of $\mathrm{End}(\widetilde{E})$ and ϕ_2. We repeat this process until $(N, 2) = 1$ and this takes at most $\log N$ steps.

Finally, let $\mathcal{O}_0$ be a special maximal order (Sect. 2.2) and E_0 be a special curve w.r.t. $\mathcal{O}_0$ (Remark 15). Since we know both $\mathrm{End}(E_0)$ and $\mathrm{End}(\widetilde{E})$, we can compute a connecting path φ' from E_0 to $\widetilde{E}$ such that $(\deg\varphi', 2p) = 1$ using the KLPT algorithm from [KLPT14]. Clearly we have a weak isogeny representation of $\varphi \circ \varphi' : E_0 \to E$. □

Proof. (Proof of Theorem 3). By Lemma 32, we can assume that in the IsERP, the curve $\widetilde{E}$ is a special maximal curve E_0 w.r.t. a special maximal order $\mathcal{O}_0$, and we can also assume that the degree N of the isogeny is coprime to $2p$. In [CII+23a], the IsERP is reduced to the GAEP in quantum polynomial time, and then the GAEP is reduced to the PQLP in polynomial time. Therefore, to solve the IsERP, it suffices to solve the PQLP when the order $\widetilde{\mathcal{O}}$ is a special maximal order and when N is coprime to $2p$. This is precisely Theorem 30. □

Proof (Proof of Theorem 1). By Proposition 24 and Proposition 25, the problem of computing $\mathrm{End}(E)$ from $\mathcal{R}_E$ is reduced to an instance of the IsERP in time polynomial in $\log p$ and $\log \sqrt{\Delta_{\mathcal{R}}}/p$, where the latter dependence comes from the observation that N returned by Algorithm 2 is of size $O(\mathrm{poly} \log \Delta_{\mathcal{R}})$. Then the theorem is implied by Theorem 3. □

6 Implications to Isogeny-Based Cryptography

In this section, we talk about implications of our main results (Theorem 1 and Theorem 3) to isogeny-based cryptography. In Sect. 6.1, we explain the impact of our results to pSIDH key exchange protocol [Ler22a]. In Sect. 6.2 and Sect. 6.3, we study reductions between hard problems that are essential in isogeny-based cryptography. Even though polynomial time classical reductions between these hard problems have been established, we discover that our quantum algorithms can make these reductions *tighter*. Here, a *tighter* reduction means a *lower query complexity*, which is often beneficial in practice and will in turn allow to select smaller parameters while preserving security guarantees.

6.1 pSIDH Cannot Be Repaired by Changing the Degree

pSIDH is a key exchange protocol proposed by Leroux in 2022 [Ler22a]. The public key is a weak isogeny representation of an isogeny of degree D from a public curve, where D is a large prime. The private key is the quaternion ideal corresponding to the public key isogeny under the Deuring correspondence, and equivalently, the endomorphism ring of the codomain curve. The original pSIDH proposal was broken by [CII+23a] since the IsERP which lies under the security of pSIDH is solved by a quantum algorithm presented in [CII+23a] when the degree D is an odd integer with $O(\log \log p)$ many distinct factors. In light of that result and its limitations, one could hope to repair pSIDH by choosing an integer D that has lots of factors, but our Theorem 3 implies that such a modification is still insecure.

6.2 Quantum Reduction of the Supersingular Endomorphism Ring Problem to the One Endomorphism Problem

For a given supersingular elliptic curve E over a finite field $\mathbb{F}_{p^2}$, finding even one endomorphism on E has been believed to be a hard problem, known as the *one endomorphism problem*, or OneEnd. Clearly, the OneEnd problem reduces to the $\mathsf{EndRing}$ problem. In 2024, Page and Wesolowski [PW24] proved that the $\mathsf{EndRing}$ and OneEnd problems are equivalent, under probabilistic polynomial time classical reductions. For the convenience of discussion, [PW24] considers a bounded version of OneEnd as there exists arbitrary length endomorphisms. They define the OneEnd_λ problem to be the OneEnd problem where the solution α is required to satisfy $\log(\deg \alpha) \leq \lambda(\log p)$ for a function $\lambda : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$.

In their algorithm of turning an oracle $\mathscr{O}$ for $\mathsf{OneEnd}_{\,\lambda}$ into an $\mathsf{EndRing}$ algorithm ([PW24, Algorithm 3]), even though not explicitly stated, it is clear from the algorithm that polynomially many queries to the OneEnd_λ oracle $\mathscr{O}$ are needed. We look into the details of their algorithm [PW24, Algorithm 3] and the complexity analysis [PW24, Theorem 7.2] to give a more concrete estimate on the query complexity to the best of our knowledge.

Most of the queries to the oracle $\mathscr{O}$ happen during the second loop of [PW24, Algorithm 3], which is from Step 11 to Step 28. Let $N = N_t$ be as defined in Step 10 of [PW24, Algorithm 3], in the worst case, $N = [\mathrm{End}(E) : R]$, the index computed in Step 9. Each iteration of the loop is a *success* if either Step 21 or Step 24 is reached, and the chance of this happening is $\Omega((\log N)^{-12})$. At the end of each success, the value N is updated to $N/2$ in the worst case. For this loop to finish, there should be at least $\log([\mathrm{End}(E) : R])$ many successes. This implies that in the worst case, the number of queries to $\mathscr{O}$ is

$$O\left(\sum_{i=0}^{\lfloor \log N \rfloor} (\log \frac{N}{2^i})^{12}\right) = O(\log^{13} N) \leq O((\log(p\lambda \log p))^{13}),$$

where the inequality that $[\mathrm{End}(E) : R] \leq 2^{O(\log(p\lambda \log p))}$ is from the proof of [PW24, Theorem 7.2]. To put this into perspective, in the case when λ is the scalar multiplication map by a small constant, and when p is a prime of 256-bits, $(\log(p\lambda \log p))^{13} \approx (2^{16})^{13} = 2^{208}$. Ignoring the possible effects from the hidden constant in O, this implies that 2^{208} queries to the oracle $\mathscr{O}$ are needed in the worst case. In the potentially better case when the index $[\mathrm{End}(E) : R]$ from Step 9 only contains a few large factors, despite the reduced number of successes needed, the number of queries is still $O((\log p\lambda(\log p))^{12})$, which is roughly 2^{192} using the previously suggested values for p and λ. While [PW24] is of great theoretical importance, its inefficiency means it can only have a limited effect on practical security analysis.

In what follows, we show that the expected number of queries can be made to $O(1)$ with our quantum algorithm (Algorithm 6). This algorithm essentially borrows the first 6 steps of [PW24, Algorithm 3] and replaces the remaining steps with our quantum algorithm. Using the analysis in [PW24, Theorem 7.2], this constant number of queries is in fact 16.

Theorem 33 (heuristic). *(EndRing reduces to OneEnd) Algorithm 6 is a quantum reduction from $\mathsf{EndRing}$ to $\mathsf{OneEnd}_{\,\lambda}$ with an expected polynomial runtime in* $\log p$ *and* $\lambda(\log p)$*, and* $O(1)$ *number of classical queries to the* OneEnd_λ *problem oracle* $\mathscr{O}$.

Algorithm 6. Turning an oracle $\mathscr{O}$ for OneEnd_λ into an $\mathsf{EndRing}$ algorithm

```
Input: A supersingular elliptic curve $E/\mathbb{F}_{p^2}$ and access to an oracle $\mathscr{O}$ that solves the $\mathsf{OneEnd}_\lambda$ problem.
Output: The endomorphism ring $\mathrm{End}(E)$.
$k \leftarrow \left\lceil \frac{\log\left(12.9\cdot(1+\sqrt{3})\cdot\sqrt{p}+13\right)}{\log\left(\frac{3}{2\sqrt{2}}\right)} \right\rceil$
$\mathcal{R} \leftarrow \mathbb{Z}$
while $\mathrm{rank}_{\mathbb{Z}}(\mathcal{R}) \neq 4$ do
    $\alpha \leftarrow \mathrm{Rich}_k^{\mathscr{O}}(E)$, a random endomorphism of $E$ [PW24, Algorithm 1]
    $\mathcal{R} \leftarrow$ the ring generated by $\mathcal{R}$ and $\alpha$
end while
Compute $\mathrm{End}(E)$ from $\mathcal{R}$ (Theorem 1)
return An efficient representation of $\mathrm{End}(E)$
```

Proof. Following the proof of [PW24, Theorem 7.2], the loop in Algorithm 6 requires $O(1)$ number of queries to $\mathscr{O}$, and it takes time polynomial in $\log p$ and $\lambda(\log p)$. The index $[\mathrm{End}(E) : \mathcal{R}] \leq 2^{O(\log p\cdot\lambda(\log p))}$, then from Theorem 1, the last step takes time polynomial in $\mathrm{poly}(\log p)$ and $\log([\mathrm{End}(E) : \mathcal{R}])$ on a quantum computer. □

2-Special Soundness of the SQIsign Identification Scheme. SQIsign is a post-quantum digital signature scheme proposed in [DFKL+20], known for having the most compact public keys and signatures among all known post-quantum constructions. Since its initial proposal, several different variants have been introduced [DLRW24, BFD+24, NO24, DF24], and all of them are constructed by applying the Fiat-Shamir transform to identification protocols.

One application of the reduction from $\mathsf{EndRing}$ to OneEnd is to prove the 2-special soundness property of the underlying identification protocols. For instance, in SQIsign2D-West [BFD+24], it is shown that if $e_{\mathrm{chl}} + e_{\mathrm{rsp}} \leq e$ (where $e_{\mathrm{chl}}, e_{\mathrm{rsp}}$ and e are fixed integers), then the identification protocol in [BFD+24] has the 2-special soundness property for the language

$$\{(E_{\mathrm{pk}}, \alpha) \mid \alpha \in \mathrm{End}(E_{\mathrm{pk}})\backslash\mathbb{Z} \text{ in efficient representation}\}.$$

This language corresponds to the *one endomorphism* problem. The classical reduction of the $\mathsf{EndRing}$ problem to the OneEnd problem shown in [PW24] increased the confidence in the hardness of this relation (equivalently, in the 2-special soundness of the identification protocol). However, as discussed in Sect. 6.2, the classical reduction requires $\mathrm{poly}(\log p)$ many queries to the OneEnd problem oracle $\mathscr{O}$ which can be impractical. In contrast, our heuristic quantum reduction reduces the number of classical queries to just $O(1)$, significantly tightening the reduction and strengthening the argument for the 2-special soundness property of the SQIsign identification scheme.

6.3 Quantum Reductions Between Hard Problems in Isogeny-Based Cryptography

Besides the EndRing and OneEnd problems introduced earlier, there are also other important hard problems in isogeny-based cryptography. For instance: the ℓ-Isogeny problem asks to find an ℓ-isogeny path between two given supersingular elliptic curves, and the Isogeny problems asks to find an arbitrary isogeny between two given supersingular elliptic curves.

It has been shown from [EHL+18, Wes21, PW24] that these problems are equivalent under polynomial time classical reductions, and at the core of the equivalences are the reductions of EndRing to each of them, as shown in Fig. 3a.

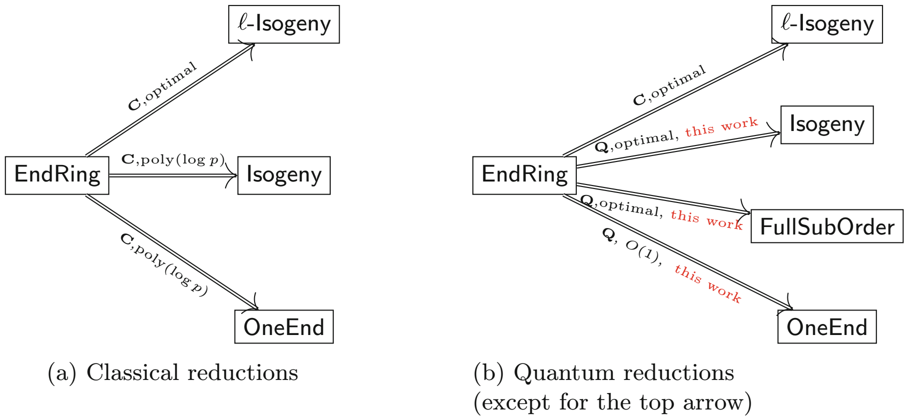
Despite all being classical polynomial time, the three reductions from EndRing to other problems are of different tightness. The reduction from EndRing to ℓ-Isogeny has *optimal tightness*, meaning that solving one instance of the EndRing problem requires solving only one instance of the ℓ-Isogeny. In contrast, the other two reductions from EndRing to Isogeny and OneEnd both require $\mathrm{poly}(\log p)$ may queries to the oracles for the Isogeny or OneEnd problems, which can be impractical as discussed in Sect. 6.2 for the case of OneEnd.

The post-quantum nature of isogeny-based cryptography suggests the interest in studying the reductions between these hard problems with quantum algorithms.

- The reduction from EndRing to Isogeny is in fact the IsERP (Problem 5), therefore, with our polynomial time heuristic quantum algorithm that solves the IsERP (Theorem 3), we obtain a quantum reduction from EndRing to Isogeny of optimal tightness.
- With the discussion from Sect. 6.2, the quantum reduction from EndRing to OneEnd obtained from Theorem 33 is tighter than the classical reduction from [PW24], reducing the query complexity from $\mathrm{poly}(\log p)$ to $O(1)$.
- Let us introduce the problem of finding a full rank suborder of $\mathrm{End}(E)$ given a supersingular elliptic curve E as the FullSubOrder problem, then our main theorem (Theorem 1) can be described as a quantum reduction from EndRing to FullSubOrder of optimal tightness.

We summarize the reductions between hard problems using quantum algorithm in Fig. 3b.

Remark 34. In Fig. 3, there are no edges going from the hard problems ℓ-Isogeny, Isogeny, FullSubOrder and OneEnd to EndRing because such reductions are the easier directions and all of them are polynomial classical reductions of optimal tightness. There are also no reductions between the hard problems ℓ-Isogeny, Isogeny, FullSubOrder and OneEnd themselves as either there are trivial reductions between them or the reductions are induced from the ones going through EndRing. Finally, we did not include FullSubOrder in Fig. 3a as the existing polynomial time classical reduction of EndRing to FullSubOrder is only the trivial one induced from that to OneEnd, which is less interesting. One might consider the classical algorithm that computes $\mathrm{End}(E)$ from a suborder in [EHL+20], but



(a) Classical reductions

(b) Quantum reductions (except for the top arrow)

Fig. 3. Polynomial time reductions of EndRing to ℓ-Isogeny, Isogeny, and OneEnd respectively. Figure 3a represents the state-of-the-art classical reductions, Fig. 3b represents the quantum reductions obtained from this work (except for the top arrow). On the edges, "$\mathbf{C}, \mathbf{Q}$" represents *classical* and *quantum* reductions respectively. Labels "optimal, $O(1)$, poly$(\log p)$" measure the query complexity of each reduction.

this algorithm assumes that the suborder is Bass, and more critically, it runs in subexponential time instead of polynomial.

Remark 35. One drawback of our quantum reductions is that they rely on more heuristics than the classical ones. But as discussed in Sect. 7, we expect that at least some heuristics can be removed at the cost of an increased runtime complexity. Importantly, this increase would still remain within polynomial bounds, and the query complexity would be unaffected.

7 Future Work

Remark 27, Remark 31 and [CP25, Appendix A] briefly summarize and discuss all the heuristics used the proof of our main results. Many of the heuristics are the "common" KLPT-type heuristics as discussed in Remark 27. Following [Wes21] one can probably remove these heuristics (assuming only GRH) at the cost of larger output sizes. For the remaining two heuristics (beyond GRH), we leave their removal to future work.

Acknowledgements. We thank anonymous reviewers for many helpful suggestions that improved the exposition of this paper. We thank Yi-Fu Lai for helpful discussions in applications of this algorithm. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT - No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VR20192203. Christophe Petit was supported by EPSRC through grant number EP/V011324/1.

References

[BFD+24] Basso, A., et al.: SQIsign2D-west: the fast, the small, and the safer. Cryptology ePrint Archive, Paper 2024/760 (2024)

[CGL09] Charles, D.X., Goren, E.Z., Lauter, K.E.: Cryptographic hash functions from expander graphs. J. Cryptol. **22**(1), 93–113 (2009). https://eprint.iacr.org/2006/021

[CII+23a] Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. LNCS, vol. 14440, pp. 99–130. Springer, Singapore (2023)

[CII+23b] Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. Cryptology ePrint Archive, Paper 2023/779 (2023)

[CLM+18] Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15

[CLP24] Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: group action from 2-dimensional isogenies. In: Tang, Q., Teague, V. (eds.) PKC 2024. LNCS, vol. 14603, pp. 190–216. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-57725-3_7

[Cor08] Cornacchia, G.: Su di un metodo per la risoluzione in numeri interi dell' equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. Giornale di Matematiche di Battaglini **46**, 33–90 (1908)

[Cox89] Cox, D.A.: Primes of the form x2 + ny2: fermat, class field theory, and complex multiplication (1989)

[CP25] Chen, M., Petit, C.: Computing the endomorphism ring of a supersingular elliptic curve from a full rank suborder (2025). to appear on ePrint

[DF24] Duparc, M., Fouotsa, T.B.: SQIPrime: a dimension 2 variant of SQISignHD with non-smooth challenge isogenies. Cryptology ePrint Archive, Paper 2024/773 (2024)

[DFKL+20] De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12491, pp. 64–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_3

[DLRW24] Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: new dimensions in cryptography. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024. LNCS, vol. 14651, pp. 3–32. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58716-0_1

[EHL+18] Eisenträger, K., Hallgren, S., Lauter, K., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 329–368. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_11

[EHL+20] Eisentraeger, K., Hallgren, S., Leonardi, C., Morrison, T., Park, J.: Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. Open Book Series (2020)

[ES24] Eisentraeger, K., Scullard, G.: Connecting kani's lemma and path-finding in the bruhat-tits tree to compute supersingular endomorphism rings. ArXiv, abs/2402.05059 (2024)

[FFK+23] Feo, L.D., et al.: SCALLOP: scaling the CSI-FISH. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023. LNCS, vol. 13940, pp. 345–375. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-31368-4_13

[KLPT14] Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ-isogeny path problem. LMS J. Comput. Math. **17**(A), 418–432 (2014)

[Koh96] Kohel, D.R.: Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California, Berkeley (1996)

[Ler22a] Leroux, A.: A new isogeny representation and applications to cryptography. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022. LNCS, vol. 13792, pp. 3–35. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22966-4_1

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

[MW23] Merdy, A.H.L., Wesolowski, B.: The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448 (2023). https://eprint.iacr.org/2023/1448

[NO24] Nakagawa, K., Onuki, H.: SQIsign2D-east: a new signature scheme using 2-dimensional isogenies. Cryptology ePrint Archive, Paper 2024/771 (2024). https://eprint.iacr.org/2024/771

[PW24] Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024. LNCS, vol. 14656, pp. 388–417. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58751-1_14

[Rob22] Robert, D.: Some applications of higher dimensional isogenies to elliptic curves (overview of results). Cryptology ePrint Archive, Paper 2022/1704 (2022). https://eprint.iacr.org/2022/1704

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

[Voi21] Voight, J.: Quaternion Algebras. Graduate Texts in Mathematics, vol. 288. Springer, Cham (2021)

[Wat69] Waterhouse, W.C.: Abelian varieties over finite fields. Annales scientifiques de l'École Normale Supérieure **2**(4), 521–560 (1969)

[Wes21] Wesolowski, B.: The supersingular isogeny path and endomorphism ring problems are equivalent. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 1100–1111 (2021)