Designs, Codes and Cryptography
https://doi.org/10.1007/s10623-024-01435-5

Finding orientations of supersingular elliptic curves and quaternion orders

Sarah Arpin[1] · James Clements[2] · Pierrick Dartois[3] · Jonathan Komada Eriksen[4] · Péter Kutas[5,6] · Benjamin Wesolowski[7]

Received: 15 September 2023 / Revised: 8 May 2024 / Accepted: 23 May 2024

Abstract
An oriented supersingular elliptic curve is a curve which is enhanced with the information of an endomorphism. Computing the full endomorphism ring of a supersingular elliptic curve is a known hard problem, so one might consider how hard it is to find one such orientation. We prove that access to an oracle which tells if an elliptic curve is $\mathfrak{O}$-orientable for a fixed imaginary quadratic order $\mathfrak{O}$ provides non-trivial information towards computing an endomorphism corresponding to the $\mathfrak{O}$-orientation. We provide explicit algorithms and in-depth complexity analysis. We also consider the question in terms of quaternion algebras. We provide algorithms which compute an embedding of a fixed imaginary quadratic order into a maximal order of the quaternion algebra ramified at p and ∞. We provide code implementations in Sagemath (in Stein et al. Sage Mathematics Software (Version 10.0), The Sage Development Team, http://www.sagemath.org, 2023) which is efficient for finding embeddings of imaginary quadratic orders of discriminants up to $O(p)$, even for cryptographically sized p.

Keywords Isogeny-based cryptography · Cryptography · Isogeny · Supersingular · Supersingular elliptic curves · Orientations · Quaternion algebras

Mathematics Subject Classification 11T71 · 11R52 · 11G05

1 Introduction

Isogeny-based cryptography is a relatively new branch of post-quantum cryptography which is based on hard problems supposedly intractable even for quantum computers. The underlying hard problems were first introduced publicly in 2006 by the hash-function proposal

Communicated by S. D. Galbraith.

This research was funded in part by the Agence Nationale de la Recherche under grant ANR MELODIA (ANR-20-CE40–0013), ANR CIAO (ANR-19-CE48–0008), and the France 2030 program under grant agreement No. ANR-22-PETQ-0008 PQ-TLS. 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 János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the UNKP-23-5 New National Excellence Program. Péter Kutas is also partly supported by EPSRC through grant number EP/V011324/1.

Extended author information available on the last page of the article of Charles-Goren-Lauter [12], and the works of Couveignes [15] and Rostovtsev-Stolbunov [56]. Since then, this field has blossomed with the introductions of new schemes such as SIDH [31] (now broken by [10, 41, 55]), CSIDH [11], and SQISign [27]. The hardness of all isogeny-based schemes is based on some variant of the path finding problem, which asks to find an isogeny between two given supersingular elliptic curves. The quaternion analogue of this hard problem has been efficiently solved [34], but the problem remains hard for supersingular elliptic curves. Path finding in the supersingular isogeny graph is equivalent to endomorphism ring computation, which was first heuristically proven in [21] and then rigorously (assuming GRH) proven in [65]. The key recovery of CSIDH was reduced to endomorphism ring computations in [66].

To study the hardness of the path finding problem it is natural to add some data to the elliptic curves and study how this data interacts with the graph structure. One way to do this is to add the information of an orientation to the elliptic curve vertices. Informally, an orientation on an elliptic curve E is an embedding of an imaginary quadratic order $\mathfrak{O}$ into the endomorphism ring of E which cannot be extended to a superorder of $\mathfrak{O}$. The resulting isogeny graph admits an abelian group action, which is used in cryptographic protocols such as CSIDH [11], Scallop [26], OSIDH [13], and SETA [18]. The group action is crucial for defining the Uber Isogeny Problem [66, Problem 4], whose hardness underlies all isogeny-based schemes. One might suspect that being given the information of an orientation could weaken the difficulty of the path finding problem, but this depends heavily on the given orientation and does not typically weaken the hardness of the path finding problem [1, 66]. A natural question to consider would be how to find an orientation on a curve, given that one exists. This is the $\mathfrak{O}$-Orienting Problem. It is also natural to consider the decisional version of this problem: given a supersingular elliptic curve E and a quadratic order $\mathfrak{O}$, can one decide whether E is orientable by $\mathfrak{O}$?

The search version of the $\mathfrak{O}$-Orienting Problem underlies the security of:

- OSIDH [13, 46], and
- FESTA [3] and its variants [45], [44] due to the following observation.

In all of these schemes there is a secret isogeny of known degree d from a curve which has small endomorphisms (usually the curve $y^2 = x^3 + x$). In all these cases the codomain of this isogeny is primitively oriented by $\mathbb{Z}[d\omega]$ where ω is the mentioned special small degree endomorphism. Finding this orientation will reveal the secret isogeny. Furthermore, the problem of finding a fixed degree isogeny to a special curve (if it exists) is believed to be a hard problem. Since the best algorithms for attacking the aforementioned schemes are exponential time, the best algorithms for solving the $\mathfrak{O}$-Orienting Problem are also exponential time. The problem of deciding whether a curve is oriented is semantically a weaker problem and has not been previously studied. It is natural to question how the decision variant relates to the search variant of this problem, e.g. can it be solved in subexponential time? The search variant of a problem is always at least as hard as the decision variant. If there exists a reduction from search to decision, it shows that the decision variant of the problem reveals some nontrivial information. In cryptography, such "search-to-decision" reductions are helpful in understanding hardness assumptions.

Interestingly, these problems are not even efficiently solved on the quaternion side, except in very special cases. In [66] the case where $\operatorname{disc}(\mathfrak{O})$ is small (i.e., $\operatorname{disc}(\mathfrak{O}) < \sqrt{p}$) is studied. This can be accomplished in polynomial time as in general this is just the smallest non-scalar endomorphism (this argument should actually work with a small modification for cases where $\operatorname{disc}(\mathfrak{O}) < p^{2/3}$). In [4, Appendix A] this is conjecturally improved to $p^{0.8}$ using Coppersmith techniques.

In this work, we give reductions between the search and decisional variants of these problems, and provide algorithms for the quaternion variant of these problems.

1.1 Our contributions

1.1.1 Reduction from Search to Decisional $\mathfrak{O}$-Orienting Problem

When the discriminant of $\mathfrak{O}$ is smaller than the characteristic p of the base field, we prove a subexponential reduction from the computational to the decisional version of the $\mathfrak{O}$-Orienting Problem. In particular, we provide an explicit algorithm (Algorithm 4.3) to find an $\mathfrak{O}$-orientation of an orientable elliptic curve in subexponential time and space when given access to an oracle deciding whether any elliptic curve is $\mathfrak{O}$-orientable. Access to this oracle is unlikely to be possible, but theoretical access to such a perfect oracle shows that finding an endomorphism is not significantly more difficult than simply deciding if one exists. In Sect. 4.2 we provide an in-depth analysis and proof of the complexity of the algorithm (Theorem 4.13). This proves that such an oracle gives non-trivial information since finding an orientation automatically yields a non-scalar endomorphism and the best known algorithms to find a non-scalar endomorphism on a supersingular elliptic curve are exponential [19, Sect. 4], [22, 28]. Note that in this paper we always invoke a perfect oracle which always returns the correct answer. The case of imperfect oracles (i.e., oracles which return the correct answer with probability $1 - \epsilon$ for some fixed $\epsilon > 0$) is left for future work.

Before treating the general case, we prove a polynomial reduction when $\mathfrak{O}$ is the maximal order of $\mathbb{Q}(\sqrt{-d})$ and d is the product of small distinct primes in Sect. 3. This allows us to illustrate the spirit of the more general algorithm in a less complicated setting. We provide an explicit algorithm for this case (Algorithm 3.1) and prove in Theorem 3.10 that this algorithm runs in polynomial time.

1.1.2 Quaternion Order Embedding Problem

In Sect. 5, we consider the Quaternion Order Embedding Problem (Problem 2.6) which is the quaternion analogue of the $\mathfrak{O}$-Orienting Problem. That is, given a maximal quaternion order $\mathcal{O} \subset B_{p,\infty}$ and a quadratic order $\mathfrak{O}$ which embeds into $\mathcal{O}$, find an embedding $\iota : \mathfrak{O} \hookrightarrow \mathcal{O}$ that cannot be extended to a superorder of $\mathfrak{O}$. In Sect. 5.1 we present a general algorithm to solve the problem of finding embeddings using a factorization oracle. We provide a complexity analysis based on several heuristics in Sect. 5.2. In Sect. 5.5 we show that finding embeddings which cannot be extended (i.e., orientations), only adds a small factor to the running time. We prove efficiency for the curve with j-invariant 1728, and describe a practical method for removing the dependence on the factorization oracle. Our algorithm improves the state of the art, since it is efficient up to even $\operatorname{disc}(\mathfrak{O}) = O(p)$. We provide an implementation in Sagemath [61] which, for small discriminant orders, is fast for cryptographically sized p.

2 Preliminaries

We provide a concise summary of the necessary background and the state of the art algorithms which we use in this paper.

2.1 Supersingular elliptic curves and quaternion algebras

Let p be a prime. An elliptic curve over $\overline{\mathbb{F}_p}$ is called *supersingular* if any one of the following equivalent conditions holds:

(1) $\operatorname{End}(E)$ is isomorphic to a maximal order in a quaternion algebra
(2) $E[p^r] = 0_E$ for all $r \geq 1$
(3) $j(E) \in \mathbb{F}_{p^2}$ and the multiplication-by-p map $[p]$ is purely inseparable
(4) The dual to the p^r-power Frobenius is purely inseparable for all $r \geq 1$.

See [60] for additional properties and proofs of equivalence.

We use the endomorphism ring heavily in what follows, so we describe here the necessary definitions and properties of quaternion objects. For more generality and more detail, we encourage the reader to see [62].

A (definite) quaternion algebra $\mathcal{A}$ is a noncommutative algebra which has rank 4 over $\mathbb{Q}$, and can be specified by generators i, j such that:

$$\mathcal{A} = \mathbb{Q} + \mathbb{Q}i + \mathbb{Q}j + \mathbb{Q}k : \quad i^2, j^2 \in \mathbb{Q}, \quad i^2, j^2 < 0, \quad k := ij = -ji.$$

An order $\mathcal{O}$ in $\mathcal{A}$ is a $\mathbb{Z}$-submodule of $\mathcal{A}$ of rank 4 which is also a subring. An order is said to be maximal if it is not properly contained in any other order. For any full rank lattice (sub-$\mathbb{Z}$-module of rank 4) I in $\mathcal{A}$, we define its left order

$$O_L(I) := \{\alpha \in \mathcal{A} : \alpha I \subseteq I\}.$$

The right order $O_R(I)$ is defined analogously. A full rank lattice I of $\mathcal{A}$ is said to be invertible if there exists a lattice I' such that $II' = O_L(I) = O_R(I')$ and $I'I = O_R(I) = O_L(I')$. A full rank lattice in $\mathcal{A}$ is said to be a left (resp. right) $\mathcal{O}$-ideal if $\mathcal{O} \subseteq O_L(I)$ (resp. $\mathcal{O} \subseteq O_R(I)$). For every order $\mathcal{O}$ of $\mathcal{A}$ we define a left class set of equivalence classes of invertible left ideals: invertible left $\mathcal{O}$ ideals I, J are equivalent in the left class set of $\mathcal{O}$ if and only if there exists $\gamma \in \mathcal{A}^\times$ such that $I = \gamma J$. The left class set of invertible ideals is finite. The right class set of invertible ideals is analogously defined and is also finite.

For a fixed prime p, we define the (unique up to isomorphism) quaternion algebra $B_{p,\infty}$ to be the definite quaternion algebra ramified precisely at p and ∞. The endomorphism rings of supersingular elliptic curves over $\overline{\mathbb{F}_p}$ are isomorphic to maximal orders in $B_{p,\infty}$:

Theorem 2.1 (Deuring [20]) *Fix a maximal order M of the quaternion algebra $B_{p,\infty}$ ramified precisely at p and ∞. There is a bijection between isomorphism classes of supersingular elliptic curves over $\overline{\mathbb{F}_p}$ and the left class set of the order M.*

Given a supersingular elliptic curve $E/\overline{\mathbb{F}_p}$, one might ask to compute $\operatorname{End}(E)$ in different forms: to compute endomorphisms of E which generate $\operatorname{End}(E)$, or to compute the isomorphism class of $\operatorname{End}(E)$ abstractly in the quaternion algebra $B_{p,\infty}$. This problem is computationally difficult in all formulations. The supersingular endomorphism problem was recently shown to be equivalent to the problem of finding a single non-scalar endomorphism [47]. A priori, the information of one endomorphism ω of E reveals an imaginary quadratic order $\mathbb{Z}[\omega]$ embedded within $\operatorname{End}(E)$. In Sect. 2.2, we provide more background information on such embeddings.

2.2 Orientations

Definition 2.2 (Orientation) Let $\mathfrak{O}$ be an imaginary quadratic order. An $\mathfrak{O}$-orientation of a supersingular elliptic curve $E/\overline{\mathbb{F}_p}$ is an embedding $\iota : \mathfrak{O} \hookrightarrow \operatorname{End}(E)$ which cannot be extended to a larger quadratic order containing $\mathfrak{O}$. The pair (E, ι) is called an $\mathfrak{O}$-oriented supersingular elliptic curve.

Definition 2.2 corresponds to the definition of *primitive* $\mathfrak{O}$-orientation found elsewhere in the literature [1, 13, 46]. We omit the word “primitive" in our definition, as almost all of our $\mathfrak{O}$-orientations are primitive. When we want to discuss an embedding $\mathfrak{O} \hookrightarrow \mathrm{End}(E)$ which can be extended to a superorder of $\mathfrak{O}$, we highlight this by using the term “imprimitive".

The notion of an orientation as in Definition 2.2 was recently introduced to isogeny-based cryptography by Colò and Kohel [13] and was subsequently studied [1, 2, 26, 46, 66]. The quaternion counterpart of this notion has a longer history, dating back to Chevalley, Hasse, and Noether and often referred to as the theory of *optimal embeddings*.

Supersingular elliptic curves which admit an $\mathfrak{O}$-orientation are called $\mathfrak{O}$-orientable. There is an action of the class group $\mathrm{Cl}(\mathfrak{O})$ on the set of $\mathfrak{O}$-oriented supersingular elliptic curves induced by the following action of an invertible $\mathfrak{O}$-ideal $\mathfrak{a}$:

$$\mathfrak{a} * (E, \iota) := (E_{\mathfrak{a}}, (\varphi_{\mathfrak{a}})_* \iota),$$

where $E_{\mathfrak{a}}$ is the codomain of the degree-$N(\mathfrak{a})$ isogeny $\varphi_{\mathfrak{a}} : E \longrightarrow E_{\mathfrak{a}}$ with kernel $\bigcap_{\alpha \in \mathfrak{a}} \ker \alpha$. The orientation $(\varphi_{\mathfrak{a}})_*\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E_{\mathfrak{a}})$ is given via $(\varphi_{\mathfrak{a}})_*\iota(-) := \frac{1}{N(\mathfrak{a})}\varphi_{\mathfrak{a}} \circ \iota(-) \circ \widehat{\varphi_{\mathfrak{a}}}$.

For an imaginary quadratic field K, K embeds into the quaternion algebra $B_{p,\infty}$ if and only if p does not split in K. However, for a particular imaginary quadratic order $\mathfrak{O}$ and a particular supersingular elliptic curve E, it is generally difficult to decide if E is $\mathfrak{O}$-orientable. Naturally we are inclined to study the following problems and the relationship between them:

Problem 2.3 (Decisional $\mathfrak{O}$-Orienting Problem) *Given an elliptic curve E and an imaginary quadratic order $\mathfrak{O}$, determine if E is orientable by $\mathfrak{O}$.*

Problem 2.4 ($\mathfrak{O}$-Orienting Problem) *Given an elliptic curve E which is orientable by an imaginary quadratic order $\mathfrak{O}$, find the orientation.*

Remark 2.5 In a cryptographic context decisional problems are often defined in a slightly different fashion. Namely an adversary is presented with two supersingular elliptic curves E_1 and E_2 and an imaginary quadratic order $\mathfrak{O}$ one of which is oriented by $\mathfrak{O}$. Then one has to decide whether E_1 or E_2 is oriented by $\mathfrak{O}$. The reason is that whenever the discriminant is sufficiently small (i.e., much smaller than p^2), then a random supersingular elliptic curve is likely not oriented by $\mathfrak{O}$. Hence an adversary that always says "not oriented" succeeds in this game with overwhelming probability.

The search-to-decision reduction shows that a decision oracle provides non-trivial information towards finding the endomorphism ring of the curve, and this will be the focus of Sects. 3 and 4.

We explore the following quaternion variant of Problem 2.4 in Sect. 5.

Problem 2.6 (Quaternion Order Embedding Problem) *Given a maximal quaternion order $\mathcal{O}$ and an imaginary quadratic order $\mathfrak{O}$ which embeds into $\mathcal{O}$, find the embedding.*

One may also consider the group action variant of the Uber-isogeny problem, originally introduced in [18], although we do not pursue this perspective in this work:

Problem 2.7 ($\mathfrak{O}$-Uber Isogeny Problem) *Given a supersingular elliptic curve E with an $\mathfrak{O}$-orientation $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ and an $\mathfrak{O}$-orientable supersingular elliptic curve F, find an ideal $\mathfrak{a} \in \mathrm{Cl}(\mathfrak{O})$ such that $\mathfrak{a} * E = F$.*

2.3 Complexity

Throughout this paper, we shall give time and space complexity results. By default, we say an algorithm has (time) complexity $O(f(n))$ or terminates in time $O(f(n))$ when it runs in $O(f(n))$ bit operations. In particular, an algorithm is *polynomial* in the parameter n if it runs in number of bit operations which is bounded by a polynomial function of n. Very often, we shall also mention time complexity in terms of arithmetic operations over $\mathbb{Z}$ or finite fields $\mathbb{F}_p$, $\mathbb{F}_{p^2}$ and their extensions. By *arithmetic operations* or simply *operations* we mean additions, substractions, multiplications, inversions and random sampling (up to a bound over $\mathbb{Z}$). Unless explicitely stated otherwise, space complexity will always be counted in bits.

2.4 Computing modular polynomials and *j*-invariants

Given a prime number $\ell \ll p$ and the j-invariant $j(E) \in \mathbb{F}_{p^2}$ of a supersingular elliptic curve, we explain how to find all ℓ-isogenous j-invariants $j(E') \in \mathbb{F}_{p^2}$ using modular polynomials $\Phi_\ell(X, Y)$. By [54, Theorem 6.3], $\Phi_\ell(j(E), Y) \in \mathbb{F}_{p^2}[Y]$ can be computed with $\tilde{O}(\ell^2 \log(p))$ operations over $\mathbb{F}_{p^2}$, where the $\tilde{O}$ means that polynomial factors in $\log(\ell)$ are omitted.[1] In [39, Sect. 5] the authors provide an algorithm with similar complexity. We then find all the roots over $\mathbb{F}_{p^2}$ of the degree-$(\ell+1)$ polynomial $\Phi_\ell(j(E), Y)$ in $\tilde{O}(\ell^2 \log(p))$ operations over $\mathbb{F}_{p^2}$ [63, Theorem 14.14] to find the j-invariants $j(E') \in \mathbb{F}_{p^2}$ that are ℓ-isogenous to $j(E)$. On the whole, the computation costs $\tilde{O}(\ell^2 \log(p))$ operations over $\mathbb{F}_{p^2}$.

2.5 Computing an ℓ-isogeny between two *j*-invariants

Given two supersingular j-invariants $j(E) \in \mathbb{F}_{p^2}$ and $j(E') \in \mathbb{F}_{p^2}$ we explain how to find an ℓ-isogeny $\phi : E \longrightarrow E'$ in $\tilde{O}(\ell^2 \log(p))$ operations over $\mathbb{F}_{p^2}$ using a method due to Elkies.

By [7, Theorem 2], given Weierstrass equations of E and E', we can find (if it exists) a normalized ℓ-isogeny $\phi : E \longrightarrow E'$ with only $\tilde{O}(\ell)$ arithmetic operations over $\mathbb{F}_{p^2}$. By *normalized*, we mean that ϕ pulls back the invariant differential $\omega' := dx'/2y'$ of E' to the invariant differential $\omega := dx/2y$ of E ($\phi^*\omega' = \omega$).

The existence of such a normalized isogeny ϕ only depends on the choice of Weierstrass equations for E and E' which determine the constant $\lambda := \phi^*\omega'/\omega$. Knowing only $j(E)$ and $j(E')$, we have multiple choices of Weierstrass equations and we have to pick one so that $\lambda = 1$. We fix an equation for $E : y^2 = x^3 + Ax + B$, then find an equation for E' so that $\lambda = 1$. Following the method given by [58, Sect. 7] (referring to ideas introduced in [23, Sect. 3]), we take $E' : y^2 = x^3 + A'x + B'$, with

$$A' := -\frac{j'(E')^2\ell^4}{48j(E')(j(E') - 1728)} \quad B' := -\frac{j'(E')^3\ell^6}{864j(E')^2(j(E') - 1728)}, \tag{1}$$

and

$$j'(E') := -\frac{j'(E)}{\ell}\frac{\partial\Phi_\ell}{\partial X}(j(E), j(E'))\left(\frac{\partial\Phi_\ell}{\partial Y}(j(E), j(E'))\right)^{-1}, \tag{2}$$

[1] The algorithm is provided over $\mathbb{F}_p$ but the techniques of [54] easily extend to $\mathbb{F}_{p^2}$.

where

$$j'(E) := \begin{cases} \dfrac{18Bj(E)}{A} & \text{if } A \neq 0 \\ 0 & \text{if } A = j(E) = 0 \end{cases}. \tag{3}$$

The derivatives $\partial\Phi_\ell/\partial X$ and $\partial\Phi_\ell/\partial Y$ can be precomputed with $\tilde{O}(\ell^2\log(p))$ operations over $\mathbb{F}_{p^2}$ using the techniques in Sect. 2.4 (see [51, Remark 5.3.10]). Hence, in total, computing an isogeny $\phi : E \longrightarrow E'$ costs $\tilde{O}(\ell^2\log(p))$ operations over $\mathbb{F}_{p^2}$.

The above formulas fail when $j(E)$ or $j(E') \in \{0, 1728\}$, and when $\partial\Phi_\ell/\partial Y(j(E), j(E'))$ vanishes. However, when $\partial\Phi_\ell/\partial Y(j(E), j(E')) = 0$ and $\partial\Phi_\ell/\partial X(j(E), j(E')) \neq 0$, this implies there are more than one ℓ-isogeny from the isomorphism class determined by $j(E)$ to the isomorphism class determined by $j(E')$, but only one ℓ-isogeny in the reverse direction. This is only possible if $j(E) \in \{0, 1728\}$, in which case the formulas fail anyways.

Hence, the only cases when this method does not apply are $j(E)$ or $j(E') \in \{0, 1728\}$, which are very unlikely (probability $O(1/p)$) and $\partial\Phi_\ell/\partial X(j(E), j(E')) = \partial\Phi_\ell/\partial Y(j(E), j(E')) = 0$, i.e. when $(j(E), j(E'))$ is a singular point of the affine curve given by the modular equation $\Phi_\ell(X, Y) = 0$ over $\mathbb{F}_{p^2}$. Following [58, Sect. 7], we prove in Appendix A that this is very unlikely when $\log(\ell) \ll \log(p)$ (which will be the case in our paper).

We can still handle singular cases at a higher cost of $\tilde{O}(\ell^{7/2})$ operations over $\mathbb{F}_{p^2}$ with a naive algorithm. We enumerate all the cyclic subgroups of order ℓ of $E[\ell]$ (there are $\ell + 1$ of them) and use [6] to compute each ℓ-isogeny with $\tilde{O}(\sqrt{\ell})$ operations over the field extension $K/\mathbb{F}_{p^2}$ where $E[\ell]$ is defined. As will be proved in Lemma 2.12, K has degree $O(\ell)$ over $\mathbb{F}_{p^2}$ so one arithmetic operation over K is equivalent to at most $O(\ell^2)$ operations over $\mathbb{F}_{p^2}$. Since singular cases are very unlikely when $\log(\ell) \ll \log(p)$, we may assume throughout this paper that computing ℓ-isogenies between j-invariants costs $\tilde{O}(\ell^2\log(p))$ operations over $\mathbb{F}_{p^2}$ on average by Lemma A.2.

2.6 Efficiently representing an isogeny of any degree with Kani's lemma

Let $\varphi : E \longrightarrow E'$ be an isogeny of degree d between supersingular elliptic curves $E, E'/\mathbb{F}_{p^2}$. In general, we can represent φ with data of size $O(d)$. We either have direct formulas to evaluate φ (given by rational fractions) or equivalently, generators of the kernel (defined over an $\mathbb{F}_{p^2}$-extension of degree $O(d)$) from which we can derive these formulas by [64]. In this case, evaluating φ on a point takes linear time in d. We can do much better when d is smooth by representing φ as a product of small degree isogenies. This is an efficient representation, in the sense of the following definition.

Definition 2.8 Let $\mathscr{A}$ be an algorithm (to compute isogenies). An *efficient representation* of an isogeny $\varphi : E \longrightarrow E'$ defined over a finite field $\mathbb{F}_q$ (with respect to $\mathscr{A}$) is given by some data $D \in \{0, 1\}^*$ such that:

(i) D has polynomial size in $\log(\deg(\varphi))$ and $\log(q)$ (in bits).
(ii) On input D and $P \in E(\mathbb{F}_{q^k})$, $\mathscr{A}$ returns $\varphi(P)$ in polynomial time in $k\log(q)$ and $\log(\deg(\varphi))$.

We can also efficiently represent φ when d is not smooth, using an idea first introduced in the attacks against SIDH [10, 41, 52] and then reused for several other applications [16, 53, 54]: provided we can evaluate φ on some torsion points, we can "embed" φ in a smooth degree higher dimensional isogeny F. Knowing F, we can evaluate φ everywhere in polynomial time. This provides an efficient representation of φ.

In this section, we explain how to obtain such an efficient representation F of φ when we are given access to the image of φ on some torsion points (Algorithm 2.1). The evaluation of these points may be costly but other operations to compute F take polynomial time (Proposition 2.12).

Definition 2.9 (d-isogeny in higher dimension) Let $\alpha : (A, \lambda_A) \longrightarrow (B, \lambda_B)$ be an isogeny between principally polarized abelian varieties (PPAV). We denote by $\widetilde{\alpha}$ the isogeny

$$B \xrightarrow{\lambda_B} \widehat{B} \xrightarrow{\widehat{\alpha}} \widehat{A} \xrightarrow{\lambda_A^{-1}} A,$$

where $\widehat{\alpha}$ is the dual isogeny of α.

We say that α is a *d-isogeny* if $\widetilde{\alpha} \circ \alpha = [d]_A$, or equivalently if $\alpha \circ \widetilde{\alpha} = [d]_B$.

We use the following result due to Kani [33, Theorem 2.3]. A concise expression of this result may be found in [52, Lemma 3.6].

Lemma 2.10 (Kani) *Consider a commutative diagram of isogenies between PPAV:*

$$\begin{tikzcd} A' \arrow[r, "\varphi'"] & B' \\ A \arrow[u, "\psi"] \arrow[r, "\varphi"] & B \arrow[u, "\psi'"'] \end{tikzcd}$$

where φ and φ' are a-isogenies and ψ and ψ' are b-isogenies.

Then, the isogeny chain $F : A \times B' \longrightarrow B \times A'$ given in matrix notation by

$$F := \begin{pmatrix} \varphi & \widetilde{\psi'} \\ -\psi & \widetilde{\varphi'} \end{pmatrix}$$

is a d-isogeny with $d := a + b$, for the product polarizations.

If a and b are coprime, the kernel of F is

$$\ker(F) = \{(\widetilde{\varphi}(x), \psi'(x)) \mid x \in B[d]\}.$$

Let $N > d$ be a powersmooth integer coprime with d. We can always write $N = d + a_1^2 + a_2^2 + a_3^2 + a_4^2$ for some $a_1, a_2, a_3, a_4 \in \mathbb{Z}$, by Lagrange's four square theorem. Let $\alpha \in \mathrm{End}(E^4)$ be the isogeny written in matrix form as follows:

$$\alpha := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_4 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix}, \tag{4}$$

and α' be its analogue in $\mathrm{End}(E')$. Let $\Phi := \mathrm{Diag}(\varphi, \varphi, \varphi, \varphi) : E^4 \longrightarrow E'^4$. Then, Φ is a d-isogeny, α and α' are $(N - d)$-isogenies and we have a commutative diagram:

$$\begin{tikzcd} E^4 \arrow[r, "\Phi"] & E'^4 \\ E^4 \arrow[u, "\alpha"] \arrow[r, "\Phi"] & E'^4 \arrow[u, "\alpha'"] \end{tikzcd}$$

that yields an 8-dimensional N-isogeny:

$$F := \begin{pmatrix} \alpha & \widetilde{\Phi} \\ -\Phi & \widetilde{\alpha} \end{pmatrix} \in \mathrm{End}(E^4 \times E'^4), \tag{5}$$

with kernel:

$$\ker(F) := \{(\widetilde{\alpha}(P), \Phi(P)) \mid P \in E^4[N]\}, \tag{6}$$

since N and d are coprime. By the above formula, we can compute $\ker(F)$ if we can evaluate φ on generators of $E[N]$. We can then compute F as a product of small degree isogenies (as in dimension 1) and evaluate φ efficiently everywhere as a component of F. Indeed, if we want to evaluate $\varphi(P)$ for some point $P \in E$, then we can compute:

$$F(P, 0, \cdots, 0) = ([a_1]P, [a_2]P, [a_3]P, [a_4]P, -\varphi(P), -\varphi(P), -\varphi(P), -\varphi(P)).$$

Lemma 2.11 *Let $F : E^4 \times E'^4 \longrightarrow E^4 \times E'^4$ be the N-isogeny defined by (5) and suppose $N = \prod_{i=1}^s q_i^{e_i}$, where $q_1, \cdots, q_s$ are distinct primes. Then, we can decompose F as:*

$$\mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \xrightarrow{F_2} \cdots \mathcal{A}_{s-1} \xrightarrow{F_s} \mathcal{A}_s,$$

with $\mathcal{A}_0 = \mathcal{A}_s := E^4 \times E'^4$ and where F_i is a $q_i^{e_i}$-isogeny for all $i \in \{1, \cdots, s\}$.

Let $K := \ker(F)$. Moreover,

$$\ker(F_1) = K[q_1^{e_1}] = \{P \in K : [q_1^{e_1}]P = 0\} \quad \textit{and}$$
$$\ker(F_i) = F_{i-1} \circ \cdots \circ F_1(K[q_i^{e_i}]), \textit{ for } 2 \le i \le s.$$

Proof The decomposition $F = F_s \circ \cdots \circ F_1$ was proven in [16, Proposition 5.4.1].

Now, let $K_1 := K[q_1^{e_1}]$ and $K_i := F_{i-1} \circ \cdots \circ F_1(K[q_i^{e_i}])$ for all $2 \le i \le s$. Then, $F_s \circ \cdots \circ F_i(K_i) = F(K[q_i^{e_i}]) = \{0\}$, so that

$$F_i(K_i) \subseteq \ker(F_s \circ \cdots \circ F_{i+1}) \subseteq \mathcal{A}_i \left[\prod_{j \ge i+1} q_j^{e_j} \right].$$

But $F_i(K_i) \subset \mathcal{A}_i[q_i^{e_i}]$ and the primes $q_j \neq q_i$ for $j > i$, so we must have $F_i(K_i) = \{0\}$ and $K_i \subseteq \ker(F_i)$. Now, $\#K_i = \#K[q_i^{e_i}] = q_i^{8e_i} = \deg(F_i)$ since $F_{i-1} \circ \cdots \circ F_1$ has degree coprime with q_i. It follows that $K_i = \ker(F_i)$. □

Each $q_i^{e_i}$-isogeny F_i in Lemma 2.11 can be computed with $O(q_i^{8e_i})$ operations over the field of definition of their kernel with the theta model [40]. We summarize this computation in Algorithm 2.1.

Lemma 2.12 *Let $E/\mathbb{F}_{p^2}$ be a supersingular elliptic curve and $n \in \mathbb{Z}_{>0}$. Then $E[n]$ is defined over an extension of degree at most $6\phi(n)$ of $\mathbb{F}_{p^2}$, where ϕ is Euler's totient function.*

Proof We first compute the characteristic polynomial of iterates of the Frobenius. Let $\chi_{p^2} := (X - \alpha)(X - \beta)$ be the characteristic polynomial of the p^2 Frobenius π_{p^2}. Then $\chi_{p^{2\delta}} := (X - \alpha^\delta)(X - \beta^\delta)$ is the characteristic polynomial of the $p^{2\delta}$ Frobenius $\pi_{p^{2\delta}}$, for any $\delta \in \mathbb{Z}_{>0}$.

Since E is supersingular $p \mid \mathrm{Tr}(\pi_{p^2}) = \alpha + \beta$, and $|\mathrm{Tr}(\pi_{p^2})| \le 2p$ so $\mathrm{Tr}(\pi_{p^2}) \in \{0, \pm p, \pm 2p\}$ [58, Proposition 3.6]. We consider these possibilities in three cases:

Algorithm 2.1: EfficientRep returning an efficient representation of an isogeny.

```
Data: An integer $D > 0$ (smoothness bound), an oracle to evaluate an isogeny $\varphi : E \longrightarrow E'$ between
      supersingular elliptic curves and $d := \deg(\varphi)$.
Result: An 8-dimensional isogeny $F$ of $D$-powersmooth degree representing $\varphi$.
1  Select prime powers $q_1^{e1}, \cdots, q_s^{es} \leq D$ coprime with $d$ such that $N := \prod_{i=1}^{s} q_i^{e_i} > d$;
2  Find $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ such that $a_1^2 + a_2^2 + a_3^2 + a_4^2 = N - d$ using Pollack and Treviño's algorithm
   [49, §4];
3  Let $\alpha \in \mathrm{End}(E)$ and $\alpha' \in \mathrm{End}(E')$ as in equation 4 and $\Phi := \mathrm{Diag}(\varphi, \varphi, \varphi, \varphi)$;
4  for i=1 to s do
5      Generate a basis $(P_{i,1}, P_{i,2})$ of $E[q_i^{e_i}]$;
6      Compute $\varphi(P_{i,1})$ and $\varphi(P_{i,2})$;
7      For all $j \in \{1, 2\}$ and $k \in \{1, 2, 3, 4\}$, let $\underline{P}_{i,j,k} \in E^4[q_i^{e_i}]$ be the tuple with $P_{i,j}$ in position $k$ and
       0 elsewhere;
8      $\mathcal{B}_i \longleftarrow \{(\widetilde{\alpha}(\underline{P}_{i,j,k}), \Phi(\underline{P}_{i,j,k})) \mid 1 \leq j \leq 2,\ 1 \leq k \leq 4\}$;
9  end
10 $\mathcal{C}_i \longleftarrow \mathcal{B}_i$ for $1 \leq i \leq s$;
11 for i=1 to s-1 do
12     Compute $F_i$ of kernel $\langle \mathcal{C}_i \rangle$;
13     for j=i+1 to s do
14         $\mathcal{C}_j \longleftarrow F_i(\mathcal{C}_j)$;
15     end
16 end
17 Compute $F_s$ of kernel $\langle \mathcal{C}_s \rangle$;
18 Return $F := F_s \circ \cdots \circ F_1$;
```

If $\mathrm{Tr}(\pi_{p^2}) = 0$, then $\chi_{p^2} = X^2 + p^2 = (X - ip)(X + ip)$ so

$$\mathrm{Tr}(\pi_{p^{2\delta}}) = (ip)^\delta + (-ip)^\delta = \begin{cases} 2p^\delta & \text{if } \delta \equiv 0 \mod 4 \\ -2p^\delta & \text{if } \delta \equiv 2 \mod 4 \\ 0 & \text{otherwise} \end{cases}.$$

If $\mathrm{Tr}(\pi_{p^2}) = \pm p$, then $\chi_{p^2} = X^2 \mp pX + p^2 = (X \mp pe^{i\pi/3})(X \mp pe^{-i\pi/3})$ so

$$\mathrm{Tr}(\pi_{p^{2\delta}}) = (\pm 1)^\delta p^\delta (e^{i\delta\pi/3} + e^{-i\delta\pi/3}) = \begin{cases} 2p^\delta & \text{if } \delta \equiv 0 \mod 6 \\ \mp 2p^\delta & \text{if } \delta \equiv 3 \mod 6 \\ \pm p^\delta & \text{if } \delta \equiv \pm 1 \mod 6 \\ -p^\delta & \text{if } \delta \equiv \pm 2 \mod 6 \end{cases}.$$

Finally, if $\mathrm{Tr}(\pi_{p^2}) = \pm 2p$, then $\chi_{p^2} = X^2 \mp 2pX + p^2 = (X \mp p)^2$ so

$$\mathrm{Tr}(\pi_{p^{2\delta}}) = 2(\pm p)^\delta.$$

In all cases, if $\delta \equiv 0 \mod 12$, we have $\mathrm{Tr}(\pi_{p^{2\delta}}) = 2p^\delta$, so $\chi_{p^{2\delta}} = (X - p^\delta)^2$. It follows that $\pi_{p^{2\delta}} = [p^\delta]$.

Now, if we assume $\phi(n) \mid \delta$, then $p^\delta \equiv 1 \mod n$, so for all $P \in E[n]$, $\pi_{p^\delta}(P) = [p^\delta]P = P$, so that $P \in E(\mathbb{F}_{p^{2\delta}})$.

Hence, $E[n]$ is defined over $\mathbb{F}_{p^{2\delta}}$ provided that 12 and $\phi(n)$ divide δ. If n has an odd prime factor or $n = 2^k$ with $k \geq 2$, then $\phi(n)$ is even so $12 \mid 6\phi(n)$ so $\delta = 6\phi(n)$ satisfy the desired conditions. If $n = 2$, then $E[n]$ is formed by 0 and the points $(x, 0)$ where x is the root of a cubic polynomial equation over $\mathbb{F}_{p^2}$ (Weierstrass equation of E). Hence, $E[2]$ is defined over an extension of degree at most 3 of $\mathbb{F}_{p^2}$. □

Proposition 2.13 *Algorithm 2.1 terminates and is correct. It requires:*

- $O(\log(d))$ *evaluations of the input* d*-isogeny* φ *on points defined over an extension of degree* $O(D)$ *of* $\mathbb{F}_p$*;*
- $O(D^3 \log(p) \log(d) + D^{10} \log^2(d))$ *arithmetic operations over* $\mathbb{F}_p$*;*
- *and* $O(\log^2(d) + \log(d) D \log\log(D))$ *arithmetic operations over integers of size at most* $O(D \log(p))$ *bits.*

Recall that by arithmetic operations, we mean addition, substraction, multiplication and inversion. Over $\mathbb{F}_p$*, we count sampling as an arithmetic operation.*

Proof Correctness has been justified by Eq. 6 and Lemma 2.11. Termination is clear.

Now we compute the complexity. If N is only slightly bigger than d, then $s = O(\log(d))$ and finding a suitable N on line 1 of the algorithm takes $O(\log(d) D \log\log(D))$ arithmetic operations over $\mathbb{Z}$ (assuming we use the sieve of Eratosthenes).

Finding the a_i on line 2 costs $O(\log^2(N)/\log\log(N)) = O(\log^2(d))$ operations over $\mathbb{Z}$ with Pollack and Treviño's second algorithm [49, §4].

For $i \in \{1, \cdots, s\}$, a basis of $E[q_i^{e_i}]$ is defined over a field extension of degree $\delta = O(q_i^{e_i}) = O(D)$ of $\mathbb{F}_{p^2}$ by Lemma 2.12. To generate such a basis, we first sample a random point $P' \in E(\mathbb{F}_{p^{2\delta}})$ and compute $P_{i,1} := [M/q_i^{e_i}]P'$, where $M := \#E(\mathbb{F}_{p^{2\delta}})$ until P has order $q_i^{e_i}$. By the same method, we sample $P_{i,2} \in E[q_i^{e_i}]$ until $(P_{i,1}, P_{i,2})$ is a basis of $E[q_i^{e_i}]$.

To sample $P' \in E(\mathbb{F}_{p^{2\delta}})$, we first sample $x \in \mathbb{F}_{p^{2\delta}}$ repeatedly ($O(1)$ times at most) until we find $y \in \mathbb{F}_{p^{2\delta}}$ such $P' = (x, y) \in E$. Each sampling over $\mathbb{F}_{p^{2\delta}}$ costs $2\delta = O(D)$ samplings over $\mathbb{F}_p$. Computing y requires a square root computation in $\mathbb{F}_{p^{2\delta}}$ which costs $O(\log(p^{2\delta})) = O(D \log(p))$ multiplications over $\mathbb{F}_{p^{2\delta}}$ by Cipolla-Lehmer's algorithm [37]. Without loss of generality (see the proof of Lemma 2.12), we may assume that $\pi_{p^{2\delta}} = [p^\delta]$, so that:

$$M = \#E(\mathbb{F}_{p^{2\delta}}) = p^{2\delta} + 1 - \mathrm{Tr}(\pi_{p^{2\delta}}) = (p^\delta - 1)^2$$

and computing M costs $O(D)$ operations over $\mathbb{Z}$. The scalar multiplication by $M/q_i^{e_i}$ costs $O(\log(M)) = O(D \log(p))$ arithmetic operations over $\mathbb{F}_{p^{2\delta}}$. Testing that $(P_{i,1}, P_{i,2})$ is a basis costs $O(D)$ elliptic curve additions so $O(D)$ arithmetic operations over $\mathbb{F}_{p^{2\delta}}$ (we compute the $[k]P_{i,1}$ and $[l]P_{i,2}$ for $1 \leq k, l \leq q_i^{e_i} - 1$ and conclude that we have a basis if these two sets are disjoint). Only $O(1)$ samplings of $P_{i,1}$ and $P_{i,2}$ are necessary before we find a basis. Each arithmetic operation over $\mathbb{F}_{p^{2\delta}}$ costs at most $O(\delta) = O(D)$ arithmetic operations over $\mathbb{F}_p$, so the overall complexity to find a basis is $O(D^3 \log(p))$ operations over $\mathbb{F}_p$ (and $O(D)$ operations over $\mathbb{Z}$).

Line 6 costs two evaluations of φ and line 9 costs eight scalar multiplications by the a_i, costing $O(D^2 \log(d))$ operations over $\mathbb{F}_p$ each. Hence, the total cost of the loop of lines 4–9 is

$$O(s(D^2 \log(d) + D^3 \log(p))) = O(\log(d)(D^2 \log(d) + D^3 \log(p)))$$

arithmetic operations over $\mathbb{F}_p$, $O(D \log(d))$ operations over $\mathbb{Z}$ and $2s = O(\log(d))$ evaluations of φ.

Finally, computing each F_i costs $O(q_i^{8e_i}) = O(D^8)$ arithmetic operations over $\mathbb{F}_{p^2}$ and computing the basis $\mathcal{C}_j$ $(i + 1 \leq j \leq s)$ on line 14 costs $8(s - i)$ point evaluations, each costing $O(q_i^{8e_i}) = O(D^8)$ arithmetic operations over an extension of degree $O(D)$ of $\mathbb{F}_{p^2}$. The total cost of the loop of lines 11 –16 is

$$O(sD^{10} + s^2 D^{10}) = O(D^{10} \log^2(d))$$

arithmetic operations over $\mathbb{F}_p$. □

2.7 Smoothness test and factorization with the ECM method

In this section, we explain how to test if an integer N is B-smooth and find its factorization if it is the case. A naive method would be to use trial division, but it is not optimal when B is subexponential (which will be the case in the paper). An alternate method would be to factor N with the General Number Field Sieve (GNFS) [9] and test if its prime factors are $\leq B$. However, GNFS underperforms with smooth integers. Hence, as Lenstra himself suggested [38, §2.12], we use the elliptic-curve factorization method (ECM) for that purpose. Finding a prime factor of N with this method is conjectured to take $L_\ell(1/2, \sqrt{2})\,\mathrm{polylog}(N)$ bit operations [38, Conjecture 2.10], where ℓ is the smallest prime divisor of N and with the usual notation

$$L_x(\alpha, \beta) := \exp\left((\beta + o(1))(\log(x))^\alpha (\log\log(x))^{1-\alpha}\right),$$

where $o(1)$ is for $x \to \infty$. Hence, to test the B-smoothness of N, we simply apply ECM to find a factor $k \mid N$ after expected time $L_B(1/2, \sqrt{2})\,\mathrm{polylog}(N)$. If the running time exceeds what it should be, it means that N is not B-smooth and we stop. Otherwise, we continue and try to factor k and N/k recursively until we have either completely factored N or concluded it is not B-smooth. Algorithm 2.2 follows.

Algorithm 2.2: SmoothFact determining if an integer is smooth and returning its factorization [38].

```
Data: An integer $N \in \mathbb{Z}_{>0}$ and a smoothness bound $B > 0$.
Result: $\perp$ if $N$ is not $B$-smooth, and primes $\ell_1, \cdots, \ell_r \leq B$ such that $N = \prod_{i=1}^r \ell_i$ otherwise.
1  if $N$ is prime then
2      if $N \leq B$ then
3          Return $N$;
4      else
5          Return $\perp$;
6      end
7  else
8      Use ECM to find a strict divisor $k \mid N$ in time $L_B(1/2, \sqrt{2})$;
9      if ECM does not terminate in time $L_B(1/2, \sqrt{2})$ then
10         Return $\perp$;
11     else
12         $R \longleftarrow$ SmoothFact$(k, B)$, $R' \longleftarrow$ SmoothFact$(N/k, B)$;
13         if $R = \perp$ or $R' = \perp$ then
14             Return $\perp$;
15         else
16             Return $R \cup R'$;
17         end
18     end
19 end
```

If r is the number of prime divisors of N (with multiplicity), then $r = O(\log(N))$ and Algorithm 2.2 can terminate with at most r calls to ECM, so it terminates in time $L_B(1/2, \sqrt{2})\,\mathrm{polylog}(N)$. This time complexity is still a conjecture and Pomerance provides a proved complexity result in [50, Theorem 2.1] but we use the conjectured complexity to obtain better results.

3 Reduction of $\mathfrak{O}$-orienting problem for special discriminants

We begin with a special case of the problem that depends on the discriminant of the imaginary quadratic order $\mathfrak{O}$. The key ideas from this special case provide a foundation for the general cases we consider in Sect. 4.

We remind the reader that all orientations discussed are *primitive* orientations. The oracle which we use to solve the Decisional $\mathfrak{O}$-Orienting Problem 2.3 is assumed to be perfect.

Let $d = \prod_{i=1}^{r} \ell_i$ be a product of small distinct primes. Let $\mathfrak{O}$ denote the maximal order of $K := \mathbb{Q}(\sqrt{-d})$, so $\Delta_{\mathfrak{O}} = -d$ if $d \equiv -1 \mod 4$ and $-4d$ otherwise. In particular, $(\Delta_{\mathfrak{O}}/\ell_i) = 0$ for all $i = 1, \ldots, r$ and $\mathfrak{O}$ is generated by $\omega := (1+\sqrt{-d})/2$ if $d \equiv -1 \mod 4$ and by $\omega := \sqrt{-d}$ otherwise. Hence, $\alpha := \sqrt{-d}$ generates $\mathfrak{O}$ if $d \not\equiv -1 \mod 4$ or $(\mathbb{Z} + 2\mathfrak{O})$ if $d \equiv -1 \mod 4$. We use an oracle which solves Problem 2.3 to find an endomorphism φ of E to which we map α, thus determining an embedding $\mathfrak{O} \hookrightarrow \mathrm{End}(E)$ either by mapping $\alpha = \omega$ to φ or $(1+\alpha)/2 = \omega$ to $(1+\varphi)/2$. We use the fact that the primes ℓ_i are ramified in K.

We walk the $\mathfrak{O}$-oriented ℓ_i-isogeny volcanoes in order to obtain the endomorphism φ on E which is the image of the generator ω under an embedding $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$. The ideals $\mathfrak{l}_i$ above ℓ_i in $\mathfrak{O}$ determine horizontal degree-ℓ_i isogenies between $\mathfrak{O}$-oriented curves, beginning and ending with E. To see this, we need the following fact about horizontal isogenies of $\mathfrak{O}$-oriented elliptic curves:

Proposition 3.1 [46, Proposition 4.1] *Let (E, ι) be an $\mathfrak{O}$-oriented supersingular elliptic curve and ℓ be a prime number. Then:*

(i) *If ℓ does not divide the conductor of $\mathfrak{O}$, there is no ascending, $(\Delta_{\mathfrak{O}}/\ell) + 1$ horizontal and $\ell - (\Delta_{\mathfrak{O}}/\ell)$ descending ℓ-isogenies.*
(ii) *If ℓ divides the conductor of $\mathfrak{O}$, there is one ascending, no horizontal and ℓ descending ℓ-isogenies.*

Let $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ be an orientation and $\varphi := \iota(\alpha)$. Then $\deg(\varphi) = N(\alpha) = \prod_{i=1}^{r} \ell_i$ so we may write $\varphi := \varphi_r \circ \cdots \circ \varphi_1$, where φ_i is an isogeny of degree ℓ_i for all $i \in \{1, \cdots, r\}$. For each i, let $\mathfrak{O}\ell_i = (\mathfrak{l}_i)^2$. The ideals $\mathfrak{l}_i$ determine the horizontal isogenies of $\mathfrak{O}$-oriented curves:

Lemma 3.2 *In the setting described above, all of the isogenies φ_i in the decomposition of φ are horizontal.*

Proof Since $N(\alpha) = \prod_{i=1}^{r} \ell_i$, we have $\mathfrak{O}\alpha = \prod_{i=1}^{r} \mathfrak{l}_i$, $\mathfrak{l}_i$ being the unique prime ideal of $\mathfrak{O}$ lying above ℓ_i for all $i \in \{1, \cdots, r\}$. Hence, the φ_i intervening in the decomposition of $\varphi = \iota(\alpha)$ are horizontal isogenies given by the action of $\mathfrak{l}_i$. □

Now, we describe the steps to obtain an endomorphism $\varphi = \varphi_r \circ \cdots \circ \varphi_1 \in \mathrm{End}(E)$ which will be the image of α. Let $E_0 := E$.

For $i = 0$, we find the unique isogeny $\varphi_1 : E_0 \longrightarrow E_1$ which corresponds to the action of $[\mathfrak{l}_1]$ on (E_0, ι) by computing each of the $\ell_1 + 1$ outgoing isogenies and querying our oracle to find the one whose codomain E_1 is in fact orientable by $\mathfrak{O}$. We continue this process to compute each φ_i by using the oracle to find the correct degree-ℓ_i isogeny to another $\mathfrak{O}$-orientable curve. At the last step, we compute the degree-ℓ_r isogeny from $E_{r-1} \longrightarrow E_r$ and then post-compose with an isomorphism $E_r \cong E_0$: We let φ_r denote this composition. See Algorithm 3.1 for the algorithmic description of this process.

A question arises: If $\psi_i : E_{i-1} \longrightarrow E_i'$ is an ℓ_i-isogeny with E_i' $\mathfrak{O}$-orientable, how do we know that ψ_i is the unique horizontal isogeny φ_i given by the action of $\mathfrak{l}_i$ on (E_{i-1}, ι)? In fact, φ_i and ψ_i could be distinct horizontal isogenies for distinct primitive orientations $(E_{i-1}, \iota) \neq (E_{i-1}, \iota')$ (as in Example 3.3). Or ψ_i could even be descending and E_i' $\mathfrak{O}$-oriented by a different orientation than the one induced by ψ_i, $(E_i, (\psi_i)_*(\iota))$.

Example 3.3 Let $p = 41$ and $E_0 : y^2 = x^3 + 1$ defined over $\mathbb{F}_{41^2} = \mathbb{F}_{41}[\zeta]$ with $\zeta^2 + \zeta + 1 = 0$. Consider the Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$ and the automorphism $\tau : (x, y) \mapsto (\zeta x, y)$. Then $\varphi := \pi + \tau$ satisfies the polynomial equation $\varphi^2 + \varphi + 42 = 0$ so it defines an orientation of the maximal order $\mathcal{O}_K := \mathbb{Z}\left[(1 + \sqrt{-167})/2\right]$ of the imaginary quadratic field $K := \mathbb{Q}(\sqrt{-167})$, mapping $(1 + \sqrt{-167})/2$ to φ.

The prime ideal 2 splits in K so there are two horizontal 2-isogenies and one descending 2-isogeny with domain E_0. However, all three of these isogenies have the same codomain E_1 (up to isomorphism) with j-invariant $j(E_1) = 3$. So E_1 is both $\mathcal{O}_K$-oriented and $(\mathbb{Z} + 2\mathcal{O}_K)$-oriented.

In order to guarantee a unique horizontal isogeny given by the action of $\mathfrak{l}_i$ on (E_{i-1}, ι), we assume $p > |\Delta_{\mathfrak{O}}| \max_{1 \le i \le r} \ell_i$ and prove that there is precisely one (primitive) $\mathfrak{O}$-orientation ι on E_{i-1}, which ensures that there is only one isogeny φ_i corresponding to the action of $[\mathfrak{l}_i]$ on (E_{i-1}, ι). We also prove that codomains of descending isogenies are not $\mathfrak{O}$-orientable. These are consequences of [32, Theorem 2'], that we recall below.

Theorem 3.4 [32, Theorem 2'] *Let $\mathcal{O} \subset B_{p,\infty}$ be a maximal order in the quaternion algebra ramifying at p and ∞. Let $j_i : \mathfrak{O}_i \hookrightarrow \mathcal{O}$ $(i \in \{1, 2\})$ be two primitive embeddings of orders in the same imaginary quadratic field $K := \mathbb{Q} \otimes \mathfrak{O}_1 = \mathbb{Q} \otimes \mathfrak{O}_2$ of respective discriminants Δ_i. Assume that $j_1(\mathfrak{O}_1) \neq j_2(\mathfrak{O}_2)$. Then $\Delta_1 \Delta_2 \geq p^2$.*

Corollary 3.5 *Let (E, ι) be a (primitively) $\mathfrak{O}$-oriented curve. Then*

(i) *If $|\Delta_{\mathfrak{O}}| < p$, then ι and $\bar{\iota} : \alpha \longmapsto \iota(\overline{\alpha})$ are the only two (primitive) $\mathfrak{O}$-orientations of E.*
(ii) *If $|\Delta_{\mathfrak{O}}|\ell < p$ and $\psi : (E, \iota) \longrightarrow (E', \iota')$ is a descending ℓ-isogeny, then E' is not $\mathfrak{O}$-orientable.*

Proof **(i)** Let $\iota' : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ be another $\mathfrak{O}$-orientation of E. Since $|\Delta_{\mathfrak{O}}| < p$, we must have $\iota'(\mathfrak{O}) = \iota(\mathfrak{O})$ by Theorem 3.4. Hence, $\iota'^{-1} \circ \iota$ is an automorphism of $\mathfrak{O}$, so it is either the identity or the complex conjugation. The result follows.

(ii) Suppose E' is $\mathfrak{O}$ orientable and let (E', ι'') be an $\mathfrak{O}$-orientation. Let $\mathfrak{O}' := \mathbb{Z} + \ell\mathfrak{O}$. Then $\psi : (E, \iota) \longrightarrow (E', \iota')$ being descending, (E', ι') is an $\mathfrak{O}'$-orientation and $\iota'(\mathfrak{O}') \neq \iota''(\mathfrak{O})$, so that $\Delta_{\mathfrak{O}'}\Delta_{\mathfrak{O}} \geq p^2$ by Theorem 3.4. But $\Delta_{\mathfrak{O}'}\Delta_{\mathfrak{O}} = \ell^2\Delta_{\mathfrak{O}}^2 < p^2$ by hypothesis. Contradiction. □

Remark 3.6 Corollary 3.5 holds for any imaginary quadratic order $\mathfrak{O}$, not only the special form we consider in this section.

Assuming $p > |\Delta_{\mathfrak{O}}| \max_{1 \le i \le r} \ell_i$, the orientation $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ is unique up to conjugation, the horizontal ℓ_1-isogeny $\varphi_1 : E_0 \longrightarrow E_1$ given by the action of $\mathfrak{l}_1$ is uniquely determined, and it is the only ℓ_1-isogeny with $\mathfrak{O}$-oriented codomain. In this case, φ_1 can be distinguished from other ℓ_1-isogenies by an oracle query. Similarly for each further $i \in \{2, ..., r\}$, the isogeny $\varphi_i : E_{i-1} \longrightarrow E_i$ given by the action of $[\mathfrak{l}_i]$ on $(E_{i-1}, (\varphi_{i-1} \circ \cdots \circ \varphi_1)_*(\iota))$ by computing each of the $\ell_i + 1$ isogenies and querying the oracle to find the one whose codomain E_i is orientable by $\mathfrak{O}$. In particular, the isogeny $\varphi_r : E_{r-1} \longrightarrow E_r$ corresponding to the action of $\mathfrak{l}_r$ on $(E_{r-1}, (\varphi_{r-1} \circ \cdots \circ \varphi_1)_*(\iota))$ will have codomain $E_r \cong E_0$. Indeed,

$$(E_r, (\varphi_r \circ \cdots \circ \varphi_1)_*(\iota)) = [\mathfrak{l}_1 \cdots \mathfrak{l}_r] \cdot (E_0, \iota) = [\alpha\mathfrak{O}] \cdot (E_0, \iota) \cong (E_0, \iota).$$

Possibly post-composing with this isomorphism, we have an endomorphism $\varphi = \varphi_r \circ \cdots \circ \varphi_1 \in \mathrm{End}(E)$ associated to the action of the ideal $\prod_{i=1}^r \mathfrak{l}_i = \alpha\mathfrak{O}$. It follows that $\varphi = \tau \circ \iota(\alpha)$ for some automorphism $\tau \in \mathrm{Aut}(E)$. We may post-compose φ by $\tau \in \mathrm{Aut}(E)$ until the result has trace zero, as α. The trace can be computed in polynomially many isogeny evaluations using Schoof's algorithm [58, Sect. 5].

Remark 3.7 (Isomorphisms) Assuming we are working with elliptic curves in Weierstrass form, all isomorphism formulae are known. To find an isomorphism $\beta : E_r \longrightarrow E_0$, we check the codomain formula for each isomorphism from E_r until E_0 is found.

There are additional automorphisms in the two special cases of $j = 1728$ and $j = 0$ [2, Figure 3.1, Sect. 6]. At each step $\varphi_i : E_{i-1} \to E_i$ where $j(E_i) = 0$ or 1728, we must decide whether or not to post-compose with these automorphisms. The automorphisms $[\pm 1]$ will not affect the resulting trace, but we must check one nontrivial automorphism for $j = 1728$ and two for $j = 0$. This can be done after the algorithm is completed, as the oracle calls will remain unaffected.

The additional running time of choosing isomorphisms can be bounded by a constant, so does not contribute to the overall complexity.

Example 3.8 Let $p = 83$ and $\mathfrak{O} = \mathbb{Z}[\sqrt{-21}]$, the ring of integers of $K = \mathbb{Q}(\sqrt{-21})$. We see p is inert in $\mathfrak{O}$ so K embeds into the quaternion algebra $\mathrm{End}^0(E)$ for any supersingular E over $\overline{\mathbb{F}_p}$. Now, let $E/\mathbb{F}_p^2$ be the $\mathfrak{O}$-oriented curve $y^2 = x^3 + x$, we find the orientation by finding an endomorphism ω with $N(\omega) = 21 = 3 \cdot 7$ and $Tr(\omega) = 0$. From E we pick a 3-isogeny to $y^2 = x^3 + 32x + 38\sqrt{-1}$, this is also $\mathfrak{O}$-oriented. Then we pick a horizontal 7-isogeny which has codomain $y^2 = x^3 + 26x$. This curve is isomorphic to E. By composing maps we get $\omega : E \longrightarrow E$. Finally we notice $\omega \neq -\tilde{\omega}$ so the endomorphism has a non-zero trace. But by post-composing with an automorphism ι on E, we get a trace-zero endomorphism of degree 21.

If α is a generator of $\mathfrak{O}$ ($d \not\equiv -1 \mod 4$), then φ determines an $\mathfrak{O}$-orientation and we are done. Otherwise, $\mathbb{Z}[\alpha]$ has index 2 in $\mathfrak{O}$ ($d \equiv -1 \mod 4$), and $\omega = (1 + \alpha)/2$ generates $\mathfrak{O}$. Then φ determines an imprimitive $\mathbb{Z}[\alpha]$-orientation of E. This orientation cannot be primitive, otherwise, we would have $\Delta_{\mathfrak{O}}\Delta_{\mathbb{Z}[\alpha]} \geq p^2$ i.e. $4\Delta_{\mathfrak{O}}^2 \geq p^2$, which is a contradiction since we assumed that $p > |\Delta_{\mathfrak{O}}| \max_{1 \leq i \leq r} \ell_i \geq 2|\Delta_{\mathfrak{O}}|$. It follows that $(\varphi + 1)/2$ is well defined and induces an $\mathfrak{O}$-orientation on E: $\omega = (\alpha + 1)/2 \longmapsto (\varphi + 1)/2$.

Remark 3.9 (Efficient representation) Knowing how to evaluate φ (as the composition $\varphi_r \circ \cdots \circ \varphi_1$), we efficiently evaluate $(\varphi + 1)/2$ as follows: if $P \in E(\mathbb{F}_{p^k})$, we find $P' \in E(\mathbb{F}_{p^{2k}})$ such that $[2]P' = P$ and compute $\varphi(P') + P'$. Assuming the ℓ_i are polynomial in $\log(d)$, the list of isogenies $(\varphi_r, \cdots, \varphi_1)$ defines an efficient representation of both φ and $(\varphi + 1)/2$.

We summarize all the steps to determine an $\mathfrak{O}$-orientation in Algorithm 3.1.

Algorithm 3.1: Algorithm to solve the $\mathfrak{O}$-Orienting Problem 2.4 with an oracle for the Decisional $\mathfrak{O}$-Orienting Problem 2.3, special discriminant.

```
Data: A supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$; the maximal order $\mathfrak{O}$ of $\mathbb{Q}(\sqrt{-d})$, where $d := \prod_{i=1}^{r} \ell_i$
      is a product of small distinct primes, where $p > |\Delta_{\mathfrak{O}}| \max_{1 \le i \le r} \ell_i$; an oracle IsOrientable$_{\mathfrak{O}}$
      for the Decisional $\mathfrak{O}$-Orienting Problem 2.3.
Result: If $E_0$ is $\mathfrak{O}$-orientable, an efficient representation (as defined in 2.8) of an endomorphism
        $\varphi_0 \in \mathrm{End}(E_0)$ defining an $\mathfrak{O}$-orientation of $E_0$.
1  if not IsOrientable$_{\mathfrak{O}}(E_0)$ then
2      Return "$E_0$ is not $\mathfrak{O}$-orientable";
3  end
4  Endo $\longleftarrow$ [];
5  for $i = 1$ to $r$ do
6      Compute the set $\{E_{i-1,k}\}_{k=1}^{\ell_i+1}$ of codomains of the $(\ell_i + 1)$ degree-$\ell_i$ isogenies from $E_{i-1}$;
7      Looking $\longleftarrow$ True;
8      $k \longleftarrow 1$;
9      while Looking and $k \le (\ell_i + 1)$ do
10         if IsOrientable$_{\mathfrak{O}(E_{i-1,k})}$ then
11             $E_i \longleftarrow E_{i-1,k}$;
12             Compute the degree-$\ell_i$ isogeny $\varphi_i : E_{i-1} \longrightarrow E_i$;
13             Append $\varphi_i$ to Endo;
14             Looking $\longleftarrow$ False;
15         end
16         $k \longleftarrow k + 1$;
17     end
18     Test all isomorphisms $\beta : E_r \longrightarrow E_0$ until $\beta \circ \varphi_r \circ \cdots \circ \varphi_1$ has trace zero;
19     Replace $\varphi_r$ by $\beta \circ \varphi_r$ in Endo;
20 end
21 Return Endo;
```

Theorem 3.10 *Let $d := \prod_{i=1}^{r} \ell_i$ be a product of small distinct primes, $\mathfrak{O}$ be the maximal order of $\mathbb{Q}(\sqrt{-d})$ and $p > |\Delta_{\mathfrak{O}}| \max_{1 \le i \le r} \ell_i$. Then, over $\mathbb{F}_{p^2}$, Algorithm 3.1 reduces the $\mathfrak{O}$-Orienting Problem (Problem 2.4) to the Decisional $\mathfrak{O}$-Orienting Problem (Problem 2.3) in polynomial time in $\log(p)$ and $\max_{1 \le i \le r} \ell_i$.*

Proof We justified above that this algorithm terminates and is correct. For all $i \in \{1, ..., r\}$, this algorithm computes the $\ell_i + 1$ curves which are ℓ_i-isogenous to E_{i-1}, which costs $\tilde{O}(\ell_i^2 \log(p))$ operations over $\mathbb{F}_{p^2}$ by Sect. 2.4. It calls the oracle IsOrientable$_{\mathfrak{O}}$ $\ell_i + 1$ times and computes one ℓ_i-isogeny between $j(E_{i-1})$ and $j(E_i)$, which costs on average $\tilde{O}(\ell_i^2 \log(p))$ operations over $\mathbb{F}_{p^2}$ by Sect. 2.5. The number of isomorphisms $\beta : E_r \longrightarrow E_0$ is $O(1)$. Using [58, Sect. 5], we compute the trace of $\beta \circ \varphi_r \circ \cdots \circ \varphi_1$ on line 18 of Algorithm 3.1 in polynomial time in $\log(p)$, $r = O(\log(p))$ and $\max_{1 \le i \le r} \ell_i$. Operations over $\mathbb{F}_{p^2}$ have a polynomial cost in $\log(p)$ in terms of bit operations. Hence, the total cost is polynomial in $\log(p)$ and $\max_{1 \le i \le r} \ell_i$. $\square$

4 Solving the $\mathfrak{O}$-orienting problem with a decision oracle

We remind the reader that all orientations discussed are *primitive* orientations. The oracle which we use to solve the Decisional $\mathfrak{O}$-Orienting Problem 2.3 is assumed to be perfect.

4.1 Description of the algorithms

Let $\mathfrak{O}$ be an imaginary quadratic order with general discriminant $\Delta_{\mathfrak{O}}$. Given access to an oracle $\mathsf{IsOrientable}_{\mathfrak{O}}$ for the Decisional $\mathfrak{O}$-Orienting Problem 2.3, we solve the $\mathfrak{O}$-Orienting Problem 2.4 finding a an $\mathfrak{O}$-orientation $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ of any given supersingular elliptic curve $E/\mathbb{F}_{p^2}$ if it exists, and returning Null otherwise.

The idea is similar the case of special discriminant considered in Sect. 3. We compute an endomorphism corresponding to a generator of $\mathfrak{O}$ as a chain of horizontal isogenies of small degrees. However, two difficulties arise. First, the canonical generator $\omega := (s + \sqrt{\Delta_{\mathfrak{O}}})/2$ with $s := \Delta_{\mathfrak{O}} \mod 2$ of $\mathfrak{O}$ is not smooth in general. We have to find another smooth generator θ of $\mathfrak{O}$. Second, if we denote $\varphi := \iota(\theta)$ and decompose $\varphi := \varphi_r \circ \cdots \circ \varphi_1$ as a product of horizontal isogenies of degrees $\ell_i, \cdots, \ell_r$ respectively, we may not be able to find the φ_i simply by using the oracle $\mathsf{IsOrientable}_{\mathfrak{O}}$ as in Sect. 3. We are no longer guaranteed that $\ell_i \mid \Delta_{\mathfrak{O}}$, so there may be $1 + (\Delta_{\mathfrak{O}}/\ell_i) = 2$ horizontal isogenies of degree ℓ_i from a $\mathfrak{O}$-oriented elliptic curve. To search for φ, starting at root E we fill a binary tree whose nodes are $\mathfrak{O}$-oriented elliptic curves and edges are horizontal isogenies. We call such a tree an $\mathfrak{O}$-oriented $(\ell_1, \cdots, \ell_r)$-isogeny tree, see Definition 4.1. The endomorphism φ is a branch of this tree with leaf E.

Definition 4.1 An $\mathfrak{O}$-oriented $(\ell_1, \cdots, \ell_r)$-isogeny tree is a binary tree of height r whose nodes are (primitively) $\mathfrak{O}$-oriented supersingular elliptic curves and such that every node E_{i-1} of depth $i \in \{1, \cdots, r\}$ has children that are horizontally ℓ_i-isogenous to E_{i-1}.

To optimize the tree search, we propose a meet-in-the middle strategy where two half-depth such trees are computed starting at E instead of a single one:

(1) Find a generator θ of $\mathfrak{O}$ of B-smooth norm $N(\theta) := \prod_{i=1}^{r} \ell_i$.
(2) Starting at E, compute $\mathfrak{O}$-oriented $(\ell_1, \cdots, \ell_s)$-isogeny tree $\mathcal{T}_1$ and an $\mathfrak{O}$-oriented $(\ell_{s+1}, \cdots, \ell_r)$-isogeny tree $\mathcal{T}_2$ (with $s \simeq r/2$).
(3) Find a matching leaf in $\mathcal{T}_1$ and $\mathcal{T}_2$.
(4) Extract the corresponding endomorphism $\varphi \in \mathrm{End}(E)$.
(5) Infer from $\varphi = \iota(\theta)$ an efficient representation of the canonical generator $\varphi_0 := \iota(\omega)$ (in the sense of Definition 2.8).

We explain each step in detail in the following paragraphs.

4.1.1 Finding a smooth norm generator

Let $\mathfrak{O}$ be an imaginary quadratic order and ω be a generator. We want to find another generator θ of $\mathfrak{O}$ with smooth norm $N(\theta) = \prod_{i=1}^{r} \ell_i$. The computation of $\varphi = \varphi_r \circ \cdots \circ \varphi_1$ associated to θ is exponential in the ℓ_i and r, so we require the ℓ_i and 2^r to be subexponential in $\log(|\Delta_{\mathfrak{O}}|)$. For technical reasons (see Lemma 4.3), $N(\theta)$ should also be non-square and coprime to $\Delta_{\mathfrak{O}}$. In summary, we look for a generator θ of ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth norm, in the sense of Definition 4.2, with B and 2^{r_m} subexponential in $\log(|\Delta_{\mathfrak{O}}|)$.

Definition 4.2 An integer $N \in \mathbb{N}$ is (B, r_m, d)-*smooth* when its decomposition into prime factors $N = \prod_{i=1}^{r} \ell_i$ satisfies $r \leq r_m$, $\ell_i \leq B$, and $\ell_i \nmid d$ for all $i \in \{1, \cdots, r\}$. We say that N is *ns*-(B, r_m, d) *-smooth* when it is (B, r_m, d)-smooth and not a square.

We look for θ of ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth norm the form $\theta := a + \omega$ with $a \in \mathbb{Z}$ to be determined. There is no better known method to find a than to sample a at random and to test whether $N(a+\omega)$ is ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth. To make sure $N(a+\omega)$ is close to $N(\omega)$, we sample $a \in \{-\lfloor\sqrt{N(\omega)}\rfloor, \cdots, \lfloor\sqrt{N(\omega)}\rfloor\}$. We have $N(\omega) = (|\Delta_{\mathfrak{O}}| + t^2)/4$ with $t := \mathrm{Tr}(\omega) \in \{0, 1\}$. It follows that:

$$\frac{|\Delta_{\mathfrak{O}}|}{4} = N(-t/2+\omega) \le N(a+\omega) \le N(-\sqrt{N(\omega)}+\omega) \le |\Delta_{\mathfrak{O}}|$$

Since B is subexponential in $\log(|\Delta_{\mathfrak{O}}|)$, the optimal known way to test the B-smoothness of $N(a+\omega)$ is the method introduced in Sect. 2.7 using ECM with time complexity $L_B(1/2, \sqrt{2})$. Algorithm 4.1 presenting the search for $\theta = a + \omega$ follows.

Algorithm 4.1: FindSmoothGen finding a smooth generator of an imaginary quadratic order $\mathfrak{O}$.

```
Data: The discriminant $\Delta_{\mathfrak{O}}$ of an imaginary quadratic order $\mathfrak{O}$ and smoothness parameters $B > 0$ and
      $r_m \in \mathbb{Z}_{>0}$.
Result: A generator $\theta$ of $\mathfrak{O}$ having ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth norm $N(\theta)$ and its prime factors
        $\ell_1, \cdots, \ell_r$ (with multiplicity).
1  $s \longleftarrow \Delta_{\mathfrak{O}} \mod 2$;
2  $\omega \longleftarrow (s + \sqrt{\Delta_{\mathfrak{O}}})/2$;
3  repeat
4      repeat
5          repeat
6              Sample $a \overset{\$}{\longleftarrow} \{-\lfloor\sqrt{N(\omega)}\rfloor, \cdots, \lfloor\sqrt{N(\omega)}\rfloor\}$;
7          until $N(a+\omega) \wedge \Delta_{\mathfrak{O}} = 1$ and $\sqrt{N(a+\omega)} \notin \mathbb{Z}$;
8          $R \longleftarrow$ SmoothFact$(N(a+\omega), B)$ (Algorithm 2.2);
9      until $R \neq \perp$;
10     $\ell_1, \cdots, \ell_r \longleftarrow R$;
11 until $r \le r_m$;
12 Return $a + \omega$ and $\ell_1, \cdots, \ell_r$;
```

4.1.2 Filling the $\mathfrak{O}$-oriented isogeny trees

Let $E/\mathbb{F}_{p^2}$ be an $\mathfrak{O}$-orientable elliptic curve and splitting primes $\ell_1, \ldots, \ell_s \le B$. We explain here how to fill $\mathcal{T}$, the $\mathfrak{O}$-oriented $(\ell_1, \ldots, \ell_s)$-isogeny tree starting at E.

We assume $p > B|\Delta_{\mathfrak{O}}|$ so any $\mathfrak{O}$-orientable curve admits a unique $\mathfrak{O}$-orientation up to conjugation by Corollary 3.5(i). Hence, every node of $\mathcal{T}$ can be represented by j-invariant (the root $E_0 := E$ included). If E_{i-1} is a node of depth $i \in \{1, \cdots, s\}$ of $\mathcal{T}$, its children $E_{i,1}$ and $E_{i,2}$ are the only two $\mathfrak{O}$-orientable curves that are ℓ_i-isogenous to E_i, given by the action of ideals $\mathfrak{l}_i$, $\overline{\mathfrak{l}_i}$ above ℓ_i. As in Sect. 3, to find $E_{i,1}$ and $E_{i,2}$ we compute the codomain j-invariants of all degree-ℓ_i isogenies $E_i \longrightarrow E'$ and apply the decision oracle to see which are $\mathfrak{O}$-orientable. Determining such j-invariants can be done using modular polynomials in $\tilde{O}(\ell_i^2 \log(p))$ operations over $\mathbb{F}_{p^2}$, as in Sect. 2.4. The tree filling algorithm TreeFill (Algorithm 4.2) follows.

4.1.3 From a tree match to a generating endomorphism

Assume we have found θ, a generator of $\mathfrak{O}$ with ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth norm $N(\theta) = \prod_{i=1}^{r} \ell_i$. Let $\iota : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ denote the orientation with $\varphi := \iota(\theta)$. Then, we may decompose $\varphi := \varphi_r \circ \cdots \circ \varphi_1$, where φ_i is an ℓ_i-isogeny for all $i \in \{1, \cdots, r\}$.

Algorithm 4.2: TreeFill, the $\mathfrak{O}$-oriented isogeny tree filling algorithm.

```
Data: An imaginary quadratic order $\mathfrak{O}$ such that $|\Delta_{\mathfrak{O}}| < p/B$, an $\mathfrak{O}$-orientable curve $E/\mathbb{F}_{p^2}$, splitting
      primes $\ell_1, \cdots, \ell_s \leq B$ and an oracle IsOrientable$_{\mathfrak{O}}$ for the Decisional $\mathfrak{O}$-Orienting Problem 2.3.
Result: The $\mathfrak{O}$-oriented $(\ell_1, \cdots, \ell_s)$-isogeny tree $\mathcal{T}$ starting at $E$.
1  Initialize $\mathcal{T}$ at $E_0 := E$;
2  for $i = 1$ to $s$ do
3      for $j(E_{i-1}) \in$ Leaves($\mathcal{T}$) do
4          Compute $\Phi_{\ell_i}(j(E_{i-1}), Y)$;
5          Find $S_i \subset \mathbb{F}_{p^2}$, the set of roots of $\Phi_{\ell_i}(j(E_{i-1}), Y)$;
6          for $j(E_i) \in S_i$ do
7              if IsOrientable$_{\mathfrak{O}}(j(E_i))$ then
8                  Append $j(E_i)$ as a child of $j(E_{i-1})$ in $\mathcal{T}$;
9              end
10         end
11     end
12 end
13 Return $\mathcal{T}$;
```

Lemma 4.3 *Assuming $N(\theta) = \deg(\varphi)$ is coprime with $\Delta_{\mathfrak{O}}$, all the isogenies φ_i in the decomposition of φ are horizontal.*

Proof Let $\mathfrak{O}_1$ be the order associated to $\iota_1 := (\varphi_1)_*(\iota)$ i.e. such that $\iota_1(\mathfrak{O}_1) = \mathrm{End}(\varphi_1(E)) \cap \iota_1(K)$. Since $N(\theta) = \deg(\varphi)$ is coprime with $\Delta_{\mathfrak{O}}$, ℓ_1 does not divide the conductor of $\mathfrak{O}$ so φ_1 is horizontal or descending by Proposition 3.1(i). It follows that $\mathfrak{O}_1 \subseteq \mathfrak{O}$. Besides,

$$\iota_1(\theta) = \frac{1}{\ell_1}\varphi_1 \circ \iota(\theta) \circ \widehat{\varphi_1} = \varphi_1 \circ \varphi_r \circ \cdots \circ \varphi_2 \in \mathrm{End}(\varphi_1(E)).$$

Hence, $\theta \in \mathfrak{O}_1$ so $\mathfrak{O} \subseteq \mathfrak{O}$ since θ generates $\mathfrak{O}$. Consequently, $\mathfrak{O}_1 = \mathfrak{O}$ and φ_1 is horizontal. We obtain easily by induction that $\varphi_2, \cdots, \varphi_r$ are also horizontal.

□

Since the φ_i are horizontal, we may use $\mathfrak{O}$-oriented isogeny trees to find these isogenies. Let $s := \lfloor r/2 \rfloor$, $\mathcal{T}_1$ the $\mathfrak{O}$-oriented $(\ell_1, \cdots, \ell_s)$-isogeny tree starting at $E_0 := E$ and $\mathcal{T}_2$ the $\mathfrak{O}$-oriented $(\ell_{s+1}, \cdots, \ell_r)$-isogeny tree starting at E. Assume we have found a common leaf E_s in $\mathcal{T}_1$ and $\mathcal{T}_2$. The branch of $\mathcal{T}_1$ of leaf E_s is a chain of horizontal ℓ_i-isogenies $\psi_i : E_{i-1} \longrightarrow E_i$ for $i \in \{1, \cdots, s\}$ and the branch of $\mathcal{T}_2$ of leaf E_s (taken depth first) is a chain of horizontal ℓ_i-isogenies $\psi_i : E_{i-1} \longrightarrow E_i$ for $i \in \{s+1, \cdots, r\}$, with $E_r = E_0 = E$. The isogeny $\psi := \psi_r \circ \cdots \circ \psi_1$ is a horizontal isogeny of degree $\prod_{i=1}^{r} \ell_i = N(\theta)$, but we do not know *a priori* if $\psi = \varphi = \iota(\theta)$.

Lemma 4.4 *Let (E_0, ι) be an $\mathfrak{O}$-oriented supersingular elliptic curve and $\psi \in \mathrm{End}(E_0)$ a horizontal endomorphism of degree coprime to p. Then, there exists $\alpha \in \mathfrak{O}$ such that $\psi = \iota(\alpha)$.*

Proof Since ψ is horizontal, $\psi_*(\iota)$ defines an $\mathfrak{O}$-orientation on E_0, like ι. Since $|\Delta_{\mathfrak{O}}| < p$, by Theorem 3.4, we must have $\psi_*(\iota)(\mathfrak{O}) = \iota(\mathfrak{O})$, so that $\psi_*(\iota) = \iota$ or $\psi_*(\iota) = \bar{\iota}$, where $\bar{\iota}(\alpha) := \iota(\overline{\alpha})$ for all $\alpha \in K$.

If $\psi_*(\iota) = \iota$, ψ commutes with $\iota(K)$ ($K := \mathfrak{O} \otimes_{\mathbb{Z}} \mathbb{Q}$), so $\psi \in \iota(K) \cap \mathrm{End}(E_0) = \iota(\mathfrak{O})$ and $\psi = \iota(\alpha)$ for some $\alpha \in \mathfrak{O}$.

Suppose $\psi_*(\iota) = \bar{\iota}$. As Onuki proved in [46, Proposition 3.3 and Theorem 3.4], $(E, \bar{\iota})$ and $(E^{(p)}, (\pi_p)_*(\iota))$ are in the same orbit of the action of $\mathrm{Cl}(\mathfrak{O})$ on the set $\mathrm{SS}_{\mathfrak{O}}^{pr}(p)$ of (primitively) $\mathfrak{O}$-oriented supersingular elliptic curves over $\mathbb{F}_{p^2}$ ($\pi_p : E \longrightarrow E^{(p)}$ being the p-Frobenius isogeny). Hence, there exists an ideal $\mathfrak{b} \subset \mathfrak{O}$ of norm coprime with p, such that $(E, \bar{\iota}) = \mathfrak{b} \cdot (E^{(p)}, (\pi_p)_*(\iota))$, so that $\psi_*(\iota) = \bar{\iota} = (\varphi_{\mathfrak{b}} \circ \pi_p)_*(\iota)$. Consequently, $\widehat{\pi_p} \circ \widehat{\varphi_{\mathfrak{b}}} \circ \psi$ commutes with $\iota(K)$, so there exists $\alpha \in \mathfrak{O}$ such that $\widehat{\pi_p} \circ \widehat{\varphi_{\mathfrak{b}}} \circ \psi = \iota(\alpha)$ and $p \mid N(\alpha)$. Since $\mathrm{SS}_{\mathfrak{O}}^{pr}(p)$ is not empty (it contains E_0), p is either inert or ramified in K by [46, Proposition 3.2]. The prime p cannot be ramified, otherwise we would have $p \mid \Delta_{\mathfrak{O}}$, so $|\Delta_{\mathfrak{O}}| \geq p$. If p is inert and $p \mid N(\alpha)$, then $p \mid \alpha$ so that $p^2 \mid N(\alpha)$ and $p \mid \deg(\psi) N(\mathfrak{b})$. Since $N(\mathfrak{b})$ is coprime with p, $p \mid \deg(\psi)$ which contradicts our assumption.

It follows that $\psi_*(\iota) = \iota$. □

Lemma 4.5 *Let $\theta := a + \omega \in \mathfrak{O}$, with $a \in \mathbb{Z}$, $|a| \leq \sqrt{N(\omega)}$. Assume $N(\theta)$ is not a square and $\Delta_{\mathfrak{O}} \neq -3, -4$. The only $\alpha \in \mathfrak{O}$ such that $N(\alpha) = N(\theta)$ are $\alpha = \pm\theta, \pm\bar{\theta}$.*

Proof Let $\alpha := b + c\omega \in \mathfrak{O}$ with $b, c \in \mathbb{Z}$ such that $N(\alpha) = N(\theta)$. Then

$$b^2 + tbc + c^2 n = N(\alpha) = N(\theta) = a^2 + ta + n, \tag{7}$$

with $t := \mathrm{Tr}(\omega) \in \{0, 1\}$ and $n := N(\omega) = (t^2 + |\Delta_{\mathfrak{O}}|)/4$.

If $c^2 > 1$, the minimum value of $b^2 + tbc + c^2 n$ is reached when $b = -ct/2$, so

$$b^2 + tbc + c^2 n \geq \left(n - \frac{t^2}{4}\right) c^2 = \frac{|\Delta_{\mathfrak{O}}| c^2}{4} \geq |\Delta_{\mathfrak{O}}|. \tag{8}$$

But, by (7) and since $|a| \leq \sqrt{n}$, we have:

$$N(\theta) \leq 2n + t\sqrt{n} < |\Delta_{\mathfrak{O}}|$$

which contradicts (8).

So $c^2 \leq 1$ and $c \in \{0, \pm 1\}$. If $c = 0$, then $N(\theta)$ is a square which is not possible. If $c = 1$, then (7) becomes $(a - b)(a + b + t) = 0$ and we have $b = a$ or $b = -a - t$. If $c = -1$, then (7) becomes $(a + b)(a - b + t) = 0$ and we have $b = a$ or $b = a - t$. Hence, $(b, c) \in \{(a, 1), (-a - t, 1), (-a, -1), (t + a, -1)\}$ and $\alpha \in \{\pm\theta, \pm\bar{\theta}\}$. □

Remark 4.6 The cases of $\Delta_{\mathfrak{O}} = -3, -4$ are excluded from this lemma because in those cases, we have a very simple way to find the orientation:

If $\Delta_{\mathfrak{O}} = -3$, then $\mathfrak{O} = \mathbb{Z}[\zeta_3]$, with $\zeta_3 := (1 + \sqrt{-3})/2$ so any elliptic curve E that is $\mathfrak{O}$-oriented contains an automorphism of order 3. By [60, Theorem III.10.1], we must have $j(E) = 0$, so E is given by the Weierstrass equation $y^2 = x^3 + 1$ (up to isomorphism), and ζ_3 corresponds to the automorphism $(x, y) \in E \longmapsto (\xi_3 x, y) \in E$, where ξ_3 is a primitive third root of unity in $\mathbb{F}_{p^2}$.

Similarly, if $\Delta_{\mathfrak{O}} = -4$, then $\mathfrak{O} = \mathbb{Z}[i]$ so any elliptic curve E that is $\mathfrak{O}$-oriented contains an automorphism of order 4. By [60, Theorem III.10.1], we must have $j(E) = 1728$, so E is given by the Weierstrass equation $y^2 = x^3 + x$ (up to isomorphism), and i corresponds to the automorphism $(x, y) \in E \longmapsto (x, ay) \in E$, where a is a square root of -1 in $\mathbb{F}_{p^2}$.

By Lemmas 4.4 and 4.5, we must have $\psi = \pm\iota(\theta) = \pm\varphi$ or $\psi = \pm\iota(\bar{\theta}) = \pm\widehat{\varphi}$. The sign can be determined by computing $\mathrm{Tr}(\psi)$ using a generalization of Schoof's algorithm [58, Sect. 5] and comparing to $\mathrm{Tr}(\theta)$. We recover ι or $\bar{\iota} : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$ by mapping θ to $\pm\psi$.

However, the factors ψ_i of ψ have subexponential degree so they do not provide an efficient representation of ψ (enabling to evaluate ψ in polynomial time for instance). We apply EfficientRep Algorithm 2.1 to get an efficient representation of $\iota(\omega)$ or $\iota(\bar{\omega}) = \pm\psi - [a]$. The search to decision reduction Algorithm 4.3 follows.

For efficiency, only j-invariants are stored in the trees and not the ℓ_i-isogenies relating them so we use the method of Sect. 2.5 to recover them in time $\tilde{O}(\ell_i^2 \log(p))$.

Algorithm 4.3: Algorithm to solve the $\mathfrak{O}$-Orienting Problem 2.4 with an oracle for the Decisional $\mathfrak{O}$-Orienting Problem 2.3.

```
Data: A supersingular elliptic curve $E/\mathbb{F}_{p^2}$, smoothness parameters $B, r_m, D$, an imaginary quadratic
      order $\mathfrak{O}$ of discriminant $\Delta_{\mathfrak{O}} \neq -3, -4$ such that $|\Delta_{\mathfrak{O}}| < p/B$ and canonical generator $\omega$ along
      with an oracle $\mathsf{IsOrientable}_{\mathfrak{O}}$ for the Decisional $\mathfrak{O}$-Orienting Problem 2.3.
Result: If $E$ is $\mathfrak{O}$-orientable, an efficient representation $F$ (as defined in 2.8) of an endomorphism
        $\varphi_0 \in \mathrm{End}(E)$ such that $\deg(\varphi_0) = N(\omega)$ and $\mathrm{Tr}(\varphi_0) = \mathrm{Tr}(\omega)$, where $\omega$ is the canonical
        generator of $\mathfrak{O}$.
1  if not $\mathsf{IsOrientable}_{\mathfrak{O}}(E)$ then
2      Return $\perp$;
3  end
4  $\theta, \ell_1, \cdots, \ell_r \longleftarrow \mathsf{FindSmoothGen}(\Delta_{\mathfrak{O}}, B, r_m)$ (Algorithm 4.1);
5  $s \longleftarrow \lfloor r/2 \rfloor$;
6  $\mathcal{T}_1 \longleftarrow \mathsf{TreeFill}(\mathfrak{O}, E, \ell_1, \cdots, \ell_s)$ (Algorithm 4.2);
7  $\mathcal{T}_2 \longleftarrow \mathsf{TreeFill}(\mathfrak{O}, E, \ell_r, \ell_{r-1}, \cdots, \ell_{s+1})$;
8  Search for a matching leaf $j(E_s) \in \mathrm{Leaves}(\mathcal{T}_1) \cap \mathrm{Leaves}(\mathcal{T}_1)$;
9  Recover from the branch of leaf $j(E_s)$ in $\mathcal{T}_1$ the $\ell_i$-isogeny $\psi_i : E_{i-1} \longrightarrow E_i$ for all $i \in \{1, \cdots, s\}$
   (using Section 2.5);
10 Recover from the branch of leaf $j(E_s)$ in $\mathcal{T}_2$ the $\ell_i$-isogeny $\psi_i : E_{i-1} \longrightarrow E_i$ for all
   $i \in \{s+1, \cdots, r\}$;
11 Let $\psi := \psi_r \circ \cdots \circ \psi_1$;
12 Compute $\mathrm{Tr}(\psi)$ using Schoof's algorithm [58, Section 5];
13 $s \longleftarrow \Delta_{\mathfrak{O}} \mod 2$;
14 $\omega \longleftarrow (s + \sqrt{\Delta_{\mathfrak{O}}})/2$;
15 Let $\epsilon := \mathrm{Tr}(\psi)/\mathrm{Tr}(\theta)$ and $\theta := a + \omega$;
16 $F \longleftarrow \mathsf{EfficientRep}([\epsilon] \circ \psi - [a], D)$ (Algorithm 2.1);
17 Return $F$;
```

4.2 Complexity analysis

In the following, we shall count arithmetic operations over various rings ($\mathbb{Z}$, $\mathbb{F}_{p^2}$ and extensions of $\mathbb{F}_{p^2}$). To provide a unified way to present time complexity results, we shall count bit operations. We shall say an algorithm terminates in time $O(f(n))$ when it uses at most $O(f(n))$ bit operations. We shall denote by $M(p)$ the maximal number of bit operations required for an operation over $\mathbb{F}_p$ (addition, substraction, multiplication, inversion or sampling of one element).

4.2.1 Complexity of the smooth norm search (Algorithm 4.1)

To estimate the complexity of Algorithm 4.1, we need to determine the probability that $N(a+\omega)$ is ns-$(B, r_m, \Delta_{\mathfrak{O}})$-smooth. We have proven results on the distribution of B-smooth integers among random integers but not for random values of quadratic integer polynomials. For that reason, we introduce the following heuristic assumption.

Heuristic 4.7 *Let $f := X^2 - tX + N \in \mathbb{Z}[X]$, a following the uniform distribution in $\{-\lfloor\sqrt{N}\rfloor, \cdots, \lfloor\sqrt{N}\rfloor\}$, and b following the uniform distribution in $\{0, \cdots, N\}$. Then there exist constants $C > 0$, $c > 0$ such that for all $N \in \mathbb{Z}_{>0}$, $\log^c(N) \leq B \leq N$, $\log(N)/log(B) \leq r \leq \log_2(N)$ and $d \leq 4N$, we have:*

$$\mathbb{P}(f(a) \textit{ is ns} - (B, r, d) - \textit{smooth}) \geq C \cdot \mathbb{P}(b \textit{ is ns} - (B, r, d) - \textit{smooth}).$$

This heuristic assumption is supported by an estimate on the probability for a given polynomial to take B-smooth values, which is very similar to proven estimates on the B-

smoothness probability of integers [17, Theorem 1]. Such an estimate on polynomials has been proved in [42, Theorem 1.1] under a dual hypothesis on the number of prime values of polynomials when B is in a very tight range. It has been conjectured [29, Equation 1.20] that this result holds for broader values of B.

Lemma 4.8 *Let $\Psi_r(x, y, d)$ denote the number of (y, r, d)-smooth integers $\leq x$:*

$$\Psi_r(x, y, d) = \#\left\{ n \leq x \;\middle|\; n = \prod_{i=1}^{s} \ell_i,\ s \leq r \text{ and } \forall\, 1 \leq i \leq s,\ \ell_i \leq y \text{ and } \ell_i \nmid d \right\}.$$

Then, if $r \geq \log(x)/\log(y)$,

$$\Psi_r(x, y, d) \geq \binom{\pi(y) - \pi(z) - \omega_y(d) + \lfloor \frac{\log(x)}{\log(y)} \rfloor}{\pi(y) - \pi(z) - \omega_y(d)},$$

with $z := x^{1/r}$, $\pi(t)$ the number of prime numbers $\leq t$ and $\omega_y(d)$ the number of distinct prime divisors $\leq y$ of d.

Proof The proof follows from [17, §2]. We have the following inequalities (following from set inclusions):

$$\begin{aligned}
\Psi_r(x, y, d) &= \#\Bigg\{ (\alpha_\ell)_{\substack{\ell \leq y \\ \ell \nmid d}} \in \mathbb{N}^{\pi(y) - \omega_y(d)} \;\Bigg|\; \#\{\ell \leq y, \ell \nmid d \mid \alpha_\ell \neq 0\} \leq r \\
&\qquad \text{and } \sum_{\ell \leq y} \alpha_\ell \log(\ell) \leq \log(x) \Bigg\} \\
&\geq \#\left\{ (\alpha_\ell)_{\substack{z < \ell \leq y \\ \ell \nmid d}} \in \mathbb{N}^{\pi(y) - \pi(z) - \omega_y(d)} \;\middle|\; \sum_{\substack{z < \ell \leq y \\ \ell \nmid d}} \alpha_\ell \log(\ell) \leq \log(x) \right\} \\
&\geq \#\left\{ (\alpha_\ell)_{\substack{z < \ell \leq y \\ \ell \nmid d}} \in \mathbb{N}^{\pi(y) - \pi(z) - \omega_y(d)} \;\middle|\; \sum_{\substack{z < \ell \leq y \\ \ell \nmid d}} \alpha_\ell \leq \left\lfloor \frac{\log(x)}{\log(y)} \right\rfloor \right\}.
\end{aligned}$$

To conclude, we compute the cardinality of

$$S(k, n) := \left\{ (\alpha_1, \cdots, \alpha_k) \in \mathbb{N}^k \;\middle|\; \sum_{i=1}^{k} \alpha_i \leq n \right\}$$

for $k, n \in \mathbb{Z}_{>0}$ and apply it to the last set in the inequalities above. The set $S(k, n)$ is in bijection with the subsets of k elements in $\{1, \cdots, n + k\}$, via the maps:

$$\begin{aligned}
\{s_1 < \cdots < s_k\} &\longmapsto (s_1 - 1, s_2 - s_1 - 1, \cdots, s_k - s_{k-1} - 1) \\
(\alpha_1, \cdots, \alpha_k) &\longmapsto \{\alpha_1 + 1, \alpha_1 + \alpha_2 + 2, \cdots, \alpha_1 + \cdots + \alpha_k + k\}.
\end{aligned}$$

It follows that

$$\#S(k, n) = \binom{n + k}{k}.$$

□

Lemma 4.9 *Let $\psi(x,y)$ be the number of y-smooth numbers $\leq x$. Assume that* $\log(y) \ll \log(x)$ *and* $\log(y) \gg \log\log(x)$. *Then*

$$\log\left(\frac{\psi(x,y)}{x}\right) \sim -\frac{\log(x)\log\log(x)}{\log(y)}.$$

Proof It follows from [17, Theorem 1] that for all $2 < y \leq x$:

$$\log\psi(x,y) = \left(\log\left(1+\frac{y}{\log(x)}\right)\frac{\log(x)}{\log(y)} + \log\left(1+\frac{\log(x)}{y}\right)\frac{y}{\log(y)}\right) \cdot \left(1 + O\left(\frac{1}{\log(y)}\right) + O\left(\frac{1}{\log\log(x)}\right) + O\left(\left(1+\frac{\log(x)}{\log(y)}\right)^{-1}\right)\right). \tag{9}$$

Since $\log(y) \gg \log\log(x)$, we have $y \gg \log^2(x)$, so that

$$\begin{aligned}\log\left(1+\frac{y}{\log(x)}\right)\frac{\log(x)}{\log(y)} &= \left(\log\left(\frac{y}{\log(x)}\right) + \log\left(1+\frac{\log(x)}{y}\right)\right)\frac{\log(x)}{\log(y)}\\ &= \log(x) - \frac{\log(x)\log\log(x)}{\log(y)} + \frac{\log^2(x)}{y\log(y)}(1+o(1))\\ &= \log(x) - \frac{\log(x)\log\log(x)}{\log(y)} + o(1)\end{aligned}$$

and

$$\log\left(1+\frac{\log(x)}{y}\right)\frac{y}{\log(y)} = \frac{\log(x)}{\log(y)}(1+o(1)).$$

It follows finally by 9 that

$$\log\left(\frac{\psi(x,y)}{x}\right) \sim -\frac{\log(x)\log\log(x)}{\log(y)}.$$

□

Lemma 4.10 *Let $\psi_r^*(x,y,d)$ be the number of ns-(y,r,d)-smooth integers $\leq x$. Let $z := x^{1/r}$ and $\varepsilon := z/y$. Assume that $r \geq \log(x)/\log(y)$, $d = O(x)$, $\log(y) \ll \log(x)$, $\log(y) \gg \log\log(x)$ and $\log(1-\varepsilon) \ll \log\log(y)$. Then*

$$\log\left(\frac{\psi_r^*(x,y,d)}{x}\right) \sim -\frac{\log(x)\log\log(x)}{\log(y)}$$

as $x, y, r, d \longrightarrow +\infty$.

Proof The number of squares smaller than x being bounded by $\sqrt{x}$, we have

$$\psi_r^*(x,y,d) \geq \psi_r(x,y,d) - \sqrt{x}.$$

And by lemma 4.8,

$$\psi_r(x,y,d) \geq \binom{n+k}{k}$$

with $k := \pi(x) - \pi(z) - \omega_y(d)$, $n := \lfloor \log(x)/\log(y) \rfloor$, so that

$$\log \psi_r(x, y, d) \geq \log \binom{n+k}{k} = (n+k)\log(n+k) - k\log(k) - n\log(n) \\ + \frac{1}{2}\log(n+k) - \frac{1}{2}\log(k) - \frac{1}{2}\log(n) + O(1). \qquad (10)$$

We have $\pi(t) = t/\log(t) + O(t/\log(t)^2)$ as $t \longrightarrow +\infty$ and

$$\omega_y(d) = O(\log(d)) = O(\log(x)) = o(y/\log^2(y)),$$

since $y \gg \log^\alpha(x)$ for all $\alpha > 0$, because $\log(y) \gg \log\log(x)$. It follows that

$$k = \pi(x) - \pi(z) - \omega_y(d) = \frac{(1-\varepsilon)y}{\log(y)} + O\left(\frac{y}{\log(y)^2}\right).$$

Besides, we have $\log(1-\varepsilon) \ll \log\log(y)$, so

$$\log((1-\varepsilon)\log(y)) = \log(1-\varepsilon) + \log\log(y) \sim \log\log(y)$$

so $\log((1-\varepsilon)\log(y)) \longrightarrow +\infty$ and $(1-\varepsilon)\log(y) \longrightarrow +\infty$ i.e. $1-\varepsilon \gg 1/\log(y)$. It follows that

$$k \sim \frac{(1-\varepsilon)y}{\log(y)}.$$

Furthermore, $\log(y) \gg \log\log(x)$ so

$$\log\left(\frac{y}{\log^2(x)}\right) = \log(y) - 2\log\log(x) \sim \log(y)$$

so $y/\log^2(x) \longrightarrow +\infty$ and $y \gg \log^2(x)$. It follows that

$$\frac{n^2}{k} \sim \frac{\log^2(x)}{(1-\varepsilon)y\log(y)} = o\left(\frac{\log^2(x)}{y}\right) = o(1),$$

so 10 becomes

$$\begin{aligned}\log \psi_r(x, y, d) &\geq n\log\left(\frac{k}{n}\right) + n - \frac{1}{2}\log(n) + O(1) \\ &= \frac{\log(x)}{\log(y)}\log\left(\frac{(1-\varepsilon)y}{\log(x)}\right) + \frac{\log(x)}{\log(y)} - \frac{1}{2}\log\left(\frac{\log(x)}{\log(y)}\right) + O(1) \\ &= \log(x) - \frac{\log(x)\log\log(x)}{\log(y)} + o\left(\frac{\log(x)\log\log(y)}{\log(y)}\right) \\ &\quad (\text{since } \log(1-\varepsilon) \ll \log\log(y)) \\ &= \log(x) - \frac{\log(x)\log\log(x)}{\log(y)}(1+o(1)).\end{aligned}$$

It follows that

$$\begin{aligned}\frac{\psi_r(x, y, d)}{\sqrt{x}} &\geq \exp\left(\frac{1}{2}\log(x) - \frac{\log(x)\log\log(x)}{\log(y)}(1+o(1))\right) \\ &= \exp\left(\frac{1}{2}\log(x)(1+o(1))\right) \longrightarrow +\infty,\end{aligned}$$

since $\log(y) \gg \log\log(x)$. Finally, we have

$$\log\left(\frac{\psi_r^*(x,y,d)}{x}\right) = \log\left(\frac{\psi_r(x,y,d)}{x}\right) + \log\left(1 - \frac{\sqrt{x}}{\psi_r(x,y,d)}\right)$$
$$\geq -\frac{\log(x)\log\log(x)}{\log(y)}(1+o(1)) + o(1).$$

Besides, $\psi_r^*(x,y,d) \leq \psi(x,y)$, so we conclude by Lemma 4.9. □

Proposition 4.11 *Let $\Delta := |\Delta_{\mathfrak{O}}|$ and $\varepsilon := \Delta^{1/r_m}/B$. We assume that B is subexponential in $\log(\Delta)$, $\varepsilon < 1$ and $\log(1-\varepsilon) \ll \log\log(B)$. Then Algorithm 4.1 has expected time complexity (in bit operations)*

$$T_{FS}(\Delta, B, r_m) = \exp\left((1+o(1))\frac{\log(\Delta)\log\log(\Delta)}{\log(B)}\right.$$
$$\left. + (\sqrt{2}+o(1))\sqrt{\log(B)\log\log(B)}\right),$$

assuming the ECM method has the complexity conjectured in [38, Conjecture 2.10].

Proof By Heuristic 4.7 (since $\varepsilon < 1$ i.e. $r_m \geq \log(\Delta)/\log(B)$), the probability to find an ns-(B, r_m, Δ)-smooth value of $N(a+\omega)$ stisfies

$$\mathbb{P}(B, r_m, \Delta) \geq C \cdot \frac{\psi_{r_m}^*(N(\omega), B, \Delta)}{N(\omega)},$$

where $C > 0$ is a constant. Since $N(\omega) = (\Delta + t^2)/2$ with $t := \mathrm{Tr}(\omega) = \Delta \mod 2$ and B is subexponential in $\log(\Delta)$, we have $\Delta = O(N(\omega))$, $N(\omega) = O(\Delta)$, $\log(B) \ll \log(N(\omega))$ and $\log(B) \gg \log\log(N(\omega))$. We also have $r_m \geq \log(\Delta)/\log(B)$ and $\log(1-\varepsilon) \ll \log\log(B)$, so we may apply Lemma 4.10:

$$\log\left(\frac{\psi_{r_m}^*(N(\omega), B, \Delta)}{N(\omega)}\right) \sim -\frac{\log(N(\omega))\log\log(N(\omega))}{\log(B)}$$
$$\sim -\frac{\log(\Delta)\log\log(\Delta)}{\log(B)}(1+o(1)).$$

Hence, taking into account the ECM method complexity, Algorithm 4.1 terminates in time

$$T_{FS}(\Delta, B, r_m) = \frac{L_B(1/2, \sqrt{2})\,\mathrm{polylog}(\Delta)}{\mathbb{P}(B, r_m, \Delta)} = \exp\left((1+o(1))\frac{\log(\Delta)\log\log(\Delta)}{\log(B)}\right.$$
$$\left. + (\sqrt{2}+o(1))\sqrt{\log(B)\log\log(B)}\right).$$

□

4.2.2 Complexity of the tree filling algorithm (Algorithm 4.2)

Proposition 4.12 *With inputs $B > 0$, an imaginary quadratic order $\mathfrak{O}$ with $|\Delta_{\mathfrak{O}}|B < p$, primes $\ell_1, \cdots, \ell_s \leq B$ splitting in $\mathfrak{O}$ and an oracle* IsOrientable$_{\mathfrak{O}}$ *for Problem 2.3 running in constant time, Algorithm 4.2 runs in time*

$$O\left(2^s B^2\, \mathrm{polylog}(B)\log(p)M(p)\right),$$

where $M(p)$ is the time complexity of operations over $\mathbb{F}_p$. It also uses $O(2^s \log(p))$ bits of memory.

Proof Filling-in tree $\mathcal{T}$ in Algorithm 4.2 costs for all $1 \leq i \leq s$, 2^{i-1} calls to $\mathsf{IsOrientable}_{\mathfrak{O}}$ and the computation of 2^{i-1} sets of j-invariants ℓ_i-isogenous to the same elliptic curve. Each call to $\mathsf{IsOrientable}_{\mathfrak{O}}$ costs $O(1)$ and each j-invariants computation costs $O(\ell_i^2 \operatorname{polylog}(\ell_i)\log(p))$ operations over $\mathbb{F}_{p^2}$ by Sect. 2.4. Arithmetic operations over $\mathbb{F}_{p^2}$ cost $O(M(p))$. Hence, the total cost of filling tree $\mathcal{T}$ is

$$\begin{aligned} T_{tree}(s, B, p) &= \sum_{i=1}^{s} 2^{i-1} O\left(\ell_i^2 \operatorname{polylog}(\ell_i)\log(p)M(p)\right) \\ &= \sum_{i=1}^{s} 2^{i-1} O(B^2 \operatorname{polylog}(B)\log(p)M(p)) \\ &= O\left(2^s B^2 \operatorname{polylog}(B)\log(p)M(p)\right). \end{aligned}$$

The memory used by Algorithm 4.2 is the size of tree $\mathcal{T}$, which contains $\sum_{i=1}^{s} 2^{i-1} = 2^s - 1$ j-invariants defined over $\mathbb{F}_{p^2}$. Each j-invariant takes $2\log(p)$ bits to store, so the algorithm uses $O(2^s \log(p))$ bits of memory. □

4.2.3 Complexity of the search to decision reduction algorithm (Algorithm 4.3)

Theorem 4.13 *Let $\Delta := |\Delta_{\mathfrak{O}}|$. Assume Heuristic 4.7 and that the ECM method has the complexity conjectured in [38, Conjecture 2.10]. Then, with smoothness parameters*

$$B := L_\Delta\left(\frac{1}{2}, \frac{\sqrt{2}}{2}\right), \quad r_m := \left\lceil \sqrt{\frac{2\log(\Delta)}{\log\log(\Delta)}} \right\rceil + 1 \quad \text{and} \quad D := O(\log(p))$$

and provided $B\Delta < p$, Algorithm 4.3 terminates in time

$$L_\Delta(1/2, \sqrt{2})\log(p)M(p).$$

It also requires

$$O\left(2^{\sqrt{2\log(\Delta)/\log\log(\Delta)}}\log(p)\right)$$

bits of memory.

Proof We already have proved the termination of Algorithm 4.3 when $B\Delta < p$. This is a consequence of Lemma 4.3, Lemma 4.4 and Heuristic 4.7 (which prove that $\mathsf{TreeFill}$ and $\mathsf{FindSmoothGen}$ terminate).

On the whole, the total time complexity (in bit operations) of Algorithm 4.3 is

$$T(B, \Delta, r_m, p) = T_{FS} + 2T_{tree} + T_{iso} + T_{trace} + T_{rep},$$

where:

- T_{FS} is the execution time of $\mathsf{FindSmoothGen}$ (Algorithm 4.1), given by Proposition 4.11:

$$\begin{aligned} T_{FS}(\Delta, B, r_m) = \exp\Bigg(&(1 + o(1))\frac{\log(\Delta)\log\log(\Delta)}{\log(B)} \\ &+ (\sqrt{2} + o(1))\sqrt{\log(B)\log\log(B)}\Bigg). \end{aligned}$$

- T_{tree} is the execution time of TreeFill (Algorithm 4.2), given by Proposition 4.12:
$$T_{tree}(B, s, p) = O\left(2^s B^2 \operatorname{polylog}(B)\log(p)M(p)\right).$$
with $s = r_m/2 + O(1)$.
- T_{iso} is the time taken in lines 9 and 10 of Algorithm 4.3 to recover the chain of ℓ_i-isogenies $\psi_i : E_{i-1} \longrightarrow E_i$, given the sequence of j-invariants $j(E_0) = j(E), j(E_1), \cdots, j(E_r) = j(E)$. By Sect. 2.5, recovering an ℓ_i-isogeny from the j-invariants of its domain and codomain costs $O(\ell_i^2 \operatorname{polylog}(\ell_i)\log(p))$ operations over $\mathbb{F}_{p^2}$. Hence, we have
$$T_{iso} = O(r_m B^2 \operatorname{polylog}(B)\log(p)M(p))$$
- T_{trace} is the time needed to compute the trace of $\psi = \psi_r \circ \cdots \circ \psi_1$. We use Schoof's algorithm [58, Sect. 5]. Namely, we look for primes $p_1, \cdots, p_t$ such that $\prod_{i=1}^t p_i > 4\sqrt{\deg(\psi)}$ and evaluate ψ on $E[p_i]$ to find $\tau_i \in \mathbb{Z}/p_i\mathbb{Z}$ such that $\psi^2 - [\tau_i]\psi + [\deg(\psi)]$ is zero on $E[p_i]$ and recover $\operatorname{Tr}(\psi)$ by solving $\operatorname{Tr}(\psi) \equiv \tau_i \mod p_i$ for all $i \in \{1, \cdots, t\}$ via Chinese remainder theorem. Since $\deg(\psi) = N(\theta) \leq \Delta$, we can choose $t = O(\log(\Delta))$ and $p_i = O(\log(\Delta))$. Hence, the dominant cost is the evaluation via ψ of $O(\log(\Delta))$ points all defined over an extension of degree $O(\log(\Delta))$ of $\mathbb{F}_{p^2}$ (by Lemma 2.12). This cost amounts to
$$T_{trace}(B, r_m, \Delta, p) = O(r_m B \log^3(\Delta)M(p)).$$
- T_{rep} is the running time of EfficientRep (Algorithm 2.1). Since $\deg([\epsilon]\circ\psi - [a]) = N(\omega) \leq (\Delta+1)/4$, we can find a D-powersmooth number coprime with $\deg([\epsilon]\circ\psi - [a])$ when $D = O(\log(\Delta))$ (line 1 of Algorithm 2.1). Hence, by Proposition 2.13, the dominant cost of the call to EfficientRep is given by $O(\log(\Delta))$ evaluations of ψ on points defined over an extension of degree $O(\log(\Delta))$ of $\mathbb{F}_p$, which amounts to
$$T_{rep}(\Delta, B, r_m, p) = O(r_m B \log^3(\Delta)M(p)).$$

It follows that:
$$\begin{aligned} T(B, r_m, \Delta, p) &= T_{FS} + 2T_{tree} + T_{iso} + T_{trace} + T_{rep} \\ &= \exp\left((1+o(1))\frac{\log(\Delta)\log\log(\Delta)}{\log(B)}\right. \\ &\quad \left. + (\sqrt{2}+o(1))\sqrt{\log(B)\log\log(B)}\right) \\ &\quad + M(p)\log(p)\exp\left(\frac{\log(2)r_m}{2} + 2\log(B)\right) \end{aligned}$$

But by Proposition 4.11, we have $r_m = \log(\Delta)/\log(B\varepsilon)$ with $\log(1-\varepsilon) \ll \log\log(B)$. We can impose that $\varepsilon \longrightarrow 0$, so that $\log(1-\varepsilon) \ll \log\log(B)$ and that $\log(\varepsilon) \ll \log(B)$, so that $r_m \sim \log(\Delta)/\log(B)$. Heuristically, the quantity $T(\Delta, B, r_m, p)$ is minimal when the arguments of the two exponentials are close, i.e. when
$$\frac{\log(\Delta)\log\log(\Delta)}{\log(B)} \simeq 2\log(B),$$
the other terms being negligible. Hence, we choose
$$B = \exp\left(\frac{\sqrt{2}}{2}\sqrt{\log(\Delta)\log\log(\Delta)}\right) = L_\Delta\left(\frac{1}{2}, \frac{\sqrt{2}}{2}\right),$$

so that

$$T(B, r_m, \Delta, p) = M(p)\log(p)L_\Delta\left(\frac{1}{2}, \sqrt{2}\right).$$

and

$$\begin{aligned} r_m &= \sqrt{\frac{2\log(\Delta)}{\log\log(\Delta)}}\left(1 + \frac{\sqrt{2}\log(\varepsilon)}{\sqrt{\log(\Delta)\log\log(\Delta)}}\right)^{-1} \\ &= \sqrt{\frac{2\log(\Delta)}{\log\log(\Delta)}} - \frac{2\log(\varepsilon)}{\log\log(\Delta)}. \end{aligned}$$

Hence, we can set $r_m := \lceil\sqrt{2\log(\Delta)/\log\log(\Delta)}\rceil + 1$, so that $\log(\varepsilon) = O(\log\log(\Delta)) = o(\log(B))$.

The space complexity is dominated by the trees $\mathcal{T}_1$ and $\mathcal{T}_2$, so Algorithm 4.3 uses

$$O(2^{r_m/2}\log(p)) = O\left(2^{\sqrt{2\log(\Delta)/\log\log(\Delta)}}\log(p)\right)$$

bits of memory by Proposition 4.12. □

Corollary 4.14 *Given an imaginary quadratic order $\mathfrak{O}$ of discriminant $\Delta_{\mathfrak{O}}$ and a prime $p > L_{|\Delta_{\mathfrak{O}}|}(1/2, \sqrt{2}/2)|\Delta_{\mathfrak{O}}|$, then, over $\mathbb{F}_{p^2}$ the $\mathfrak{O}$-orienting Problem (Problem 2.4) reduces to the Decisional $\mathfrak{O}$-orienting Problem (Problem 2.3) in time*

$$L_{|\Delta_{\mathfrak{O}}|}(1/2, \sqrt{2})\log(p)M(p),$$

using

$$O\left(2^{\sqrt{2\log(|\Delta_{\mathfrak{O}}|)/\log\log(|\Delta_{\mathfrak{O}}|)}}\log(p)\right)$$

bits of memory.

5 $\mathfrak{O}$-orienting problem for quaternion orders

Isogeny problems can often be translated to quaternion problems via the Deuring correspondence, and in many cases, the quaternion problems are easier to solve. In this section we consider the quaternion analogue of the $\mathfrak{O}$-Orienting Problem as it was stated earlier:

Problem 2.6 (Quaternion Order Embedding Problem) *Given a maximal quaternion order $\mathcal{O} \subset B_{p,\infty}$ and an imaginary quadratic order $\mathfrak{O}$ which embeds into $\mathcal{O}$, find the embedding.*

Similarly to the curve setting, we define a primitive $\mathfrak{O}$-embedding of $\mathcal{O}$ to be an embedding $\iota : \mathfrak{O} \hookrightarrow \mathcal{O}$ which cannot be extended to a superorder of $\mathfrak{O}$, also known as an optimal embedding [62, Chapter 30]. We also address this problem for primitive embeddings.

In this section, we present a general algorithm and analyse its complexity, noting special cases. For complexity analysis we assume an efficient factorization oracle exists, however, we provide a practical alternative for running the algorithm without such an oracle. For embedding small discriminant quadratic orders $\mathfrak{O}$, our algorithm improves the state of the art by being efficient up to $\mathrm{disc}(\mathfrak{O}) = O(p)$.

Before moving on to the actual algorithms we give a brief technical overview of the main idea. First, we compute a short prime norm N ($\approx \sqrt{p}$) connecting ideal between a quaternion order $\mathcal{O}'$ isomorphic to $\mathcal{O}$ and a special extremal order. Our goal is to compute an element of prescribed trace and norm in $\mathcal{O}'$ and then one can easily construct an element with said trace and norm in $\mathcal{O}$ as well. For simplicity assume that the prescribed trace is 0. The trace 0 part of $\mathcal{O}'$ is a rank 3 lattice and one can compute the Hermite Normal Form (HNF) of this lattice. This means that one has a basis of the form $e_{11}i + e_{12}j + e_{13}k, e_{22}j + e_{23}k, e_{33}k$ and even though e_{ij} are not likely to be integers, their denominator is a divisor of $2N$. When looking for an element of trace 0 and norm smaller than p the coefficients of this element with respect to this HNF basis will have a very specific structure. Namely, the coefficient of $e_{11}i + e_{12}j + e_{13}k$ will be smaller than p in absolute value and thus can be easily determined by looking at the norm modulo p. Then one only has to work out the two other coefficients which is equivalent to solving a binary quadratic form where the quadratic part is positive definite. This can then essentially be reduced to Cornacchia's algorithm [57]. We can extend this to filter out imprimitive solutions.

5.1 Finding general embeddings

First, we present an algorithm for finding embeddings, and in the next section, we use this to define primitive embeddings. Suppose we are given a maximal quaternion order $\mathcal{O} \subset B_{p,\infty}$ in terms of a $\mathbb{Z}$-basis, and an imaginary quadratic order $\mathfrak{O} = \mathbb{Z}[\omega]$, by generator ω of reduced trace t and reduced norm d.

We start with an observation: suppose an embedding $\iota : \mathbb{Z}[\omega] \hookrightarrow \mathcal{O}$ exists and let $\alpha = \iota(\omega)$. Since $\omega^2 - t\omega + d = 0$ we must also have $\alpha^2 - t\alpha + d = 0$. Hence α also has trace t and norm d. Finding any element α of norm d and trace t is enough to define an embedding ι, solving Problem 2.6. This is the approach we take in Algorithm 5.1, finding $\alpha \in \mathcal{O}$ of a given norm and trace. We make the assumption $p \neq 2$ and conventionally use $1, i, j, k$ as a basis of $B_{p,\infty}$ with $i^2 = -q$ and $j^2 = -p$. If $p \equiv 3 \pmod 4$ we take $q = 1$. If $p \equiv 5 \pmod 8$ we take $q = 2$. If $p \equiv 1 \pmod 8$ we take q to be a prime $q \equiv 3 \pmod 4$ such that p is not a quadratic residue modulo q. While $p \equiv 3 \mod 4$ is the most relevant for isogeny-based cryptography, we consider general p. We fix a maximal order $\mathcal{O}_0$ in the following way:

Proposition 5.1 [48, Proposition 5.2] *The following definitions give a maximal order in $B_{p,\infty}$ for any $p \neq 2$:*

$$\mathcal{O}_0 = \begin{cases} \mathbb{Z}[\frac{1+j}{2}, \frac{i+k}{2}, j, k] & \text{if } p \equiv 3 \mod 4 \\ \mathbb{Z}[\frac{1+j+k}{2}, \frac{i+2j+k}{4}, j, k] & \text{if } p \equiv 5 \mod 8 \\ \mathbb{Z}[\frac{1+i}{2}, \frac{i+ck}{q}, \frac{j+k}{2}, k] & \text{if } p \equiv 1 \mod 8 \end{cases}$$

where c is an integer such that q divides $c^2p + 1$ where q and c exist by [21, Proposition 1].

Our algorithm will work with the basis of $\mathcal{O}$ in column-style Hermite normal form (HNF). We denote the basis vectors e_0, e_1, e_2, e_3. Then we can write $\mathcal{O}$ as:

$$\begin{aligned} \mathcal{O} = \langle & e_{00} + e_{01}i + e_{02}j + e_{03}k, \\ & e_{11}i + e_{12}j + e_{13}k, \\ & e_{22}j + e_{23}k, \\ & e_{33}k\rangle_{\mathbb{Z}} \end{aligned} \tag{11}$$

with coefficients $e_{mn} \in \mathbb{Q}$. For example, see the orders in Proposition 5.1 above. We know the basis is full rank, so $e_{nn} \neq 0$ for $n = 0, 1, 2, 3$, and we prove some additional properties:

Lemma 5.2 *Given the basis* (e_{mn}) *of a maximal order* $\mathcal{O} \subset B_{p,\infty}$ *in column-style HNF as above, the following properties hold:*

(1) $e_{mn} \geq 0$ *for all* n, m
(2) *The denominators of each rational entry* e_{mn}*, when expressed in simplest form, divide* $K \cdot N(I)$ *where* $K = 2, 4$ *or* $2q$ *depending on whether* $p \equiv 3 \mod 4$*, or* $\equiv 5 \mod 8$ *or* $\equiv 1 \mod 8$ *respectively*
(3) $e_{00} = \frac{1}{2}$
(4) $e_{22}e_{33} \leq N(I)$
(5) $e_{01} = 0$ *or* $e_{01} = 1/(2Ke_{22}e_{33})$ *where* K *is defined in (2)*

where $I = I(\mathcal{O}_0, \mathcal{O}) := N\mathcal{O}_0\mathcal{O}$ *is the connecting left-ideal from* $\mathcal{O}_0$ *to* $\mathcal{O}$ *and* $N := [\mathcal{O} : \mathcal{O}_0 \cap \mathcal{O}]$.

Proof We can prove the statements as follows:

(1) Requirement of HNF.
(2) As defined, I is the connecting ideal between $\mathcal{O}_0$ and $\mathcal{O}$. The ideal I is contained in both $\mathcal{O}_0$ and $\mathcal{O}$ and $N(I) \cdot \mathcal{O} = I\bar{I} \subseteq \mathcal{O}_0$ [62, Proposition 16.6.15]. Therefore the largest denominator of all e_{mn}s is at most $N(I)$ times the largest denominator of $\mathcal{O}_0$ as given in Proposition 5.1.
(3) The trace of any element must be integral hence $2e_{00} \in \mathbb{Z}$. We must also have $1 \in \mathcal{O}$ hence 1 can be written as a linear combinations of the basis (e_i), where where taking the trace gives $k_0e_{00} = 1$ for some $k_0 \in \mathbb{Z}$. This implies either $e_{00} = \frac{1}{2}$ or 1 and $\mathrm{Tr}(\mathcal{O}) = \mathbb{Z}$ or $2\mathbb{Z}$ respectively. The (non-reduced) discriminant of any maximal order in $B_{p,\infty}$ is p^2, so by definition $p^2 = \det(\mathrm{Tr}(e_me_n)) \in \mathrm{Tr}(\mathcal{O})$, but p is odd, so $p^2 \notin 2\mathbb{Z}$ and we must have $e_{00} = \frac{1}{2}$.
(4) As $\mathcal{O}$ and $\mathcal{O}_0$ are maximal, they both have the same discriminant. Hence the change of basis matrix must have determinant 1 [62, Lemma 15.2.5], which means $\prod e_{nn} = \prod f_{nn} = \frac{1}{2 \cdot K}$, where $(f)_n$ is the basis of $\mathcal{O}_0$ specified in Proposition 5.1. Then using (3) we have $e_{11} = 1/(Ke_{22}e_{33})$. The result follows from (2).
(5) $1 \in \mathcal{O}$ so there is some $n \in \mathbb{Z}$ such that $\frac{1}{e_{00}}e_{01} - ne_{11} = 0$. From above $e_{00} = \frac{1}{2}$, and $e_{11} = \frac{1}{Ke_{22}e_{33}}$ so $2e_{01} = \frac{n}{Ke_{22}e_{33}}$. But to be in HNF we must have already reduced e_{01} as much as possible hence $n = 0$ or 1.

□

Further, we will also use the following lemma to bound the denominators (taken from [16, Lemma 5.2.2]):

Lemma 5.3 *Let* $\mathcal{O} \subset B_{p,\infty}$ *be a maximal order with connecting ideal* $I = I(\mathcal{O}_0, \mathcal{O})$*, then there exists an equivalent ideal* $J \sim I$ *with* $N(J) \leq \frac{2\sqrt{2}}{\pi}\sqrt{p}$

We now describe the algorithm. We address arbitrary trace in Remark 5.4 and Algorithm 5.1 has no restrictions on the trace. However, for simplicity, we first describe the algorithm under the assumption that the trace of ω is zero.

Step 1 : **Compute HNF** Put the basis of $\mathcal{O}$ into column-style Hermite normal form (HNF). In general, we can replace the order $\mathcal{O}$ by an isomorphic order $\mathcal{O}'$, having denominator bounded by $N := K \cdot N(I') = O(\sqrt{p})$, where I' is an ideal equivalent to the connecting $(\mathcal{O}_0, \mathcal{O})$-ideal I, and where K is defined in (2) of Proposition 5.2, by taking the ideal from Lemma 5.3. We return to this in Sect. 5.3, but for now, by passing to the isomorphic order "closest" to $\mathcal{O}_0$, we assume that N is of size $O(\sqrt{p})$

Step 2 : **Fix trace** To find a trace zero element α of norm d, we may write an arbitrary element in the following form:

$$\alpha = \alpha_0 e_0 + \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3.$$

Note that since we are working in Hermite Normal Form only e_0 contributes to the trace of α so we set $\alpha_0 = 0$ to get $\mathrm{Tr}(\alpha) = 0$.
For the condition on the norm, consider the case $p \equiv 3 \mod 4$ for simplicity. However, note that this generalizes for any prime $p \neq 2$. Then we have the rational ternary quadratic form:

$$(\alpha_1 e_{11})^2 + p(\alpha_1 e_{12} + \alpha_2 e_{22})^2 + p(\alpha_1 e_{13} + \alpha_2 e_{23} + \alpha_3 e_{33})^2 = \mathrm{nrd}(\alpha) = d.$$

Step 3 : **Find $\boldsymbol{\alpha_1} \mod \boldsymbol{p}$** Since α_1 controls the coefficient of i it is the only term without a factor of p. Hence working modulo p removes terms containing α_2 and α_3, and we can find $\alpha_1 \equiv r \mod p$.

$$r_{\pm} := \frac{\pm\sqrt{d}}{e_{11}} \mod p$$

Fix the least positive residue class representative $r = r_+$, as we can execute the remainder of the algorithm a second time on r_- if necessary. Then substitute $\alpha_1 = r + kp$ giving a rational ternary quadratic form in k, α_2 and α_3.

Step 4 : **A binary quadratic form** As defined in Step 1, we may multiply by the denominator N^2 to obtain integral coefficients. Rearranging we have:

$$pN^2(\gamma_1^2 + \gamma_2^2) = N^2(d - \alpha_1^2 e_{11}^2)$$

where

$$\gamma_1 = \alpha_1 e_{12} + \alpha_2 e_{22}, \qquad \gamma_2 = \alpha_1 e_{13} + \alpha_2 e_{23} + \alpha_3 e_{33}.$$

Let $v := N^2(\gamma_1^2 + \gamma_2^2)$ and notice $v \geq 0$. From the right-hand side above we see its value depends on k.

$$v = \frac{N^2(d - (r + kp)^2 e_{11}^2)}{p}$$

Clearly, v decreases as k increases. Without loss of generality, we can assume $k \geq 0$, and since $v \geq 0$ we get an upper bound on k. We can iterate over this range of k which is precisely

$$k = 0, ..., \left\lfloor \frac{\sqrt{d}}{pe_{11}} - \frac{r}{p} \right\rfloor$$

where for each iteration, we compute v using the above equation, and with k fixed are left with the integral binary quadratic form $v = N^2(\gamma_1^2 + \gamma_2^2)$.

Step 5 : **Cornacchia's Algorithm** Writing the above form as $\beta_1^2 + \beta_2^2 = v$ we solve for integral pairs (β_1, β_2) using Cornacchia's algorithm. For a valid solution we can write it in the form:

$$\beta_1 = N\gamma_1 = N\alpha_1 e_{12} + N\alpha_2 e_{22}$$
$$\beta_2 = N\gamma_2 = N\alpha_1 e_{13} + N\alpha_2 e_{23} + N\alpha_3 e_{33}$$

and solve for α_2 and α_3

$$\alpha_2 = \frac{\beta_1 - N\alpha_1 e_{12}}{Ne_{22}}, \qquad \alpha_3 = \frac{\beta_2 - N\alpha_1 e_{13} - N\alpha_2 e_{23}}{Ne_{33}}.$$

Finally, we must check $\alpha_2, \alpha_3 \in \mathbb{Z}$. If this is true we have a valid solution $\alpha = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3$. If not we continue trying the next solution to Cornacchia's, or move on to the next iteration of k in Step 4. If no solutions are found it means $\mathbb{Z}[\omega]$ does not embed into $\mathcal{O}$.

Remark 5.4 (Arbitrary trace t) Suppose the element we are searching for does not have trace zero. We can always reduce the problem to finding an element of trace zero. Suppose $t \in 2\mathbb{Z}$, then since $\mathcal{O}$ is a ring we have $1 \in \mathcal{O}$ so $\alpha - t/2 \in \mathcal{O}$ has trace zero and norm $d - t^2/4 \in \mathbb{Z}$. We can search for this trace zero element and then use this to find α. Similarly if t is odd, we have a trace zero element $2\alpha - t \in \mathcal{O}$ of norm $4d - t^2$. Once this is found, we translate back, divide by 2 and check if $\alpha \in \mathcal{O}$ in Step 5 of the algorithm. If not, we continue searching.

Note for t odd, this is not optimal as the scaling increases d by a factor of 4, and hence the number of iterations of k by a factor of 2, which asymptotically makes no difference, but in practice, can double the running time. Instead we can avoid this by incorporating additional constant terms for the non-zero trace. These details are included in Algorithm 5.1, which we use for our implementation.

The complete algorithm for arbitrary trace is summarised in Algorithm 5.1. Additionally, we describe a few further generalisations and improvements:

Remark 5.5 Algorithm 5.1...

- results in an embedding, but this does not necessarily define a primitive $\mathbb{Z}[\omega]$-embedding. This is discussed in Sect. 5.5.
- can be adapted to work with any prime $p \neq 2$, not specifically $p \equiv 3 \mod 4$. For general, $B_{p,\infty} = \left(\frac{-q,-p}{\mathbb{Q}}\right)$, q appears in the equations for r, v and the maximum k, and the binary quadratic form to solve is $\beta_1^2 + q\beta_2^2 = v$ instead of the sum of two squares. Cornacchia's still works since for $B_{p,\infty}$, q and p are always coprime.
- can be adapted to non-maximal orders. The value N gains the index of the order within a maximal order as a factor.
- is more efficient iterating from largest k to smallest, as this minimizes the values of v used in Cornacchia's.
- can be improved by using a congruence condition to rule out some cases where Cornacchia's does not have any solutions, before executing Cornacchia's. In the case $p \equiv 3 \mod 4$, we test for solutions by noting v can be written as the sum of two squares if and only if, in its prime factorization, every prime which is 3 mod 4 occurs an even number of times. For arbitrary p, a similar necessary but not sufficient congruence condition can test the splitting of v to rule out some cases.

5.2 Complexity analysis of Algorithm 5.1

In this section we give results on the asymptotic complexity of Algorithm 5.1, in particular giving average-case results and a probabilistic worst-case result. We start by giving a worst-case running time. Note that there are three reasons why Cornacchia's algorithm may not be efficient at finding all solutions to $\beta_1^2 + q\beta_2^2 = v$:

Algorithm 5.1: Algorithm to find embeddings of quadratic order in quaternion order, for $B_{p,\infty}$, $p \neq 2$.

```
Data: Maximal order $\mathcal{O} \subset B_{p,\infty}$, given in terms of basis $e_0, e_1, e_2, e_3$. Quadratic order in the form
      $\mathbb{Z}[\omega]$ given by $\omega$.
Result: Returns element $\alpha \in \mathcal{O}$, which defines an embedding $\iota : \mathbb{Z}[\omega] \hookrightarrow \mathcal{O}$ by $\omega \mapsto \alpha$. Or returns $\perp$
        if no element $\alpha$ exists.
1  Compute $d = \mathrm{nrd}(\omega)$ and $t = \mathrm{Tr}(\omega) \in \mathbb{Q}$;
2  Compute Hermite normal form of order, giving basis $e_0, e_1, e_2, e_3$ in form of Equation (11). Denote
   coefficient $n$ of vector $m$ as $e_{mn}$;
3  Compute $\alpha_0 := \frac{t}{2e_{00}}$;
4  if $d, \alpha_0 \notin \mathbb{Z}$ then
5      Return $\perp$;
6  end
7  Compute $r_\pm := \frac{1}{e_{11}}\left(\pm\sqrt{d - (\alpha_0 e_{00})^2} - \alpha_0 e_{01}\right) \mod p$;
8  Set $r = r_+$;
9  Compute $N := lcm(\{\mathrm{Denom}(e_{mn}) : 0 \leq m \leq 3, m \leq n \leq 3\})$ where $\mathrm{Denom}(n)$ denotes the smallest
   denominator of $n \in \mathbb{Q}$;
10 for $k = \left\lfloor \frac{1}{pe_{11}}\left(\sqrt{d - (\alpha_0 e_{00})^2} - \alpha_0 e_{01} - re_{11}\right)\right\rfloor$ decreasing to 0 do
11     Compute $v = \frac{N^2(d-(\alpha_0 e_{00})^2-(\alpha_0 e_{01}+(r+kp)e_{11})^2)}{p}$;
12     Run Cornacchia's algorithm to find all solutions $\beta_1^2 + \beta_2^2 = v$. Store solutions in array $C$;
13     for $(\beta_1, \beta_2)$ in $C$ do
14         Set $\alpha_2 = \frac{\beta_1 - N\alpha_0 e_{02} - N\alpha_1 e_{12}}{Ne_{22}}$;
15         Set $\alpha_3 = \frac{\beta_2 - N\alpha_0 e_{03} - N\alpha_1 e_{13} - N\alpha_2 e_{23}}{Ne_{33}}$;
16         if $\alpha_2 \in \mathbb{Z}$ and $\alpha_3 \in \mathbb{Z}$ then
17             Return $\alpha = \alpha_0 e_0 + \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3$;
18         end
19     end
20 end
21 Repeat from line 8 with $r = r_-$;
22 Return $\perp$;
```

(1) It requires a factorization of v. To this end, we assume we have an efficient factorization oracle such as Shor's algorithm. See Sect. 5.3 later on for a practical alternative to using a factorization oracle.

(2) Cornacchia's algorithm typically only refers to finding primitive solutions where $gcd(\beta_1, \beta_2) = 1$. To also find imprimitive solutions we must run Cornacchia's on $\beta_1^2 + q\beta_2^2 = v/g^2$ for every square $g^2 \mid v$ and scale up the solutions $(g\beta_1, g\beta_2)$. The number of squares dividing v can be subexponential in v. However, we can say the probability of this for random v is very small, in fact asymptotically there is $\frac{\pi^2}{6} \sim 61\%$ chance v is square-free.

(3) While just solving for primitive solutions, we must iterate over all the solutions Cornacchia gives. Internally Cornacchia must iterate over all solutions x to the equation $x^2 \equiv -q \mod v$, where the number of solutions can be exponential in v if v has a large number of distinct prime factors. For example, experimentally with $p \equiv 3 \mod 4$ and $d \sim p$ we get some integers $v \sim p$ where if v has lots of distinct prime factors, there can be as many as $v^{0.15} \sim p^{0.15}$ solutions which is exponential. We resolve this issue by bounding the number of factors of v by the following probability estimate known as the fundamental theorem of probabilistic number theory:

Lemma 5.6 (Erdős–Kac theorem [24]) *For a positive integer n, the number of distinct prime factors of n follows the standard normal distribution with mean* $\log\log n$ *and standard deviation* $\sqrt{\log\log n}$ *as* $n \to \infty$.

This gives us the following result:

Theorem 5.7 *Let $0.5 \leq P < 1$. Assuming the heuristic that v is distributed like random integers and hence the number of distinct prime factors follows Lemma 5.6, and given an efficient Algorithm 5.1 is within*

$$O\left(T\left(\frac{P+1}{2}\right)\cdot \log(N^2 d)^{\mathcal{F}(\frac{P+1}{2})+1}\left\lceil \frac{N}{p}\sqrt{d-\frac{t^2}{4}}\right\rceil \cdot polylog(X)\right)$$

with probability P. With $N = 2N(I) = O(\sqrt{p})$ (Lemma 5.3) this is

$$O\left(T\left(\frac{P+1}{2}\right)\cdot \log(pd)^{\mathcal{F}(\frac{P+1}{2})+1}\left\lceil \frac{1}{\sqrt{p}}\sqrt{d-\frac{t^2}{4}}\right\rceil \cdot polylog(X)\right)$$

where X is the total size of the inputs, and $T(P)$ is a value large enough such that the asymptotic probability that a random number has less than $T(P)$ perfect square divisors is larger than P. We define $\mathcal{F}$ as the inverse cumulative distribution function of the standard normal distribution where a sample is less than $\mathcal{F}(P)$ with probability P. For example, for $P = 0.95$ we have $\mathcal{F}\left(\frac{P+1}{2}\right) < 2$ and $T\left(\frac{P+1}{2}\right) \sim 4$.

Proof Steps 1–3 of the algorithm are efficient as polynomial time algorithms exist for computing Hermite normal form [30] [14, Chapter VII], and fixing α_0 and solving α_1 modulo p is efficient. In Step 4 a worst-case input will result in iterating k over it's full range of values which is $O(\frac{1}{pe_{11}}\sqrt{d-(\alpha_0 e_{00})^2})$, where the trace is fixed through $\alpha_0 = \frac{t}{2e_{00}}$ so $(\alpha_0 e_{00})^2 = \frac{t^2}{4}$. And by Proposition 5.2 we have $\frac{1}{e_{11}} \leq N$. Then for each iteration over k, Cornacchia's algorithm is used in Step 5. To be efficient at finding primitive solutions we have to bound the number of distinct prime factors of v, by Lemma 5.6 with probability $\frac{P+1}{2}$, v is less than $\mathcal{F}(\frac{P+1}{2})$ standard deviations above the mean,

$$\text{Number of factors of } v \leq \log\log(v) + \mathcal{F}\left(\frac{P+1}{2}\right)\sqrt{\log\log(v)}$$

hence it is certainly true that

$$\text{Number of factors of } v \leq (\mathcal{F}\left(\frac{P+1}{2}\right) + \log\log(v).$$

Then it follows that the number of square roots found in Cornacchia's algorithm is less than $O(2^{(\mathcal{F}(\frac{P+1}{2})+1)\log\log(v)}) = O(\log(v)^{(\mathcal{F}(\frac{P+1}{2})+1)})$, so we can bound the running time of Cornacchia by $O(\log(v)^{(\mathcal{F}(\frac{P+1}{2})+1)}) \cdot polylog(v)$, and clearly for each v we have $v \leq N^2 d < O(pd)$ and hence $polylog(v) = polylog(X)$. The final consideration is for finding imprimitive solutions using Cornacchia's algorithm which requires repeating for every square dividing v. By definition this is at most $T(\frac{P+1}{2})$ repetitions with probability $\frac{P+1}{2}$. The probability both this condition and v having the correct number of factors is at least $\frac{P+1}{2} + \frac{P+1}{2} - 1 = P$. □

Now we give a result for the average-case running time:

Theorem 5.8 *Making the following assumptions, regarding iterating over k:*

- *Each v_k is distributed like random integers and hence the expected number of distinct prime factors is $\log\log(v_k)$ by Lemma 5.6, and there is a high probability it only has a few square divisors.*
- *Additionally, the probability each v_k is the sum of two squares is independent and at least the probability a random integer less than $(Nd)^2$ is the sum of two squares.*
- *The first solution to Cornacchia's algorithm has β_1, β_2 uniformly distributed modulo $e_{22}N$ and $e_{33}N$ respectively.*

Then given an efficient factorization oracle, in the case $p \equiv 3 \mod 4$, the average-case running time of Algorithm 5.1 is $O(\min\{N^3, \lceil \frac{N}{p}\sqrt{d-\frac{t^2}{4}}\rceil\} \times polylog(X))$ and substituting $N = O(\sqrt{p})$ from Lemma 5.3 it is $O(\min\{p\sqrt{p}, \lceil \frac{1}{\sqrt{p}}\sqrt{d-\frac{t^2}{4}}\rceil\} \times polylog(X))$ where X is the total size of all inputs.

We use the following result of Landau [35]:

Lemma 5.9 *The number of integers representable as the sum of two squares from from 0 to $n \in \mathbb{N}$ is the limit $C\frac{n}{\sqrt{\log n}}$ as $n \to \infty$, where $C \approx 0.764$ is the Landau-Ramanujan constant. Hence for sufficiently large n, the number of integers representable is greater than $\frac{1}{2}\frac{n}{\sqrt{\log n}}$. (In fact, experimentally this appears true for all $n \geq 0$).*

Proof of Theorem 5.8 It's clear the running time is the product of the number of iterations over k, and the running time of Cornacchia's, because all other operations are polynomial time. In the case $p \equiv 3 \mod 4$ we are solving the sum of two squares, hence by Lemma 5.9, and using the first assumption, we expect (for sufficiently large d) less than $2\sqrt{\log((Nd)^2)}$ iterations until we find a k where Cornacchia's gives at least one solution.

Now recall that finding one solution to Cornacchia's algorithm is not necessarily enough, since we need to satisfy the conditions $\alpha_2, \alpha_3 \in \mathbb{Z}$. This amounts to checking:

$$\beta_1 - N\alpha_0 e_{02} - N\alpha_1 e_{12} \equiv 0 \mod e_{22}N$$
$$\beta_2 - N\alpha_0 e_{03} - N\alpha_1 e_{13} - N\alpha_2 e_{23} \equiv 0 \mod e_{33}N.$$

Therefore, by the second assumption we expect to have an integral solution after $e_{22}N \times e_{33}N$ solutions from Cornacchia's. Noting that $e_{22}e_{33} \leq N/2$ from Proposition 5.2, that's $N^3/2$ solutions. In total we expect $O(N^3\sqrt{\log(Nd)})$ iterations of k. This is bounded above by the maximum number of iterations from Theorem 5.7. Finally, Cornacchia's algorithm uses the efficient factorization oracle to factorize each v_k and on average v_k is expected to have $\log\log(v_k)$ distinct prime factors by the first assumption, hence internally Cornacchia's computes at most $2^{\log\log(v_k)} = \log(v_k)$ square roots, which is efficient. Then to find imprimitive solutions, we only repeat Cornacchia's a constant number of times as the expected number of squares dividing v is very small. Overall this takes time polylog in each $v_k \leq N^2 d = O(pd)$, so this term can be incorporated into $polylog(X)$. □

From this we observe the following:

Remark 5.10 (Efficient for orders close to $\mathcal{O}_0$) Given an efficient factorization oracle, consider the algorithm applied to the order $\mathcal{O}_0$. Here we have $N = 2$, hence the algorithm is efficient; the average-case running time is $polylog(X)$. For orders close to $\mathcal{O}_0$, such as a curve l-isogenous to the curve with j-invariant 1728, we gain a factor of l in N, hence for small l the algorithm is still efficient. However with each step from $\mathcal{O}_0$, N gains a factor of the degree of the isogeny, so it gets exponentially harder the further you walk, until we reach the point $N \sim \sqrt{p}$.

For completeness, we now consider the case of arbitrary primes $p \neq 2$. Then the quaternion algebra containing order $\mathcal{O}$ is $B_{p,\infty} = \left(\frac{-q,-p}{\mathbb{Q}}\right)$ where q is either 2 or a prime $q \equiv 3 \mod 4$ with Legendre symbol $\left(\frac{q}{p}\right) = -1$.

By the same argument as Theorem 5.7, with high probability P the worst-case running time is within

$$O\left(T(\frac{P+1}{2}) \cdot \log(N^2 d)^{\mathcal{F}(\frac{P+1}{2})+1} \left\lceil \frac{N}{p}\sqrt{\frac{1}{q}(d - \frac{t^2}{4})} \right\rceil \cdot polylog(X)\right)$$

which is the same as before except the additional factor of $\frac{1}{\sqrt{q}}$ appears requiring more iterations over k. Then from Lemma 5.2 part (2), for a different value of K, we get $N = 2qN(I) = O(q\sqrt{p})$. Applying this along with Lemma 5.3 gives:

$$O\left(T(\frac{P+1}{2}) \cdot \log(q^2 pd)^{\mathcal{F}(\frac{P+1}{2})+1} \left\lceil \sqrt{\frac{q}{p}(d - \frac{t^2}{4})} \right\rceil \cdot polylog(X)\right).$$

Typically, q is treated as a constant, as for random primes, q will typically be of negligible size, so asymptotically the complexity is the same. However, q is actually unbounded; you can construct a prime such that the minimum value for q is larger than any given threshold. Hence we treat q as a variable in our analysis.

We have a similar variation on the average time complexity, however, the proof is more complex:

Theorem 5.11 *Given an efficient factorization oracle, for arbitrary $p \neq 2$, making the same assumptions as in Theorem 5.8 (replacing sum of two squares with x^2+qy^2), the average-case running time of Algorithm 5.1 is*

$$O\left(\min\left\{\frac{q^2 p\sqrt{p}}{C(-4q)}, \left\lceil \sqrt{\frac{q}{p}(d - \frac{t^2}{4})} \right\rceil\right\} \cdot polylog(X)\right)$$

where C is a special function generalising the Landau-Ramanujan constant, and X is the total size of all inputs.

Proof The proof is the same as Theorem 5.8, except instead of solving the sum of two squares, we are solving $x^2 + qy^2 = v_k$ using Cornacchia's algorithm. This means in the proof we cannot use Lemma 5.9 on the sum of two squares. However, this result generalises.

Bernays proved that the number of integers between 0 and n represented by a binary quadratic form $f(x, y)$ converges to $C(\Delta_f)\frac{n}{\sqrt{\log n}}$ as $n \to \infty$, where Δ_f is the discriminant of the form f and $C(\Delta_f)$ is a constant depending on Δ_f [5]. In our case $f(x, y) = x^2+qy^2$, hence $\Delta_f = -4q$.

Additionally the bound $e_{22}e_{33} \leq \frac{N}{2}$ from the proof of Theorem 5.8 becomes $e_{22}e_{33} \leq \frac{N}{2q}$ by Proposition 5.2. And we have $N = 2qN(I) = q \cdot O(1)\sqrt{p}$. $\square$

For an explicit formula for $C(-4q)$, see the results of Moree and Osburn [43], and for a summary of results on $C(\cdot)$ see the work of Brink, Moree and Osburn [8].

Next, we note how the complexity changes in other contexts:

Remark 5.12 (Suborders) Suppose you are given a quaternion order $\mathcal{O} \subset B_{p,\infty}$ which is not necessarily maximal. As stated in Remark 5.5, Algorithm 5.1 still works. The complexity is the same with the subtlety that N is multiplied by the index of the suborder within a maximal order.

Remark 5.13 (p-extremal orders) Suppose $\mathcal{O}$ is a p-extremal order, and has suborder $R + jR \subseteq \mathcal{O}$ and we are trying to find an embedding into this suborder. For ω a generator of R and $\alpha = \alpha_0 + \alpha_1\omega$, $\alpha' = \alpha_0' + \alpha_1'\omega \in R$, the norm equation is:

$$nrd(\alpha + j\alpha') = f(\alpha_0, \alpha_1) + pf(\alpha_0', \alpha_1') = d$$

where f is a binary quadratic form of discriminant $disc(R)$. The approach in the KLPT algorithm [34, Sect. 3.2] randomly samples α_0' and α_1' until $d - pf(\alpha_0', \alpha_1') = q$ is a prime which is split in R where the ideal factors of (q) are principal, and hence its generator gives a solution to $q = f(\alpha_0, \alpha_1)$. Assuming sampled integers q satisfy the distribution of primes less than d, this takes roughly $2h(R)\log(d)$ iterations. Note that we require d to be large enough for the set $d - pf(x, y)$ to contain enough primes.

Our algorithm is very similar but works in reverse. Also assuming d is sufficiently large ($d > p^{2+\epsilon}$), we sample k until $q = \frac{d - f(\alpha_0, \alpha_1)}{p} \in \mathbb{Z}$ has a solution to the equation $q = f(\alpha_0', \alpha_1')$. By the same argument, if we wait until q is a prime, split in R, where the ideals of norm q are principal, we are guaranteed a solution, so we also expect $2h(R)\log(d)$ iterations, which is efficient assuming R has a small class number.

Finally note we do not necessarily need a factorization oracle using the technique presented in the next section.

5.3 Rerandomization and small discriminant

Consider the special case of small discriminant orders $\mathbb{Z}[\omega]$ in Algorithm 5.1. Previously, the best known efficient algorithm, as stated in [66] was simply to look for small vectors in $\mathcal{O}$. This works for $|\operatorname{disc}(\mathbb{Z}[\omega])| < 2\sqrt{p} - 1$ and is stated below. Note that an alternative approach that heuristically works for $|\operatorname{disc}(\mathbb{Z}[\omega])| < p^{0.8}$ is outlined in [4].

Proposition 5.14 [66, Proposition 6] *Assume that* $|\operatorname{disc}(\mathbb{Z}[\omega])| < 2\sqrt{p} - 1$. *Then, there is a probabilistic polynomial time algorithm that solves Problem 2.6.*

Under certain heuristics, we can rerandomize Algorithm 5.1 by considering isomorphic orders, potentially in different representations of $B_{p,\infty}$, to avoid factoring and bound the denominator $N < O(\sqrt{p})$. The result of this is a corollary (Corollary 5.16) which gives a heuristic polynomial time algorithm for solving Problem 2.6 for $\operatorname{disc}(\mathbb{Z}[\omega])$ in $O(p)$, or deciding that no solution exists.

The first step is to bound the number of values to try Cornacchia on in Algorithm 5.1. By Cornacchia's algorithm, here we mean finding all solutions to $x^2 + qy^2 = v$, not just primitive ones.

Lemma 5.15 *Fix a positive integer M. As in the context of Algorithm 5.1, take $d = \operatorname{nrd}(\omega)$, N as the common denominator of the HNF basis, and k the variable we iterate over. Suppose we have*

$$d \leq \frac{qp^2(\frac{M}{2} - 1)^2}{N^2},$$

then Algorithm 5.1 performs at most M executions of Cornacchia's algorithm, each with some additional polynomial time arithmetic, and all other operations are polynomial time.

Proof As stated in proof of Theorem 5.7, all computations performed in the algorithm, excluding the iterations over k, can be achieved in polynomial time. For each iteration over k we compute a value v using basic arithmetic operations, and naively use Cornacchia's algorithm only once. Iterating over k happens twice, once using r_+ and once using r_-. Therefore it is enough to show for each $r \in \{r_+, r_-\}$ we perform at most $\lfloor M/2 \rfloor$ iterations over k. As in proof of Theorem 5.7, and taking into account general p, we can consider the maximum number of iterations over k for each r. It is sufficient to show this is bounded by $M/2$,

$$\left\lfloor \frac{1}{pe_{11}} \left(\sqrt{\frac{d - (\alpha_0 e_{00})^2}{q}} - \alpha_0 e_{01} - re_{11} \right) \right\rfloor + 1 \leq \frac{M}{2}$$

which since $e_{01} \geq 0$ by Proposition 5.2, is certainly true if

$$\frac{1}{pe_{11}} \cdot \sqrt{\frac{d}{q}} + 1 \leq \frac{M}{2}$$

and noting $e_{11} \geq \frac{1}{N}$, we see this is true by its equivalence to the condition in statement of the Lemma,

$$d \leq \frac{qp^2(\frac{M}{2} - 1)^2}{N^2}.$$

□

From this, we obtain a result about $\operatorname{disc}(\mathbb{Z}[\omega])$ since we can translate generator ω to either $\sqrt{-\operatorname{disc}(\mathbb{Z}[\omega])}/2$ or $(1 + \sqrt{-\operatorname{disc}(\mathbb{Z}[\omega])})/2$ hence $N(\omega) = d \leq (|\operatorname{disc}(\mathbb{Z}[\omega])| + 1)/4$. Recalling that we can bound N, the denominator of $\mathcal{O}$, to $O(\sqrt{p})$, we see that Lemma 5.15 says that when $\operatorname{disc}(\mathbb{Z}[\omega])$ is in $O(p)$, and assuming q in $O(1)$, the only potentially expensive part in Algorithm 5.1 is Cornacchia on a constant number of instances.

The general idea of our rerandomized version is then the common technique of only running Cornacchia on "good" instances. However, if the discriminant is small, then the embedding is with large probability unique, hence we might end up discarding the correct solution. Therefore, we need to rerandomize until *all* $O(1)$ Cornacchia instances are good before the algorithm can be sure that no embedding exists.

We define a "good" instance to be $x^2 + qy^2 = v_k$ where v_k can be factorized in polynomial time, has $O(\log\log(N^2 d))$ distinct prime factors, and $O(\log(N^2 d))$ square divisors. The set of prime numbers satisfies these conditions, and with the heuristic that the events of each v_k being prime are independent and follow from the density of primes, we expect at most some constant multiple of $\log(N^2 d)^C$ iterations until C of the Cornacchia instances are primes. With $C = O(1)$ instances from the small discriminant condition, this is efficient.

Now we discuss how we rerandomize the order. Let $\mathcal{O}_0 \subseteq B_{p,\infty}$ be a maximal order with negligible denominator K (e.g. the "standard" maximal order from Proposition 5.1). As has been pointed out, any maximal order $\mathcal{O} \subseteq B_{p,\infty}$ will have denominator bounded by KN, where N is the norm of the connecting ideal from $\mathcal{O}_0$ to $\mathcal{O}$. Hence we can consider any equivalent ideal $J = I\gamma$ of small norm, and instead solve the problem in the isomorphic order $\mathcal{O}_R(J) = \gamma^{-1}\mathcal{O}\gamma$, before transferring the solution back to $\mathcal{O}$.

As we rerandomize, we might need to try many distinct $J \sim I$ of small norm. Heuristically, for random orders $\mathcal{O}$, we can expect there to be an abundance of small, equivalent ideals (i.e. with $N(J)$ in $O(\sqrt{p})$). The problem is that this heuristic fails completely if there exists some equivalent ideal I' with $\mathrm{nrd}(I') \ll \sqrt{p}$, i.e. in the case that $\mathcal{O}$ is too "close" to $\mathcal{O}_0$. We can fix this by considering other maximal orders $\mathcal{O}_0'$ with negligible denominator in other representations of $B_{p,\infty}$.

Specifically, we can generate representations $B_i = (-q_i, -p \mid \mathbb{Q})$, where we take q_i to be the smallest primes satisfying

$$q_i \equiv 3 \pmod 4, \quad \left(\frac{-q_i}{p}\right) = -1.$$

These quaternion algebras are indeed ramified at p and ∞ [62, Proposition 14.2.7]. In each of these representations, regardless of congruence conditions on p, we can take a standard choice of maximal order $\mathcal{O}_{0,i}$ with denominator $2q_i$ as

$$\mathcal{O}_{0,i} := \mathbb{Z} \oplus \mathbb{Z}\frac{1+i}{2} \oplus \mathbb{Z}j \oplus \mathbb{Z}\frac{(1+i)j}{2q_i},$$

[62, Exercise 15.5]. Heuristically, these choices of maximal orders of the different quaternion algebra representations are "independent" in the sense that there is no reason that these should be close to each other, so it is enough to try a small, fixed number of such orders, as (heuristically), the probability that $\mathcal{O}$ is close to all $\mathcal{O}_{0,i}$ is negligible.

Finally, the explicit isomorphisms $B_i \cong B_j$ are also easy to find and compute, using [25, Lemma 10]. The whole algorithm is summarized in Algorithm 5.2.

Algorithm 5.2: Rerandomized version of Algorithm 5.1

```
Data: A $\mathbb{Z}[\omega]$-orientable maximal order $\mathcal{O} \subset B$, where $B$ is a quaternion algebra ramified at $p$ and $\infty$.
      $r \in \mathbb{N}$, a maximum number of randomizations to try.
Result: An element $\alpha \in \mathcal{O}$, which defines an embedding $\iota : \mathbb{Z}[\omega] \hookrightarrow \mathcal{O}$ by $\omega \mapsto \alpha$.
1  Compute $r$ representations $B_i = (-q_i, -p, \mathbb{Q})$ for $q_i \in O(1)$ of $B_{p,\infty}$;
2  for $i = 1, \dots, r$ do
3      Set $\mathcal{O}_{0,i} \subseteq B_i$ to be a maximal order with denominator dividing $2q_i$;
4      Compute an isomorphism $\varphi_i : B \to B_i$;
5      Set $\mathcal{O}_i = \varphi(\mathcal{O})$;
6      Let $I$ be a connecting $(\mathcal{O}_{0,i}, \mathcal{O}_i)$-ideal;
7      for $J = I\gamma$, with $N(J)$ in $O(\sqrt{p})$ do
8          Compute $\beta$ by running Algorithm 5.1 on $\mathcal{O}_R(J)$, only running Cornacchia on prime numbers;
9          if $\beta \neq \perp$ then
10             Set $\alpha' = \gamma\beta\gamma^{-1}$;
11             Return $\varphi_i^{-1}(\alpha')$;
12         end
13     end
14 end
```

This gives the following corollary:

Corollary 5.16 *For arbitrary $p \neq 2$, given a maximal order $\mathcal{O} \subseteq B_{p,\infty}$, and a quadratic order $\mathfrak{O}$ with $|\mathrm{disc}(\mathfrak{O})|$ in $O(p)$, Algorithm 5.2 computes an $\mathfrak{O}$-embedding of $\mathcal{O}$, or decides that none exists. Under the heuristics discussed above the algorithm terminates in probabilistic polynomial time in* $\log(p)$.

Proof By Lemma 5.15, and the subsequent discussion, the only potentially expensive step of running Algorithm 5.1 are the (constant number of) Cornacchia instances. By only running the Cornacchia on prime instances (or more generally, "good" instances), we expect to have to run Algorithm 5.1 at most $O(\log(p)^{O(1)})$ times by the prime number theorem, and the fact that v is in $O(p)$. Further, it is clear that if all $O(1)$ values to try Cornacchia on is prime, and a solution is still not found, a solution cannot exist, hence the algorithm can conclude that no solution exists. □

5.4 From embeddings to primitive embeddings

Algorithms 5.1 and 5.2 find all possible embeddings $\iota : \mathbb{Z}[\omega] \hookrightarrow \mathcal{O}$. Every embedding gives a primitive $\mathbb{Z}[\omega']$-embedding where $\mathbb{Z}[\omega] \subseteq \mathbb{Z}[\omega']$. We separate between these two cases, and call those embeddings which give primitive superorder embeddings $\mathbb{Z}[\omega] \subsetneq \mathbb{Z}[\omega']$ *imprimitive embeddings*. Finding primitive embeddings solves Problem 2.6, and we consider this question in Sect. 5.5. In this section, we focus on the imprimitive embeddings. First, we explain how to differentiate between primitive and imprimitive embeddings.

For any element $\alpha \in \mathcal{O}$, write $\tilde{\alpha}$ for its class in the lattice $\mathcal{O}/\mathbb{Z}$. Consider the discriminant form

$$\begin{aligned} \Delta : \mathcal{O}/\mathbb{Z} &\longrightarrow \mathbb{Q} \\ \tilde{\alpha} &\longmapsto \mathrm{Tr}(\alpha) - 4\,\mathrm{nrd}(\alpha), \end{aligned}$$

an integral quadratic form of rank 3. It does not depend on the choice of a representative α of the class $\tilde{\alpha}$, and we also write $\Delta(\alpha)$. Let d be an integer. We say that a solution $\alpha \in \mathcal{O}$ of $\Delta(\alpha) = d$ is *primitive* if $\tilde{\alpha}$ is a primitive element of the lattice $\mathcal{O}/\mathbb{Z}$, i.e., it is not of the form $\tilde{\alpha} = b\tilde{\beta}$ for some element $\tilde{\beta} \in \mathcal{O}/\mathbb{Z}$ and integer $b > 1$. We now show primitive solutions correspond directly to primitive embeddings.

Lemma 5.17 *An element $\alpha \in \mathcal{O}$ is a primitive solution of $\Delta(\alpha) = d$ if and only if $\mathbb{Z}[\alpha] \subseteq \mathcal{O}$ is a primitive embedding.*

Proof Suppose α is an imprimitive solution of $\Delta(\alpha) = d$, i.e., there are integers a and $b > 1$ such that $(\alpha - a)/b \in \mathcal{O}$. Then, $\mathbb{Z}[\alpha] \subsetneq \mathbb{Z}[(\alpha - a)/b] \subseteq \mathcal{O}$, hence $\mathbb{Z}[\alpha] \subseteq \mathcal{O}$ is not primitive. Conversely, suppose $\mathbb{Z}[\alpha] \subseteq \mathcal{O}$ is not primitive, so there exists $\beta \in (\mathbb{Q}[\alpha] \cap \mathcal{O}) \backslash \mathbb{Z}[\alpha]$. There exists integers a and $b > 1$ such that $\alpha = a + b\beta$. In particular, $\tilde{\alpha} = b\tilde{\beta}$, so α is not a primitive solution. □

Now we know primitive embeddings come from primitive solutions, we can determine if an embedding is primitive, and if not extend it to its superorder, very fast using a gcd computation:

Lemma 5.18 *Given a maximal order $\mathcal{O}$ with basis e_0, e_1, e_2, e_3 and an element $\alpha \in \mathcal{O}$ of trace $t = \mathrm{Tr}(\omega)$ and norm $d = \mathrm{nrd}(\omega)$, there is a polynomial time algorithm which:*

- *determines whether embedding ι defined by extending $\omega \mapsto \alpha$ is a primitive or imprimitive embedding of $\mathbb{Z}[\omega]$,*
- *and if it's imprimitive outputs (a, b, α') defining a superorder $\mathbb{Z}[\frac{\omega - a}{b}] \supsetneq \mathbb{Z}[\omega]$ which ι can be extended to, through the map $\frac{\omega - a}{b} \mapsto \alpha'$.*

Proof First convert the upper diagonal basis e_0, e_1, e_2, e_3 of $\mathcal{O}$ into an lower diagonal basis f_0, f_1, f_2, f_3,

$$\begin{aligned}\mathcal{O} = \langle f_0, f_1, f_2, f_3\rangle_{\mathbb{Z}} = \langle & f_{00}, \\ & f_{10} + f_{11}i, \\ & f_{20}i + f_{21}i + f_{22}j, \\ & f_{30} + f_{31}i + f_{32}j + f_{33}k\rangle_{\mathbb{Z}}\end{aligned} \tag{12}$$

and compute a function applying the change of basis transformation taking coefficients of e_is onto coefficients of f_is. This is polynomial time as a variant of the Hermite normal form algorithm, and can be seen as precomputation. Then given a solution α, we change the basis to obtain:

$$\alpha = \gamma_0 f_0 + \gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3$$

Since $\mathcal{O}$ is a ring we have $1 \in \mathcal{O}$, and every norm is integral so it cannot contain a rational number less than one. Hence $f_0 = f_{00} = 1$. For α to be a primitive solution there should be no $a, b \in \mathbb{Z}$ with $b > 1$ such that $\alpha - a = b\tau$ where $\tau \in \mathcal{O}$. Equivalently, for any a, when expressing $\alpha - a$ in terms of f_is, the coefficients should not all be divisible by any $b > 1$. Note that we have:

$$\alpha - a = (\gamma_0 - a) f_0 + \gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3$$

where $a = \gamma_0$ can be chosen, setting the first coefficient to zero. Then the solution is primitive if and only if $\gamma_1, \gamma_2, \gamma_3$ share no factor. This can be checked with a gcd computation. Note that if the solution is imprimitive, so we have $gcd(\gamma_1, \gamma_2, \gamma_3) = b > 1$, we return $(\gamma_0, b, \frac{\gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3}{b})$ defining an embedding giving a primitive $\mathbb{Z}[\frac{\omega - \gamma_1}{b}]$-embedding. □

Algorithm 5.3 is a concise version of this. Hence using Algorithm 5.1 for the embedding we find we always get a primitive $\mathbb{Z}[\omega']$-embedding without affecting asymptotic time complexities. Furthermore, if we iterate over the full range of k in Algorithm 5.1, we find all embeddings and hence all primtive superorder embeddings.

Algorithm 5.3: Checking if a solution is primitive, and getting primitive superorder embedding.

```
Data: Element $\alpha \in \mathcal{O}$ in terms of a basis $e_i$ which is a solution to the discriminant form of $\omega$.
Result: True if it is a primitive solution, otherwise False and output $(a, b, \alpha')$ where $\alpha'$ is a primitive
        solution to the discriminant form of $\frac{\omega - a}{b}$.
1  Precompute lower diagonal Hermite normal form basis $f_i$. And store operations performed giving a
   linear change of basis transformation matrix $M$;
2  Apply transformation $M$ to $\alpha$, giving coefficients $\gamma_i$ such that $\alpha = \gamma_0 f_0 + \gamma_1 f_1 + \gamma_2 f_2 + \gamma_3 f_3$;
3  Let $S = \{\gamma_1, \gamma_2, \gamma_3\} \setminus \{0\}$;
4  if $|S| = 0$ then return False, $(\gamma_0, \infty, 0)$
5  Compute $g = \gcd(S)$ using Euclidean algorithm;
6  if $g == 1$ then
7      return True;
8  else
9      return False, $(\gamma_0, g, \frac{\gamma_1}{g} f_1 + \frac{\gamma_2}{g} f_2 + \frac{\gamma_3}{g} f_3)$;
10 end
```

5.5 Finding primitive $\mathbb{Z}[\omega]$-embeddings-solving problem 2.6

To solve Problem 2.6 we must find *primitive* $\mathbb{Z}[\omega]$-embeddings. We have Algorithm 5.1 for finding embeddings, and we have Algorithm 5.3 which can check if an embedding is primitive.

To combine them, we modify Algorithm 5.1 to include the pre-computation of basis f_i and the change of basis transformation at the start, then when each solution is found, check if it is primitive using Algorithm 5.3 and only stop searching if it is. The worst-case running time doesn't change, since finding a primitive embedding takes at most as long as finding all embeddings. However, the average-case running time does increase, heuristically by the total number of solutions divided by the number of primitive solutions. We now provide a further heuristic argument that this ratio can be bounded from above.

Let $f(\gamma_1, \gamma_2, \gamma_3) = \lambda$ be the solution to the ternary quadratic norm form of the order defined in 12 with fixed trace. We have shown the solution is primitive if and only if $gcd(\gamma_1, \gamma_2, \gamma_3) = gcd(|\gamma_1|, |\gamma_2|, |\gamma_3|) = 1$.

Consider the rational solution $x_1 = x_2 = x_3 \in \mathbb{Q}$ such that $f(x_1, x_1, x_1) = wx_1^2 = \lambda$ where $w \in \mathbb{Q}$ is the sum of coefficients in the form. Then $|x_1| = |\sqrt{\lambda/w}|$ and since the norm form is positive definite, this means if one variable were to increase, another must decrease in absolute value. Therefore $\min(|\gamma_1|, |\gamma_2|, |\gamma_3|) \leq \lfloor\sqrt{\lambda/w}\rfloor$.

Now reconsider our integral solution. Suppose the solution is not primitive so $gcd(|\gamma_1|, |\gamma_2|, |\gamma_3|) \neq 1$, then there is a prime number ≥ 2 that divides all three numbers. This prime factor must be in the set $S = \{2, 3, 5, ..., \lfloor\sqrt{\lambda/w}\rfloor\} \cap \{\text{Primes } p\}$ as it must divide the smallest of these three numbers.

Heuristically, we assume that $\gamma_1, \gamma_2, \gamma_3$ are distributed like random numbers in the sense that some $q \in S$ divides one of them with uniformly random probability $1/q$. And assume independence of the probabilities of different factors $q_1, q_2 \in S$ occurring. Then the probability 2 divides all three numbers is $1/2^3$, the probability 3 divides them is $1/3^3$, and the probability any $q \in S$ divides them is $1/q^3$. Combined, the probability a number in S divides all three is:

$$\mathbb{P}[(\gamma_1, \gamma_2, \gamma_3) \text{ imprimitive}] = \sum_{\text{primes } q \in S} \frac{1}{q^3} \leq \sum_{\text{all primes } q \in \mathbb{N}} \frac{1}{q^3} = P(3) \leq 0.175$$

where $P(3)$ is the prime zeta function at 3. Hence the probability a random solution is primitive is over 80% so if we find 5 independent solutions we would expect at least one to be primitive. Therefore assuming the heuristics above, if we modify algorithm 5.1 to ignore imprimitive solutions, the average running time should only increase by at most a factor of 5.

Note that this argument makes some strong assumptions. Experimentally for some parameters, we see the probability the first solution found is primitive is around 80%, but on other parameter choices, it is considerably lower. With all the parameters we tested, we found the probability is always over 50%, which still suggests the average running time is only worsened by a small factor, but it gives reason to doubt these assumptions. In particular, consider independence. The existence of embeddings comes with symmetry hence we may find two solutions where there is only a change of sign in the defining formulae. This means the probability q divides the coefficients of one solution might have a strong dependence on whether q divides the coefficients of the second solution. We leave a more complete analysis of the probability of finding primitive embeddings for future research.

Appendix A Singular points on the curve $\Phi_\ell(X, Y) = 0$

If K is a field, we denote by $\tilde{X}_0(\ell, K)$ the curve $\Phi_\ell(X, Y) = 0$ over the field K. We also define:

$$S_0(\ell, \mathbb{F}_{p^2}) := \left\{ j \in \mathbb{F}_{p^2} \text{ supersingular} \;\middle|\; \exists j' \in \mathbb{F}_{p^2},\ \ \Phi_\ell(j, j') = \frac{\partial \Phi_\ell}{\partial X}(j, j') = \frac{\partial \Phi_\ell}{\partial Y}(j, j') = 0 \right\}.$$

Lemma A.1 *Assume that $2\ell < p$. Then, $\# S_0(\ell, \mathbb{F}_{p^2}) = O(\ell^{3+o(1)})$.*

Proof Let $j(E) \in S_0(\ell, \mathbb{F}_{p^2})$ and $j(E') \in \mathbb{F}_{p^2}$ such that $(j(E), j(E'))$ is a singular point of $\tilde{X}_0(\ell, \mathbb{F}_{p^2})$ i.e. such that

$$\Phi_\ell(j(E), j(E')) = \frac{\partial \Phi_\ell}{\partial X}(j(E), j(E')) = \frac{\partial \Phi_\ell}{\partial Y}(j(E), j(E')) = 0.$$

Schoof proved in [58, Sect. 7] that there exists a lift $(j(\widetilde{E}), j(\widetilde{E}'))$ over $\mathbb{C}$ of $(j(E), j(E'))$ that is also a singular point of the curve $\tilde{X}_0(\ell, \mathbb{C})$. Schoof deduced that there exists two ℓ-isogenies $\phi, \psi : \widetilde{E} \longrightarrow \widetilde{E}'$ over $\mathbb{C}$ that are not equal up to pre or post composition by an isomorphism, so that $\varphi := \widehat{\psi} \circ \phi$ is a cyclic endomorphism of $\widetilde{E}$ of degree ℓ^2. Hence, φ is non-scalar and $\mathrm{End}(\widetilde{E})$ is isomorphic to an imaginary quadratic order $\mathfrak{O}$ and $\mathbb{Z}[\varphi]$ is mapped to a suborder of $\mathfrak{O}$ via this isomorphism. It follows that $\mathrm{disc}(\mathfrak{O}) \mid \mathrm{disc}(\mathbb{Z}[\varphi])$. But

$$\mathrm{disc}(\mathbb{Z}[\varphi]) = \mathrm{Tr}(\varphi)^2 - 4\deg(\varphi) = \mathrm{Tr}(\varphi)^2 - 4\ell^2.$$

Since $\mathfrak{O}$ is imaginary quadratic, so is $\mathbb{Z}[\varphi]$ and $\mathrm{disc}(\mathbb{Z}[\varphi]) < 0$, so that $|\mathrm{disc}(\mathfrak{O})| \le |\mathrm{disc}(\mathbb{Z}[\varphi])| \le 4\ell^2$.

Since $2\ell < p$, p does not divide the conductor of $\mathfrak{O}$ (otherwise, $p^2 \mid \mathrm{disc}(\mathfrak{O})$ so $p^2 \le 4\ell^2$). It follows by [46, Lemma 3.1] (generalizing [36, Chapter 13, Theorem 12]), that E is (primitively) $\mathfrak{O}$-oriented. Besides, by [46, Proposition 3.3 and Theorem 3.4] there are at most $2\#\,\mathrm{Cl}(\mathfrak{O})$ j-invariants of supersingular $\mathfrak{O}$-oriented curves. By Siegel's theorem [59], we have $\#\,\mathrm{Cl}(\mathfrak{O}) = O(|\mathrm{disc}(\mathfrak{O})|^{1/2+o(1)}) = O(\ell^{1+o(1)})$, so there are at most $O(\ell)$ j-invariants of $\mathfrak{O}$-oriented supersingular elliptic curves. Taking into account all possible imaginary quadratic orders $\mathfrak{O}$ of discriminant $|\mathrm{disc}(\mathfrak{O})| \le 4\ell^2$, we conclude that $j(E)$ lies in a set of cardinality $O(\ell^{3+o(1)})$, which completes the proof. □

Lemma A.2 *Assume that $\log(\ell) \ll \log(p)$. Then, computing an ℓ-isogeny between two ℓ-isogenous supersingular j-invariants chosen uniformly at random takes on average $\tilde{O}(\ell^2 \log(p))$ operations over $\mathbb{F}_{p^2}$.*

Proof As discussed in Sect. 2.5, the average number of operations over $\mathbb{F}_{p^2}$ to compute an ℓ-isogeny between supersingular ℓ-isogenous j-invariants is:

$$\begin{aligned} N := &(1 - \mathbb{P}((j(E), j(E'))\text{singular}) - \mathbb{P}(j(E) \text{ or } j(E') = 0, 1728))\tilde{O}(\ell^2 \log(p)) \\ &+ (\mathbb{P}((j(E), j(E')) \text{ singular}) + \mathbb{P}(j(E) \text{ or } j(E') = 0, 1728))O(\ell^{7/2}) \end{aligned}$$

Since there are $\sim p/12$ supersingular j-invariants by [60, Theorem V.4.1.c], we obtain by Lemma A.1:

$$\mathbb{P}((j(E), j(E')) \text{ singular}) \le \mathbb{P}(j(E) \in S_0(\ell, \mathbb{F}_{p^2})) = O\left(\frac{\ell^{3+o(1)}}{p}\right)$$

Besides, $\mathbb{P}(j(E) \text{ or } j(E') = 0, 1728) = O(1/p)$. Since $\log(\ell) \ll \log(p)$, we have $\ell^{13/2+o(1)} \ll p$ so the dominant term of N is $\tilde{O}(\ell^2 \log(p))$ and all other terms are negligible. Hence, $N = \tilde{O}(\ell^2 \log(p))$ and the proof is complete. □

Acknowledgements This project began at KU Leuven Isogeny Days in 2022, and the authors extend their gratitude to the organizers.

Code availability https://github.com/jtcc2/finding-orientations.

Declarations

Conflict of interest The authors have no financial or proprietary interests in any material discussed in this article.

References

1. Arpin S., Chen M., Lauter K.E., Scheidler R., Stange K.E., Tran H.T.: Orienteering with one endomorphism. La Mat. (2023). https://doi.org/10.1007/s44007-023-00053-2.
2. Arpin S., Chen M., Lauter K.E., Scheidler R., Stange K.E., Tran H.T.N.: Orientations and cycles in supersingular isogeny graphs. To appear in the Proceedings of Women in Number Theory 5 (2022).
3. Basso A., Maino L., Pope G.: Festa: Fast Encryption from Supersingular Torsion Attacks. Springer, Berlin (2023).
4. Benčina B., Kutas P., Merz S.-P., Petit C., Stopar M., Weitkämper C.: Improved algorithms for finding fixed-degree isogenies between supersingular elliptic curves. Cryptol. ePrint Arch. **2023**, 1618 (2023).
5. Bernays P.: Über die Darstellung von positiven: ganzen Zahlen durch die primitiven, binären quadratischen Formen einer nicht-quadratischen Diskriminante. Dieterich, Mainz (1912).
6. Bernstein D.J., De Feo L., Leroux A., Smith B.: Faster computation of isogenies of large prime degree. In: Proceedings of the Fourteenth Algorithmic Number Theory Symposium, pp. 39–55, University of California, Berkeley, MSP (2020).
7. Bostan A., Morain F., Salvy B., Schost E.: Fast algorithms for computing isogenies between elliptic curves. Math. Comput. **77**, 1755–1778 (2008).
8. Brink D., Moree P., Osburn R.: Principal forms X^2+nY^2 representing many integers. Abh. Math. Semin. Univ. Hambg. **81**(2), 129–139 (2011).
9. Buhler J.P., Lenstra H.W., Pomerance C.: Factoring integers with the number field sieve. In: Lenstra A.K., Lenstra H.W. (eds.) The Development of the Number Field Sieve, pp. 50–94. Springer, Berlin (1993).
10. Castryck W., Decru T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23–27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pp. 423–447. Springer (2023).
11. Castryck W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III, volume 11274 of Lecture Notes in Computer Science, pp. 395–427. Springer (2018).
12. Charles D.X., Goren E.Z., Lauter K.E.: Cryptographic hash functions from expander graphs. J. Cryptol. **22**(1), 93–113 (2009).
13. Colò L., Kohel D.: Orienting supersingular isogeny graphs. J. Math. Cryptol. **14**(1), 414–437 (2020).

14. Coppel W.A.: Number Theory: An Introduction to Mathematics. Springer, New York (2009).
15. Couveignes, J.M.: Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, page 291, (2006).
16. Dartois P., Leroux A., Robert D., Wesolowski B.: SQISignHD: new dimensions in cryptography. IACR Cryptol. ePrint Arch., pp. 436 (2023).
17. de Bruijn N.G.: On the number of positive integers $\leq x$ and free of prime factors $>y$, II. Proc. Koninkl. Nederl. Akad. van Wetenschappen Ser. A **3**, 239–247 (1966).
18. De Feo L., de Saint Guilhem C.D., Fouotsa T.B., Kutas P., Leroux A., Petit C., Silva J., Wesolowski B.: Séta: supersingular encryption from torsion attacks. In: Tibouchi M., Wang H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV, volume 13093 of Lecture Notes in Computer Science, pp. 249–278. Springer (2021).
19. Delfs C., Galbraith S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Des. Codes Cryptogr. **78**(2), 425–440 (2016).
20. Deuring M.: Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. Abh. Math. Sem. Hansischen Univ. **14**, 197–272 (1941).
21. Eisenträger K., Hallgren S., Lauter K.E., Morrison T., Petit C.: Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In: Nielsen J.B., Rijmen V. (eds.) 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, volume 10822 of Lecture Notes in Computer Science, pp. 329–368. Springer (2018).
22. Eisenträger 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 Ser. **4**, 215–232 (2020).
23. Elkies N.D.: Elliptic and modular curves over finite fields and related computational issues. In: Computational perspectives on number theory (Chicago, IL, 1995), volume 7 of AMS/IP Studies Advanced Mathematics, pp. 21–76. American Mathematics Society, Providence, RI (1998).
24. Erdös P., Kac M.: The Gaussian law of errors in the theory of additive number theoretic functions. Am. J. Math. **62**, 738–742 (1940).
25. Eriksen J.K., Panny L., Sotáková J., Veroni M.: Deuring for the people: supersingular elliptic curves with prescribed endomorphism ring in general characteristic. IACR Cryptol. ePrint Arch., 106 (2023).
26. Feo L.D., Fouotsa T.B., Kutas P., Leroux A., Merz S., Panny L., Wesolowski B.: SCALLOP: scaling the csi-fish. In: Boldyreva A., Kolesnikov V. (eds.) Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7–10, 2023, Proceedings, Part I, volume 13940 of Lecture Notes in Computer Science, pp. 345–375. Springer (2023).
27. Feo L.D., Kohel D., Leroux A., Petit C., Wesolowski B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: Moriai S., Wang H. (eds.) 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, volume 12491 of Lecture Notes in Computer Science, pp. 64–93. Springer, 2020.
28. Fuselier J., Iezzi A., Kozek M., Morrison T., Namoijam C.: Computing supersingular endomorphism rings using inseparable endomorphisms (2023).
29. Granville A.: Smooth numbers: computational number theory and beyond. Math. Sci. Res. Inst. Publ. **44**, 267–323 (2008).
30. Hafner J.L., McCurley K.S.: Asymptotically fast triangularization of matrices over rings. SIAM J. Comput. **20**(6), 1068–1083 (1991).
31. Jao D., Feo L.D.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang B. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings, volume 7071 of Lecture Notes in Computer Science, pp. 19–34. Springer (2011).
32. Kaneko M.: Supersingular j-invariants as singular moduli $mod\ p$. Osaka J. Math. **26**(4), 849–855 (1989).
33. Kani E.: The number of curves of genus two with elliptic differentials. J. Reine Angew. Math. **1997**(485), 93–122 (1997).
34. Kohel D., Lauter K., Petit C., Tignol J.-P.: On the quaternion-isogeny path problem. LMS J. Comput. Math. **17**(A), 418–432 (2014).
35. Landau E.: Über die einteilung der positiven ganzen zahlen in vier klassen nach der mindestzahl der zu ihrer additiven zusammensetzung erforderlichen quadrate. Arch. Math. Phys. **13**, 305–312 (1908).
36. Lang S.: Elliptic functions, volume 112 of Graduate Texts in Mathematics, second edition. Springer, New York. With an appendix by J. Tate (1987).

37. Lehmer D.H.: Computer technology applied to the theory of numbers. In: Studies in Number Theory, pp.117–151 (1969).
38. Lenstra H.W.: Factoring integers with elliptic curves. Ann. Math. **126**(3), 649–673 (1987).
39. Leroux A.: Computation of hilbert class polynomials and modular polynomials from supersingular elliptic curves. Cryptology ePrint Archive, Paper 2023/064 (2023). https://eprint.iacr.org/2023/064.
40. Lubicz D., Robert D.: Fast change of level and applications to isogenies. In: Research in Number Theory (ANTS XV Conference), 9(1) (2023).
41. Maino L., Martindale C., Panny L., Pope G., Wesolowski B.: A direct key recovery attack on SIDH. In: Hazay C., Stam M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23–27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pp. 448–471. Springer (2023).
42. Martin G.: An asymptotic formula for the number of smooth values of a polynomial. J. Number Theory **93**, 108–182 (2002).
43. Moree P., Osburn R.: Two-dimensional lattices with few distances. Enseign. Math. **52**(3–4), 361–380 (2006).
44. Moriya T.: Is-cube: an isogeny-based compact kem using a boxed sidh diagram. In: Cryptology ePrint Archive (2023).
45. Nakagawa K., Onuki H.: Qfesta: Efficient algorithms and parameters for festa using quaternion algebras. In: Cryptology ePrint Archive (2023).
46. Onuki H.: On oriented supersingular elliptic curves. Finite Fields Appl. **69**, 101777 (2021).
47. Page A., Wesolowski B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Cryptology ePrint Archive, Paper 2023/1399. https://eprint.iacr.org/2023/1399 (2023).
48. Pizer A.: An algorithm for computing modular forms on $\gamma_0(n)$. J. Algebra **64**(2), 340–390 (1980).
49. Pollack P., Treviño E.: Finding the four squares in Lagrange's Theorem. Integers **18A**, A15 (2018).
50. Pomerance C.: Fast, rigorous factorization and discrete logarithm algorithms. In: Johnson D.S., Nishizeki T., Nozaki A., Wilf H.S. (eds.) Discrete Algorithms and Complexity, pp. 119–143. Academic Press, New York (1987).
51. Robert D.: Efficient algorithms for abelian varieties and their moduli spaces (2021). http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf.
52. Robert D.: Breaking SIDH in polynomial time. In: Cryptology ePrint Archive, Paper 2022/1038 (2022).
53. Robert D.: Evaluating isogenies in polylogarithmic time. In: IACR Cryptology ePrint Archive, p. 1068 (2022).
54. Robert D.: Some applications of higher dimensional isogenies to elliptic curves (overview of results). In: Cryptology ePrint Archive, Paper 2022/1704. https://eprint.iacr.org/2022/1704 (2022).
55. Robert D.: Breaking SIDH in polynomial time. In: Hazay C., Stam M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23–27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pp. 472–503. Springer (2023).
56. Rostovtsev A., Stolbunov A.: Public-key cryptosystem based on isogenies. In: Cryptology ePrint Archive, Paper 2006/145 (2006). https://eprint.iacr.org/2006/145.
57. Sawilla R.E., Silvester A.K., Williams H.C.: A new look at an old equation. In: Algorithmic Number Theory: 8th International Symposium, ANTS-VIII Banff, Canada, May 17–22, 2008 Proceedings 8, pp. 37–59. Springer (2008).
58. Schoof R.: Counting points on elliptic curves over finite fields. J. Théo. Nombr. Bordeaux **7**, 219–254 (1995).
59. Seigel C.L.: Über die classenzahl quadratischer zahlkörper. Acta Arith. **1**(1), 83–86 (1935).
60. Silverman J.H.: The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 1, 2nd edn Springer, Dordrecht (2009).
61. Stein W., et al.: Sage Mathematics Software (Version 10.0). The Sage Development Team (2023). http://www.sagemath.org.
62. Voight J.: Quaternion Algebras. Graduate Texts in Mathematics, vol. 288. Springer, Cham (2021).
63. von zur Gathen J., Gerhard J.: Modern Computer Algebra, third edition. Cambridge University Press, Cambridge (2013).
64. Vélu J.: Isogénies entre courbes elliptiques. Compt. Rend. Acad. Sci. **273**, 238–241 (1971).
65. Wesolowski B.: The supersingular isogeny path and endomorphism ring problems are equivalent. In: 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7–10, 2022, pp. 1100–1111. IEEE (2021).
66. Wesolowski B.: Orientations and the supersingular endomorphism ring problem. In: Dunkelman O., Dziembowski S. (eds.) Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Con-

ference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part III, volume 13277 of Lecture Notes in Computer Science, pp. 345–371. Springer (2022).

Authors and Affiliations

Sarah Arpin[1] · James Clements[2] · Pierrick Dartois[3] · Jonathan Komada Eriksen[4] · Péter Kutas[5,6] · Benjamin Wesolowski[7]

✉ Sarah Arpin
s.a.arpin@math.leidenuniv.nl

James Clements
james.clements@bristol.ac.uk

Pierrick Dartois
pierrick.dartois@u-bordeaux.fr

Jonathan Komada Eriksen
jonathan.k.eriksen@ntnu.no

Péter Kutas
p.kutas@bham.ac.uk

Benjamin Wesolowski
benjamin.wesolowski@math.u-bordeaux.fr

1 Mathematics Institute, Universiteit Leiden, Leiden, The Netherlands

2 School of Computer Science, University of Bristol, Bristol, UK

3 Centre Inria de l'Université de Bordeaux, Institut de Mathématiques de Bordeaux, UMR 5251, Bordeaux, France

4 Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Trondheim, Norway

5 Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

6 Hungary and School of Computer Science, University of Birmingham, Birmingham, UK

7 ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France