ORIENTEERING WITH ONE ENDOMORPHISM

SARAH ARPIN, MINGJIE CHEN, KRISTIN E. LAUTER, RENATE SCHEIDLER, KATHERINE E. STANGE, HA T. N. TRAN

ABSTRACT. In supersingular isogeny-based cryptography, the path-finding problem reduces to the endomorphism ring problem. Can path-finding be reduced to knowing just one endomorphism? It is known that a small endomorphism enables polynomial-time path-finding and endomorphism ring computation [36]. As this paper neared completion, it was shown that the endomorphism ring problem in the presence of one known endomorphism reduces to a vectorization problem [54]. In this paper, we give explicit classical and quantum algorithms for path-finding to an initial curve using the knowledge of one endomorphism. An endomorphism gives an explicit orientation of a supersingular elliptic curve. We use the theory of oriented supersingular isogeny graphs and algorithms for taking ascending/descending/horizontal steps on such graphs. Although the most general runtimes are subexponential, we show that every supersingular elliptic curve has (potentially large) endomorphisms whose exposure would lead to a classical polynomial-time path-finding algorithm.

1. INTRODUCTION

The security of isogeny-based cryptosystems depends upon a constellation of hard problems. Central are the path-finding problem (to find a path between two specified elliptic curves in a supersingular ℓ-isogeny graph), and the endomorphism ring problem (to compute the endomorphism ring of a supersingular elliptic curve). Only exponential algorithms are known for general path-finding, in the absence of information beyond the j-invariants you wish to navigate between. However, if the endomorphism rings are known, the KLPT algorithm allows for polynomial-time pathfinding [31]. In fact, it is known that the pathfinding and endomorphism ring problems are equivalent [22, 55].

A natural question to ask is whether knowledge of a single explicit endomorphism (which generates only a rank 2 subring of the rank 4 endomorphism ring) can be used for path-finding. Answering this question is the goal of this paper: we give explicit algorithms transforming knowledge of one endomorphism into a way-finding tool that can detect ascending, descending and horizontal directions with regards to the corresponding orientation, and use this to walk to $j = 1728$.

The question of the security of one endomorphism has recently been 'in the air,' for example, with the uber isogeny assumption of [19]. Knowledge of a small explicit endomorphism is known to be a weakness [36]. As this paper was being completed, a related study was also made available [54]; see Section 1.3 for a comparison with this paper.

By *explicit endomorphism*, we mean one given in some form in which its action on the curve is computable, and its minimal polynomial is known (but note that, given an endomorphism, both its norm and trace are in many cases computable; see Section 2.2). For example, such an endomorphism may be given as a rational map, or a composition chain of rational maps, and these are the two cases we focus on in this paper. The data of such an endomorphism is equivalent to the data of an *orientation* of the supersingular elliptic curve, namely a map $\iota : K \to \mathbb{Q} \otimes_{\mathbb{Z}} \mathrm{End}(E)$, where K is the imaginary quadratic field generated by a root of the minimal polynomial of the endomorphism.

Date: January 26, 2022.

1991 *Mathematics Subject Classification.* Primary: 14G50, 94A60, 11G05, 14K04, 11-04; Secondary: 11R52.

Key words and phrases. elliptic curve, endomorphism ring, supersingular isogeny graph, orientation, path-finding, vectorization.

Katherine E. Stange and Sarah Arpin were supported by NSF-CAREER CNS-1652238. Katherine E. Stange was also supported by Simons Fellowship 822143. Ha T. N. Tran was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (funding RGPIN-2019-04209 and DGECR-2019-00428). R. Scheidler was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (funding RGPIN-2019-04844). Mingjie Chen was supported by NSF grants DMS-1844206, DMS-1802161.

The study of orientations provides some structure to the supersingular isogeny graph, which has recently been exploited [13, 39]. In particular, the ℓ-isogeny graph of *oriented* supersingular elliptic curves over $\overline{\mathbb{F}}_p$ has a volcano structure familiar from the ordinary case. This graph maps onto the supersingular ℓ-isogeny graph over $\overline{\mathbb{F}}_p$. Our approach is to use the orientation provided by a given explicit endomorphism to discern ascending, descending and horizontal directions with regards to the volcano. This provides a sort of tool for 'orienteering.' (The sport of *orienteering* involves finding one's way to checkpoints across varied terrain using only map and compass.)

The core result of our paper is an algorithm that finds an ℓ-isogeny path from a given supersingular elliptic curve E to an initial curve E_{init}, given a single explicit endomorphism of E. We take E_{init} to be the curve with j-invariant $j = 1728$, but other choices are possible (see Section 6.3). The overall plan is as follows. First, climb the oriented volcano from E, oriented by the given endormorphism, to the volcano rim (using the given endomorphism as our 'orienteering tool'). Then, by orienting the curve $j = 1728$ with the same field, we can climb to the rim from there also. Finally, we attempt to meet by circling the rim.

This approach is limited by our ability to traverse a potentially large segment of the rim, or to hit the same rim in a large cordillera of volcanoes, whose size is generally equal to the class number of the corresponding quadratic order. If we simply walk the rim, then, classically, the runtime depends linearly on this class number, which can be expected to be exponential in $\log p$. However, using a quantum computer to solve the vectorization problem (see Section 9.1) yields a subexponential algorithm. Even in the classical case, we show that a large endomorphism which nevertheless walks us to a rim of small class number introduces a vulnerability to isogeny based cryptosystems.

1.1. **Main theorems.** We rely on a number of heuristic assumptions: (i) The Generalized Riemann Hypothesis (hereafter referred to as GRH). (ii) Powersmoothness in a quadratic sequence or form is as for random integers (a powersmooth analogue of the heuristic assumption underlying the quadratic sieve; see Heuristics 5.10 and 9.2). (iii) The orientations of a fixed j-invariant are distributed reasonably across all suitable volcanoes (Heuristic 3.8). (iv) This distribution is independent of a certain integer factorization (Heuristic 6.7). (v) The aforementioned integer factorization is prime with the same probability as a random integer (Heuristic 6.4; this heuristic is similar to those used in [21] and [31]).

We state our main results, whose proofs can be found in Section 10.1. We use the notation $L_x(y) = \exp(O((\log x)^y(\log\log x)^{1-y}))$. Our first theorem gives a classical algorithm for ℓ-isogeny pathfinding that is subexponential in $\log p$ times a certain class number, for a wide range of input endomorphisms.

Theorem 1.1. *Choose a small prime ℓ and assume the heuristic assumptions*[1] *given above. Let $\theta \in \operatorname{End}(E)$ be an endomorphism of degree d, such that that $L_d(1/2) \geq \operatorname{poly}(\log p)$. Suppose θ can be evaluated on points $P \in E(\mathbb{F}_{p^k})$ in time $T_\theta(k, p)$. Let Δ' be the ℓ-fundamental part of the discriminant Δ of θ (obtained*[2] *by removing the largest even power of ℓ), and assume that $|\Delta'| \leq p^2$. Let $h_{\Delta'}$ be the class number of the quadratic order of discriminant Δ'. Then there is a classical algorithm that finds an ℓ-isogeny path of length $O(\log p + h_{\Delta'})$ from E to the curve E_{init} of j-invariant $j = 1728$ in runtime $T_\theta(L_d(1/2), p) + h_{\Delta'} L_d(1/2)$.*

Note that the runtime depends on the class number $h_{\Delta'}$ which can be significantly smaller than h_Δ. This allows for $\operatorname{poly}(\log p)$ time algorithms for some large endomorphisms, which we discuss in a moment.

Note also that the point evaluation condition on θ is for generality. Any θ which is represented in terms of rational maps has $T_\theta(k, p) = \operatorname{poly}(d, k, \log p)$, hence the final runtime would be $h_{\Delta'} \operatorname{poly}(d \log p)$. But θ could be represented as a composition chain of isogenies in such a way that $T_\theta(k, p)$ is subexponential in d, leading to a runtime of $h_{\Delta'} L_d(1/2) \operatorname{poly}(\log p)$.

Furthermore, we have a polynomial-time algorithm if the endomorphism has small degree, or even just small discriminant (Theorem 10.1); the cryptographic weakness caused by such endomorphisms is already known by other methods [36]. There are also some large endomorphisms which are insecure, in the sense that they admit polynomial-time algorithms if they can be evaluated in polynomial time. Specifically, modifications of the algorithm lead to special cases:

(1) If ℓ is inert in the field associated to Δ, the runtime improves for endomorphisms in suitable form to $L_d(1/2) + h_{\Delta'} \operatorname{poly}(\log p)$, and the path-length is improved to $O(\log p)$ (Proposition 8.1).

[1] See Proposition 8.1 for the exact subset of heuristics needed.

[2] Except when $\ell = 2$, if $\Delta = 2^{2k}\Delta''$ where $4 \nmid \Delta''$ and $\Delta'' \equiv 2, 3 \pmod 4$, then we set $\Delta' := 4\Delta''$.

(2) If, in addition to the above, $\Delta' = \Delta$, then the runtime improves further to $h_{\Delta'}\,\mathrm{poly}(\log p)$ (Proposition 8.1).
(3) If the norm of the endomorphism has $B(p)$-powersmooth factorization and its discriminant is coprime to ℓ, then the runtime improves to $h_{\Delta'}\,\mathrm{poly}(B(p)\log p)$ (Theorem 10.4).
(4) If the input endomorphism is of size $\mathrm{poly}(\log p)$ (in trace, norm and discriminant), then the runtime improves to $\mathrm{poly}(\log p)$ (Theorem 10.1) (these endomorphisms were already known to present a security risk [36]).
(5) If norm and discriminant have suitable factorizations, then the runtime can improve to $\mathrm{poly}(\log p)$ even for non-small endomorphisms (Theorem 10.2). This shows that **there are large insecure endomorphisms** (Corollary 10.3) (to our knowledge, this is the first time this has been demonstrated, although it is possible to deduce this using similar methods from [54]; see Section 1.3).

A corollary to Theorem 1.1 is that these insecure large endomorphisms exist for every supersingular curve. We state an informal version here.

Corollary 1.2 (Corollary 10.3)**.** *Under the same heuristic assumptions as before, every supersingular curve admits an endomorphism which can be revealed in polynomial space in a form that allows for polynomial-time evaluation, and gives rise to a classical algorithm to walk to E_{init} in $\mathrm{poly}(\log p)$ time.*

The classical algorithm of Theorem 1.1 first transforms the input endomorphism to a powersmooth isogeny chain, which to our knowledge is the most efficient type of representation. However, we have endeavoured to write our component algorithms to handle an abstract notion of an input endomorphism offering certain functionalities (Section 5.1), in anticipation of their potential application to different types of endomorphism representations.

Our second theorem gives a quantum algorithm for finding a smooth isogeny to an initial curve that runs in subexponential time in the discriminant of the input endomorphism, plus factors depending on the evaluation time of the endomorphism.

Theorem 1.3. *Assume the heuristic assumptions given above. Let $\theta \in \mathrm{End}(E)$ be an endomorphism which can be evaluated on points $P \in E(\mathbb{F}_{p^k})$ in time $T_\theta(k,p) \geq \mathrm{poly}(k \log p)$. Suppose θ has discriminant Δ coprime to p with $|\Delta| \leq p^2$. Let $d = \max\{\deg\theta, |\Delta|\}$. Then there is a quantum algorithm that finds an $L_{|\Delta|}(1/2)$-smooth isogeny of norm $O(\sqrt{|\Delta|})$ from E to $j = 1728$ in runtime $T_\theta(\log d, p)L_{|\Delta|}(1/2)$.*

In both theorems, one may use other suitable initial curves besides $j = 1728$; see Section 6.3.

1.2. **Other algorithms presented.** Some of the explicit building blocks of the results above may have independent applications. In particular, we provide algorithms for the following tasks, among others:

(1) Section 4 provides methods for detecting ascending, descending and horizontal directions in general.
(2) Section 5.3 presents a technique for obtaining a prime-power powersmooth isogeny chain endomorphism from the same quadratic order as a given endomorphism (Algorithm 5.3).
(3) Section 6 discusses an algorithm which computes an orientation on the elliptic curve of j-invariant 1728 (or other suitable curves; see Section 6.3) by an ℓ-power multiple of a given discriminant (Algorithm 6.1). In other words, given a quadratic order $\mathcal{O}$, it finds $j = 1728$ somewhere in the cordillera of an order containing $\mathcal{O}$. In fact, it finds arbitrarily many such orientations, moving gradually further 'down' the volcanoes. This algorithm runs in heuristic polynomial time when the discriminant is coprime to p and less than p^2 in absolute value.
(4) Section 7.2 concerns a method for computing the class group action of $\mathrm{Cl}(\mathcal{O})$ on $\mathrm{SS}_{\mathcal{O}}$, the set of curves primitively oriented by $\mathcal{O}$. In fact, we demonstrate how to navigate $\mathrm{SS}_{\mathcal{O}}$ using the class group action of $\mathrm{Cl}(\mathcal{O}')$ for any $\mathcal{O}' \subseteq \mathcal{O}$.
(5) Section 9 provides quantum algorithms for vectorization on an oriented volcano rim (Proposition 9.3; prior work includes [9, Section 6.1], [54, Proposition 4]), and for determining the quadratic order for which a given orientation is primitive (Proposition 9.7).
(6) Section 11 contains an efficient algorithm for dividing an isogeny by $[\ell]$ (Algorithm 11.2), originally outlined by McMurdy, which is more efficient than naïve algorithms for this task. We make McMurdy's approach explicit for arbitrary ℓ (he only made explicit the case $\ell = 2$, which is more straightforward.).

1.3. **Comparison with [54].** The only other work that pertains to path-finding algorithms using an orientation is found in the excellent article [54], which covers a web of reductions between a wide variety of hard problems related to orientations, and appeared as this paper was nearing completion. That work is largely concerned with theoretical complexity reductions, although one can derive classical and quantum pathfinding algorithms from these reductions, for an abstract class of orientations (see item (4)). By contrast, in this article we focus on explicit algorithms, runtimes, and endomorphism representations, as well as numerical examples. However, it is possible to compare the region of overlap between the two articles, which is *runtimes for classical and quantum path-finding in the presence of one endomorphism.* To do so, there are several important points about the method of comparison:

(1) The paper [54] actually provides reductions from the endomorphism ring problem, which is known to be polynomially equivalent to the path-finding problem. We will ignore this distinction.
(2) The paper [54] solves the endomorphism ring problem by reducing it to the vectorization problem and solving that by the best known classical or quantum algorithms. Our algorithms can't strictly be interpreted as reductions to the vectorization problem. For example, in the classical Algorithm 8.1, we attempt to relate two oriented curves without knowing their common class group orbit (see Remark 8.2). In the quantum case, we use the quantum computer twice: once for vectorization (Section 9.1), and once to determine the order for which an orientation is primitive (Section 9.2).
(3) The paper [54] uses methods largely contained in the theory of quaternion algebras, overlapping very little with our methods.
(4) The paper [54] applies to an abstract class of *efficiently representable endomorphisms*, and provides reductions which are polynomial in the length of the representation. The definition permits endomorphisms which are exponentially large. In our paper we discuss explicit representations and their concrete practical efficiency (Section 5), and our algorithm runtimes take the conversion of arbitrary endomorphisms into suitable representations into account. Therefore, in order to compare, we will assume that input endomorphisms are in powersmooth prime-power isogeny chain form (see Section 5.3). To change into such a form can incur a subexponential runtime, depending on the form of the input endomorphism (Algorithm 5.3).

For the above reasons, we compare only runtime statements. Overall, the runtimes implied for pathfinding in the presence of an endomorphism are reassuringly similar between the two papers.

(1) The paper [54] assumes the stronger hypothesis that the discriminant of the input endomorphism has a known factorization. We do not assume this. Although the reduction to vectorization in [54] requires a factorization, in practice vectorization is more difficult than factorization, so this does not affect the runtime comparison.
(2) In contrast to our work, the work [54] is not heuristic beyond a dependence on GRH and the solution to the vectorization problem ([54, Proposition 4]). We plan to address some of our heuristics in a follow-up paper [2].
(3) Comparing the classical algorithm of Theorem 1.1, namely Algorithm 8.1, with the algorithm implied by [54, Proposition 7, Section 3 Subsection 'Computing the action', Theorem 4], we obtain similar runtimes, with the following distinctions. Both algorithms depend polynomially on the size of the representation of the endomorphism. In the case that $\Delta = \Delta'$ (the endomorphism is already at the rim), both algorithms depend on the class number h_Δ; ours linearly, and [54] in square root. In the case that $\Delta \neq \Delta'$, both depend instead on the smaller class number $h_{\Delta'}$, but ours depends on the smoothness bound of the relative conductor, while that of [54] depends upon the powersmoothness bound of the relative conductor. (We ascend volcanoes, so that our relative conductors are typically ℓ-power, but one can also alternate choices of ℓ; see the proof of Theorem 10.2 for a discussion.)
(4) Continuing the comparison of item (3), our classical algorithm directly produces a path whose length depends on the class number (since it traverses a volcano rim). A reduction to the vectorization problem as in the algorithm implied in [54] produces a path of $\mathrm{poly}(\log p)$ length, by solving the vectorization problem to find a smooth isogeny, and then, by an equivalence implied in [55], tranforming that into an ℓ-isogeny. See Remark 8.2.
(5) For every curve, we show that certain large degree endomorphisms can be expressed in $\mathrm{poly}(\log p)$ space and admit a classical path-finding algorithm in $\mathrm{poly}(\log p)$ time (Corollary 10.3). In fact, these

same endomorphisms would be susceptible to the methods of [54], although this implication is not considered there.

(6) Comparing the quantum algorithm of Theorem 1.3, namely Algorithm 9.1, with the algorithm implied by [54, Proposition 4, Proposition 7], one obtains a similar runtime, depending subexponentially on the discriminant of the input endomorphism.

(7) Finally, [54, Proposition 6] describes a probabilistic polynomial-time algorithm for computing a primitive orientation of an elliptic curve by some quadratic order of discriminant Δ. However, that algorithm only applies to orders with $|\Delta| < 2\sqrt{p} - 1$ and relies on lattice reduction to find the smallest element in the order. Our method works for $|\Delta| < p^2$ and finds orientations further 'down the volcano,' but is presented for $j = 1728$ only (but generalizes to other initial curves with good endomorphism rings in the sense of Section 6.3).

1.4. **Other contributions.** We give careful runtime analyses for various tasks related to endomorphisms represented as rational functions or as composition chains of isogenies, including evaluation, translation, division-by-$[\ell]$, and Waterhouse twisting. Additionally, we provide a review and some modest extensions to the theory of orientations as described in [39]; see Section 3, in particular Section 3.3.

In a follow-up paper [2], we establish a theoretical bijection between volcano rims and cycles in the ℓ-isogeny graph, and address some of the aforementioned heuristics for oriented supersingular ℓ-isogeny graphs used in this paper.

Throughout the paper we demonstrate our algorithms with a running example first introduced in Example 3.2. The examples are given in more detail in SageMath [49] worksheets with accompanying PDF details, available on GitHub [3].

1.5. **Outline.** In Section 2, we set some notations and conventions and also state a few runtime lemmata. In Section 3, we introduce the main object of study, namely oriented ℓ-isogeny graphs and their properties, including some heuristic behaviour. In Section 4, the relationship between an endomorphism and an orientation is explained, and we also introduce a few new definitions that aid in navigating the oriented ℓ-isogeny graph. In Section 5, we discuss the representation of endomorphisms, along with the basic functionalities for these representations required for later algorithms. We then compute orientations for the supersingular elliptic curve of j-invariant 1728 in Section 6. In Sections 7-9, we present our algorithms and give detailed runtime analyses and examples. In Section 10, we discuss the proofs of our main theorems as well as some special cases. Lastly, we leave to Section 11 the technical explanation of McMurdy's division-by-ℓ algorithm and provide its runtime analysis. Throughout the paper, to aid in reading, important assumptions will be rendered in **bold**.

1.6. **Acknowledgements.** We would like to thank Catalina Camacho-Navarro, Elena Fuchs, Péter Kutas, and Christophe Petit for helpful discussion. We especially thank Benjamin Wesolowski, who took the time to share highly valuable suggestions on an earlier draft, particularly some important corrections concerning Proposition 9.3. We would also like to thank the conference *Women in Numbers 5* for the opportunity to form this research group.

2. Background

2.1. **Notations and conventions.** Throughout the paper, let p **be a cryptographically sized prime** (upon which runtimes will depend), and let ℓ **be a small prime** (whose size will be assumed $O(1)$ for runtimes). In particular, $\ell \neq p$. We will assume **both p and ℓ are defined once throughout the paper** (so, for example, they will not be repeated as an input to every algorithm); the only exception being Section 9.

Every elliptic curve considered in the paper is to be assumed to be a **supersingular curve** over $\overline{\mathbb{F}}_p$. All such curves can be defined over $\mathbb{F}_{p^2}$. Every isogeny and endomorphism is assumed to have domains and codomains which are curves of this type. We use the notation $\mathrm{End}(E)$ for the endomorphism ring of the elliptic curve E over $\overline{\mathbb{F}}_p$, and $\mathrm{End}^0(E) := \mathbb{Q} \otimes_{\mathbb{Z}} \mathrm{End}(E)$ for the endomorphism algebra of E. We use the notation O_E for the identity element of an elliptic curve E, and $j(E)$ for the j-invariant. We use the variables φ and ψ to denote isogenies, while θ is generally reserved for endomorphisms. The dual isogeny to an isogeny φ is denoted by $\widehat{\varphi}$. Let $E^{(p)}$ denote the curve obtained by the action of Frobenius on E (acting on the Weierstrass coefficients). Let $\pi_p : E \to E^{(p)}$ denote the Frobenius isogeny, given by $\pi_p(x, y) = (x^p, y^p)$. Note that Frobenius is an endomorphism if E is defined over $\mathbb{F}_p$. Frobenius also acts on any isogeny $\varphi : E \to E'$ (acting on its coefficients) to give $\varphi^{(p)} : E^{(p)} \to (E')^{(p)}$ of the same degree. Unless otherwise specified (such as Frobenius), **isogenies will be assumed to be separable** throughout the paper (many of the algorithms herein would not apply to inseparable endomorphisms or isogenies).

There is only one fixed supersingular ℓ-isogeny graph under consideration at any time, which we denote simply by $\mathcal{G}$. Namely, this is the graph whose vertices are $\overline{\mathbb{F}}_p$-isomorphism classes of supersingular elliptic curves (which we will often refer to simply by their j-invariants), and whose directed edges are ℓ-isogenies (when there are no extra automorphisms, we can identify dual pairs to create an undirected graph).

We consider imaginary quadratic fields $K = \mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a fundamental discriminant. Then the ring of integers has the form $\mathcal{O}_K = \mathbb{Z}[\omega]$, where

$$\omega = \begin{cases} \frac{1+\sqrt{\Delta}}{2} & \Delta \equiv 1 \pmod 4, \\ \frac{\sqrt{\Delta}}{2} & \Delta \equiv 0 \pmod 4. \end{cases}$$

Since we sometimes have multiple quadratic orders under consideration, we use the notation $(\alpha, \beta)_{\mathcal{O}}$ for the ideal generated by α and β in $\mathcal{O}$. The (possibly non-maximal) orders $\mathcal{O}$ of K are parametrized by a positive integer called the conductor. If $\mathcal{O}$ has conductor f, then $\mathcal{O} = \mathbb{Z}[f\omega]$. If $\ell \nmid f$, then we say that both $\mathcal{O}$ and its discriminant are *ℓ-fundamental*. Given a discriminant Δ, its *ℓ-fundamental part* is the unique ℓ-fundamental discriminant dividing Δ.

Write $B_{p,\infty}$ for the rational quaternion algebra ramified at p and ∞. **Every quadratic field K is assumed to embed in the quaternion algebra $B_{p,\infty}$**, i.e. to be an imaginary quadratic field in which p does not split [51, Proposition 14.6.7(v)]; the only exception is in the discussion of Heuristic 6.4. Every quadratic order $\mathcal{O}$ is assumed to generate such a field K, and to **have discriminant not divisible by** p. Every quadratic discriminant is assumed to be the discriminant of such a quadratic order $\mathcal{O}$, and we write $\Delta_{\mathcal{O}}$. We denote by $\mathcal{O}_K$ the maximal order of the quadratic field K and reserve Δ_K for the discriminant of $\mathcal{O}_K$.

Complex conjugation (which is also the action of $\mathrm{Gal}(K/\mathbb{Q})$) is denoted by an overline: $\alpha \mapsto \overline{\alpha}$. We use the notation $\mathrm{Cl}(\mathcal{O})$ and $h_{\mathcal{O}}$ for the class group and class number, respectively, of a quadratic order $\mathcal{O}$.

The reduced norm and trace of $B_{p,\infty}$ coincide with the norm and trace of an element when it is considered as a quadratic algebraic number; when we discuss norm and trace it is always this we refer to.

For runtime analyses we use big O notation, including soft $\widetilde{O}$ for absorbing log factors. The notation $\mathbf{M}(n)$ will indicate the runtime of field operations (addition, multiplication, inversion) in a finite field of cardinality n; here, we note that $\mathbf{M}(n^k) = O(\mathbf{M}(n))$ when k is constant. In the later portions of the paper we are mainly concerned with the distinction between polynomial, subexponential and exponential algorithms. We write runtime as $\mathrm{poly}(x)$ if there exists a polynomial f so the runtime is $O(f(x))$. When we are concerned only with whether runtime is polynomial, we will suppress the notation $\mathbf{M}$, by assuming that $\mathbf{M}(n) = \mathrm{poly}(\log n)$. For subexponential runtimes, we use notation $L_x(y) = \exp(O((\log x)^y (\log\log x)^{1-y}))$.

For general background on isogeny-based cryptography and supersingular isogeny graphs, we will assume the reader is familiar with a resource such as [22, Section 2] or [18].

2.2. **Runtime lemmata.** In this section, we recall some basic runtimes for isogenies and torsion points, etc. The first lemma is standard.

Lemma 2.1. *Given $P, Q \in E[N]$, and $0 \le a, b < N$, computing $[a]P + [b]Q$ takes time $O((\log N)\boldsymbol{M}(p^{N^2}))$.*

Lemma 2.2 ([4, Corollary 2.5]). *Let $\varphi : E \to E'$ be an isogeny between two supersingular curves, both defined over $\mathbb{F}_{p^2}$. Then φ is defined over $\mathbb{F}_{p^{12}}$. If neither of $j(E)$ or $j(E')$ are 0 or 1728, then φ is defined over $\mathbb{F}_{p^4}$.*

Lemma 2.3. *Let t denote the smallest integer such that $E[N] \subseteq E(\mathbb{F}_{p^t})$. In particular, $t \le N^2 - 1$. Finding a basis of $E[N]$ has runtime $\widetilde{O}(N^4 (\log p)\boldsymbol{M}(p^{N^2}))$.*

Proof. This can be proven by adapting the second paragraph of the proof of Lemma 5 in [25]. In particular, the limiting runtime is the call to [53], which takes time $\widetilde{O}(N^4(\log p)\mathbf{M}(p^{N^2}))$. See also [4, Lemma 6.9]. □

Lemma 2.4. *Consider an isogeny $\varphi : E \to E'$ of degree d, and a point $P \in E(\mathbb{F}_{p^t})$, where $12 \mid t$. Then computing $\varphi(P)$ takes time $O(d\boldsymbol{M}(p^t))$. In particular, if $P \in E[N]$, then the time taken is $O(d\boldsymbol{M}(p^{\mathrm{lcm}(12,N^2)}))$.*

Proof. Write φ as a rational map $\varphi(x,y) = (\varphi_1(x), \varphi_2(x)y)$; here the denominators and numerators of $\varphi_1(x)$ and $\varphi_2(x)$ are polynomials in x of degree at most $3d$. By Lemma 2.2, we can assume that their coefficients are in $\mathbb{F}_{p^{12}} \subseteq \mathbb{F}_{p^t}$. To compute $\varphi(P)$, we apply Horner's algorithm [30, p. 467], which requires $O(d)$ operations in the field. Assume that P is an N-torsion point on E. Then t can be chosen such that $t \leq \operatorname{lcm}(t, N^2)$ by Lemma 2.3. □

In the case that $\varphi = [n]$ for some integer n, it is more efficient to use a standard a double-and-add approach, which will also take polynomial time in the degree.

Lemma 2.5 ([50], [45, Theorem 3.5], [27, Section 5.1])**.** *Vélu's formulas for an isogeny of degree d compute the isogeny in time $\widetilde{O}(d\boldsymbol{M}(p^{d^2}))$.*

By Lemma 2.2, the isogeny created has coefficients in the field $\mathbb{F}_{p^{12}}$.

Lemma 2.6. *Let $\varphi : E \to E'$ and $\psi : E' \to E''$ be isognies represented as rational maps, of respective degrees d and d', where E, E', E'', φ and ψ are defined over some finite field $\mathbb{F}$. Then computing the composition $\psi \circ \varphi : E \to E''$ as a rational map takes time $\widetilde{O}(dd'\boldsymbol{M}(\#\mathbb{F}))$.*

Proof. As usual, write $\varphi = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y\right)$ where $u(x), v(x), s(x), t(x) \in \mathbb{F}[x]$ are polynomials of degree $O(d)$ with $\gcd(u,v) = \gcd(s,t) = 1$. Similarly, write $\psi = \left(\frac{u'(x)}{v'(x)}, \frac{s'(x)}{t'(x)}y\right)$ with analogous conditions on $u'(x)$, $v'(x)$, $s'(x)$, $t'(x) \in \mathbb{F}[x]$. Then

$$\psi \circ \varphi = \left(\frac{u'(\frac{u(x)}{v(x)})}{v'(\frac{u(x)}{v(x)})}, \frac{s'(\frac{u(x)}{v(x)})}{t'(\frac{u(x)}{v(x)})} \frac{s(x)}{t(x)} y \right) .$$

Obtaining $\psi \circ \varphi$ requires computing four compositions of the form $f(\frac{u(x)}{v(x)})$ where $f \in \{u', v', s', t'\}$ has degree $O(d')$. Writing $f(x) = \sum_{i=0}^{n} f_i x^i$ with $n = O(d')$, we have

$$f\left(\frac{u(x)}{v(x)}\right) = \frac{F(u(x), v(x))}{v(x)^n} \quad \text{where} \quad F(x,y) = \sum_{i=0}^{n} f_i x^i y^{n-i} .$$

The computation of $F(u(x), v(x))$ is dominated by computing the powers of $u(x)$ and $v(x)$ which can be accomplished in time $\widetilde{O}(dd'\mathbf{M}(\#\mathbb{F}))$ using fast polynomial multiplication [26]. An alternative way to compute $F(u(x), v(x))$ that is slightly faster but has asymptotically the same runtime is via the Horner-like recursion

$$F_n(x) = f_n \ , \qquad F_{i-1}(x) = f_{i-1} v(x)^{n-i+1} + F_i(x) u(x) \quad (n \geq i \geq 1) \ ,$$

where it is easy to see that $F_0(x) = F(u(x), v(x))$. □

Lemma 2.7. *Let E be an elliptic curve defined over some finite field $\mathbb{F}$, $\theta \in \operatorname{End}(E)$ an endomorphism represented as a rational map, and N an integer. Then computing the endomorphism $\theta + [N] \in \operatorname{End}(E)$ as a rational map takes time $\widetilde{O}(\max\{\deg \theta, N^2\}\boldsymbol{M}(\#\mathbb{F}))$.*

Proof. By [46, Exercise 3.7, pp. 105f.], we have

$$[N](x,y) = \left(\frac{\phi_N(x)}{\psi_N(x)^2}, \frac{\omega_N(x,y)}{\psi_N(x,y)^3} \right) ,$$

where $\phi_N = x\psi_N^2 - \psi_{N+1}\psi_{N-1}$, $\omega_n = (\psi_{N+2}\psi_{N-1}^2 - \psi_{N-2}\psi_{N+1}^2)/4y$ and ψ_n is the n-th division polynomial on E. The required division polynomials have degree $O(N^2)$ and can be computed in $O(\log(N))$ steps using the recursive formulas

$$\psi_{2n+1} = \psi_{n+2}\psi_n^3 - \psi_{n-1}\psi_{n+1}^3 \ , \quad \psi_{2n} = \frac{1}{2y}\psi_n(\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2) \ .$$

Using the point addition formulas on E and fast polynomial multiplication techniques [26], the rational map $\theta + [N]$ can be computed using $\widetilde{O}(\max\{\deg \theta, N^2\})$ operations in $\mathbb{F}$. □

Throughout the paper, we will assume that **all endomorphisms are provided with a trace and norm** (which is the same as degree) that carries through computations; see Section 5.1. If the trace is not provided, then it can be computed using [54, Lemma 1], [22, Lemma 4], [4, Theorem 3.6].

3. Oriented isogeny graphs

In this section, we recall and strengthen basic results about oriented isogeny graphs, mainly based on work of Colò-Kohel [13] and Onuki [39], and provide some minor new extensions of the general theory.

3.1. **Orientations.** Fixing a curve E, we have $\mathrm{End}^0(E) \cong B_{p,\infty}$. The field K embeds into $B_{p,\infty}$ if and only if p does not split in K. There may be many distinct such embeddings. We define a *K-orientation* of an elliptic curve to be an embedding $\iota : K \to \mathrm{End}^0(E)$. If $\mathcal{O}$ is an order of K, then an *$\mathcal{O}$-orientation* is a K-orientation such that $\iota(\mathcal{O}) \subseteq \mathrm{End}(E)$. We say that a K-orientation ι is a *primitive* $\mathcal{O}$-orientation if $\iota(\mathcal{O}) = \mathrm{End}(E) \cap \iota(K)$. It will often be expedient to have a local notion of primitivity: for a prime ℓ, we say that a K-orientation ι is an ℓ-primitive $\mathcal{O}$-orientation if it is an $\mathcal{O}$-orientation and the index $[\mathrm{End}(E) \cap \iota(K) : \iota(\mathcal{O})]$ is coprime to ℓ. In particular, a primitive $\mathcal{O}$-orientation is exactly one which is ℓ-primitive for all primes ℓ.

If $\varphi : E \to E'$ is an isogeny of degree ℓ, where ι is a K-orientation of E, then there is an induced K-orientation $\iota' = \varphi_*(\iota)$ on E' defined $\varphi_*(\iota)(\omega) := \frac{1}{\ell}\varphi \circ \iota(\omega) \circ \widehat{\varphi} \in \mathrm{End}^0(E')$ for any $\alpha \in K$.

3.2. **Oriented isogeny graphs.** A *K-oriented elliptic curve* is a pair (E, ι) where $\iota : K \to \mathrm{End}^0(E)$ is a K-orientation. An isogeny of K-oriented elliptic curves $\varphi : (E, \iota) \to (E', \iota')$ is an isogeny $\varphi : E \to E'$ such that $\iota' = \varphi_*(\iota)$; we call this a K-oriented isogeny and write $\varphi \cdot (E, \iota) = (\varphi(E), \varphi_*(\iota))$. One verifies directly that $\varphi_2 \cdot \varphi_1 \cdot (E, \iota) = (\varphi_2 \circ \varphi_1) \cdot (E, \iota)$. A K-oriented isogeny is a *K-isomorphism* if it is an isomorphism of the underlying curves.

Fixing a quadratic field K, we define the graph $\mathcal{G}_K$ of K-oriented supersingular curves over $\overline{\mathbb{F}}_p$. This is the graph whose vertices are K-isomorphism classes of pairs (E, ι) and which has an edge connecting (E, ι) and (E', ι') for each K-oriented isogeny (defined over $\overline{\mathbb{F}}_p$) of degree ℓ between these oriented curves. If $\varphi : (E, \iota) \to (E', \iota')$ is a K-oriented isogeny, then $\widehat{\varphi} : (E', \iota') \to (E, \iota)$ is also one (since $\widehat{\varphi}_*(\iota') = \widehat{\varphi}_*(\varphi_*(\iota)) = [\ell]_*(\iota) = \iota$). Therefore the edges may be taken to be undirected by pairing isogenies with their duals, when the vertices involved are not $j = 0$ or 1728. Also, isogenies are taken up to equivalence, meaning we quotient by the same isomorphisms as for the vertices; see [39, Definition 4.1]. The graph has (out-)degree $\ell + 1$ at every vertex. (Note that our graph differs slightly from the definition in [39, Section 4], where only the images of curves over a number field with complex multiplication are included; we discuss this distinction in the next section.)

Every K-orientation is a primitive $\mathcal{O}$-orientation for a unique order $\mathcal{O} := \iota(K) \cap \mathrm{End}(E)$. Therefore, the set of vertices of $\mathcal{G}_K$ is stratified by the order $\mathcal{O}$ by which a vertex is primitively oriented.

Definition 3.1. Let $\mathrm{SS}_{\mathcal{O}}$ denote the set of isomorphism classes of K-oriented curves for which the orientation is a primitive $\mathcal{O}$-orientation.

This is a simplification of the notation $\mathrm{SS}^{pr}_{\mathcal{O}}(p)$ found in the literature [39, Section 3] [13, Section 3]. This set is non-empty if and only if p is not split in K and does not divide the conductor of $\mathcal{O}$ [39, Proposition 3.2]. As mentioned in Section 2.1, we make those assumptions throughout the paper.

Let $\varphi : (E, \iota) \to (E', \iota')$ be a K-oriented ℓ-isogeny. Suppose that ι is a primitive $\mathcal{O}$-orientation and ι' is a primitive $\mathcal{O}'$-orientation. There are exactly three possible cases:

(1) $\mathcal{O} = \mathcal{O}'$, in which case we say φ is *horizontal*,
(2) $\mathcal{O} \supsetneq \mathcal{O}'$, in which case $[\mathcal{O} : \mathcal{O}'] = \ell$ and we say φ is *descending*,
(3) $\mathcal{O} \subsetneq \mathcal{O}'$, in which case $[\mathcal{O}' : \mathcal{O}] = \ell$ and we say φ is *ascending*.

Example 3.2 (**Introducing our running example**). To illustrate the algorithms in this paper, we consider supersingular elliptic curves defined over $\overline{\mathbb{F}}_p$ for $p = 179$. As $p \equiv 3 \pmod 4$, the curve $E : y^2 = x^3 - x$ with $j(E) = 1728$ is supersingular. This curve is well-known to have extra automorphisms, and its endomorphism ring is generated by the endomorphisms $[1], [i], \frac{[1]+\pi_p}{2}, \frac{[i]+[i]\circ\pi_p}{2}$, where $[i](x, y) := (-x, iy)$ and π_p is as defined in Section 2.1. We define $K := \mathbb{Q}(\sqrt{-47})$ with $\Delta = -47$ and $\omega = \frac{1+\sqrt{-47}}{2}$. We consider the oriented 2-isogeny graph of supersingular elliptic curves with respect to this imaginary quadratic field K.

3.3. **Frobenius and class group actions.** In this section, we slightly strengthen results of Onuki [39] to give an action on oriented isogenies by a direct product of the class group with Frobenius.

Consider the effect of the Frobenius isogeny on an oriented curve, namely $\pi_p \cdot (E, \iota) = (E^{(p)}, \iota^{(p)})$ where $\iota^{(p)} := (\pi_p)_*(\iota)$. For any isogeny φ, we have $\pi_p \circ \varphi(x, y) = \varphi^{(p)}(x^p, y^p) = \varphi^{(p)} \circ \pi_p(x, y)$. Hence, one

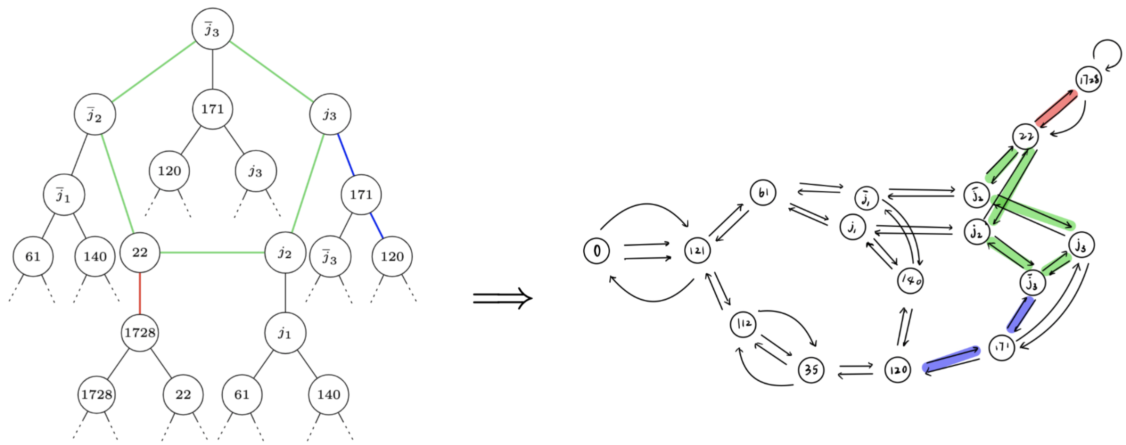

FIGURE 1. On the left hand side is a component of $\mathcal{G}_K$ for $p = 179$, $\ell = 2$ and $K = \mathbb{Q}(\sqrt{-47})$. On the right hand side is the supersingular 2-isogeny graph over $\mathbb{F}_{p^2}$. Let i denote a root of -1 in $\mathbb{F}_{p^2}$, here $j_1 = 64i + 5$, $j_2 = 99i + 107$, $j_3 = 5i + 109$. Since the oriented graph is undirected while the supersingular isogeny graph is directed, we have undirected edges in the left graph and directed edges in the right graph. Note that the green 5-cycle represents the rim of the volcano.

has $(\pi_p)_*(\iota)(\alpha) = \frac{1}{p}\pi_p \circ \iota(\alpha) \circ \widehat{\pi_p} = \frac{1}{p}\iota(\alpha)^{(p)} \circ \pi_p \circ \widehat{\pi_p} = \iota(\alpha)^{(p)}$. Since $\varphi \mapsto \varphi^{(p)}$ gives an isomorphism $\operatorname{End}(E) \cong \operatorname{End}(E^{(p)})$, we see that π_p is horizontal, so this gives an action on $\mathrm{SS}_{\mathcal{O}}$ for any $\mathcal{O}$ by the two-element group $\{1, \pi_p\} = \langle \pi_p \rangle$. In fact, it is an action on the graph, not just the vertices, i.e. it preserves adjacency.

Let $\mathcal{O}$ be a quadratic order of K. Next we define an action of $\mathrm{Cl}(\mathcal{O})$ on $\mathrm{SS}_{\mathcal{O}}$. For an invertible ideal $\mathfrak{a}$ of $\mathcal{O}$ embedded into $\operatorname{End}(E)$ via a K-orientation ι, there exists a horizontal isogeny $\varphi_{\mathfrak{a}}$ defined by the kernel $E[\iota(\mathfrak{a})] := \cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta)$ [39, Proposition 3.5], and we write

$$\mathfrak{a} \cdot (E, \iota) := \varphi_{\mathfrak{a}} \cdot (E, \iota).$$

A different choice of $\varphi_{\mathfrak{a}}$ with the same kernel gives an isomorphic oriented curve [39, Section 3.3], so this is well-defined on the oriented ℓ-isogeny graph.

Proposition 3.3. *The definitions above give a transitive action of* $\mathrm{Cl}(\mathcal{O}) \times \langle \pi_p \rangle$ *on* $\mathrm{SS}_{\mathcal{O}}$ *whose point stabilizers are either all trivial or all* $\langle \pi_p \rangle$*. In particular,* $\#\,\mathrm{SS}_{\mathcal{O}} \in \{h_{\mathcal{O}}, 2h_{\mathcal{O}}\}$*.*

Proof. We have $\pi_p \cdot \varphi_{\mathfrak{a}} \cdot (E, \iota) = (\varphi_{\mathfrak{a}})^{(p)} \cdot \pi_p \cdot (E, \iota)$. To avoid confusion we momentarily use the more specific notation $\varphi_{\mathfrak{a}}^E$ to denote the isogeny $\varphi_{\mathfrak{a}}$ with domain E. Then

$$\begin{aligned} \ker((\varphi_{\mathfrak{a}}^E)^{(p)}) &= \ker(\varphi_{\mathfrak{a}}^{E^{(p)}})^{(p)} = E[\iota(\mathfrak{a})]^{(p)} = \cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta)^{(p)} \\ &= \cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta^{(p)}) = \cap_{\theta \in \iota^{(p)}(\mathfrak{a})} \ker(\theta) = E^{(p)}[\iota^{(p)}(\mathfrak{a})]. \end{aligned} \tag{1}$$

The calculation above implies that $(\varphi_{\mathfrak{a}}^E)^{(p)} = \varphi_{\mathfrak{a}}^{E^{(p)}}$. Thus

$$\pi_p \cdot \mathfrak{a} \cdot (E, \iota) = \mathfrak{a} \cdot \pi_p \cdot (E, \iota). \tag{2}$$

The definition of $\mathfrak{a} \cdot (E, \iota)$ gives a transitive action of $\mathrm{Cl}(\mathcal{O})$ on a subset $\mathrm{SS}'_{\mathcal{O}}$ of $\mathrm{SS}_{\mathcal{O}}$ which contains at least one of (E, ι) or $\pi_p \cdot (E, \iota)$ [39, Theorem 3.4]. In particular, $\mathrm{SS}'_{\mathcal{O}}$ forms one orbit under $\mathrm{Cl}(\mathcal{O})$. But by (2) above, the action is also well defined as an action of classes on all of $\mathrm{SS}_{\mathcal{O}}$. Hence there is a well-defined action of $\mathrm{Cl}(\mathcal{O})$ on $\mathrm{SS}_{\mathcal{O}}$.

The restriction of this action to $\mathrm{Cl}(\mathcal{O})$ acts freely and transitively on a subset of $\mathrm{SS}_{\mathcal{O}}$ which contains at least one of (E, ι) or $(E^{(p)}, \iota^{(p)})$ [39, Theorem 3.4], from which the rest of the statement follows. Transitivity implies that the stabilizers are all of the same size. $\square$

Suppose $\mathcal{O}' \subseteq \mathcal{O}$ are two quadratic orders. Then there is a homomorphism $\rho : \mathrm{Cl}(\mathcal{O}') \to \mathrm{Cl}(\mathcal{O})$. Using the previous proposition, this immediately gives a group action of $\mathrm{Cl}(\mathcal{O}') \times \langle \pi_p \rangle$ on $\mathrm{SS}_{\mathcal{O}}$. It turns out that the explicit form of this action can be computed in the same way as the original action in the following sense.

Proposition 3.4. *Let $\mathcal{O}' \subseteq \mathcal{O}$ with index f. Let $\mathfrak{a}' \in \mathrm{Cl}(\mathcal{O}')$ have norm coprime to f. Suppose that E has a K-orientation ι which is $\mathcal{O}$-primitive. Let $\varphi_{\mathfrak{a}'}$ be defined as the isogeny with kernel $\cap_{\theta \in \iota(\mathfrak{a}')} \ker(\theta)$. Then $\mathfrak{a}' \cdot (E, \iota) = \varphi_{\mathfrak{a}'}(E, \iota)$.*

Proof. Let $\mathfrak{a} := \mathfrak{a}'\mathcal{O}$ be the extension to $\mathcal{O}$. In particular, $\iota(\mathfrak{a}') \subseteq \iota(\mathfrak{a}) \subseteq \mathrm{End}(E)$. We will show $\cap_{\theta \in \iota(\mathfrak{a}')} \ker(\theta) = \cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta)$. From that, we would complete the proof, since

$$\mathfrak{a}' \cdot (E, \iota) = \mathfrak{a} \cdot (E, \iota) = \varphi_{\mathfrak{a}}(E, \iota) = \varphi_{\mathfrak{a}'}(E, \iota).$$

We immediately have $\cap_{\theta \in \iota(\mathfrak{a}')} \ker(\theta) \supseteq \cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta)$. We will show the index between these two groups must divide a power of f. But the larger of the groups has cardinality coprime to f by hypothesis. So this would imply they are equal.

Write $\mathfrak{a}' = \alpha_1 \mathcal{O}' + \alpha_2 \mathcal{O}'$ and $\mathcal{O} = \mathbb{Z} + g\omega\mathbb{Z}$ using the notation of Section 2.1. Then

$$\begin{aligned}
\cap_{\theta \in \iota(\mathfrak{a}')} \ker(\theta) &= \ker(\iota(\alpha_1)) \cap \ker(\iota(\alpha_2)) \cap \ker(\iota(\alpha_1 f g\omega)) \cap \ker(\iota(\alpha_2 f g\omega)),\\
\cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta) &= \ker(\iota(\alpha_1)) \cap \ker(\iota(\alpha_2)) \cap \ker(\iota(\alpha_1 g\omega)) \cap \ker(\iota(\alpha_2 g\omega)).
\end{aligned}$$

We have $\ker(\iota(\alpha_i g\omega)) \subseteq \ker(\iota(\alpha_i f g\omega))$ with index f^2. Thus the index of $\cap_{\theta \in \iota(\mathfrak{a})} \ker(\theta)$ inside $\cap_{\theta \in \iota(\mathfrak{a}')} \ker(\theta)$ must divide a power of f. □

This has the consequence that one need not know $\mathcal{O}$ in order to compute the action of $\mathcal{O}'$ on $\mathrm{SS}_{\mathcal{O}}$.

3.4. **Volcano structure.** Any component of the oriented ℓ-isogeny graph has a *volcano structure* (see Figure 1), which is made precise by the following statement. (This behaviour is similar to the ordinary ℓ-isogeny graph, except here volcanoes have no floor; they descend forever.) Here we remind the reader that $p \neq \ell$ throughout the paper.

Proposition 3.5 ([39, Proposition 4.1]). *Consider a vertex (E, ι) of the oriented ℓ-isogeny graph associated to K, a quadratic field of discriminant Δ. Suppose that ι is a primitive $\mathcal{O}$-orientation for E. If ℓ does not divide the conductor of $\mathcal{O}$, then the following hold.*

(1) There are no ascending edges from (E, ι).
(2) There are $\left(\frac{\Delta}{\ell}\right) + 1$ horizontal edges incident with (E, ι).
(3) There are $\ell - \left(\frac{\Delta}{\ell}\right)$ descending edges from (E, ι).

If ℓ divides the conductor of $\mathcal{O}$, then the following hold.

(1) There is exactly one ascending edge from (E, ι).
(2) The remaining ℓ edges incident with (E, ι) are descending.

Furthermore, it is possible for the descending edges to be multiple, i.e. two descending edges may go to the same vertex. This occurs if and only if the unit group changes cardinality between the two relevant orders [39, Proposition 4.1]. In particular, this phenomenon may only occur if descending from a rim corresponding to the Gaussian or Eisenstein maximal orders, so it is quite limited. Further, by definition, edges which differ in type (ascending, horizontal or descending) cannot have the same oriented codomain.

Proposition 3.5 implies that each connected component of the oriented ℓ-isogeny graph is a *volcano*, containing a *rim* (comprised of the vertices with no ascending edges). From each vertex on the rim a tree radiates infinitely downward. Furthermore, only elements of $\mathrm{SS}_{\mathcal{O}}$ for which $\mathcal{O}$ is ℓ-fundamental can be at a rim. Fixing such an order $\mathcal{O}$, we can define a subgraph of the full K-oriented ℓ-isogeny graph given by those components whose rims consists of (E, ι) with ι a primitive $\mathcal{O}$-orientation. Since the components are volcanoes, we refer to this as the *$\mathcal{O}$-cordillera*. The vertices at the rims are exactly $\mathrm{SS}_{\mathcal{O}}$.

The action of an ideal class $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ gives a permutation on $\mathrm{SS}_{\mathcal{O}}$, which we can visualize as a directed graph. This consists of cycles, all of which are the same size, given by the order of $[\mathfrak{a}]$ in $\mathrm{Cl}(\mathcal{O})$. Applying this to a prime ideal $\mathfrak{l}$ of $\mathcal{O}$ lying above ℓ, the *rims* of the $\mathcal{O}$-cordillera are exactly these cycles. The rims are individually singletons, single– or double-connected pairs, or cycles, and are all of the same size dividing $h_{\mathcal{O}}$. If ℓ is inert, they are each singletons. If ℓ is ramified, they are each of size 2 with one connecting edge (the isogeny and its dual are identified). If ℓ splits into two classes of order 2, we obtain a rim of size two with two connecting edges. Otherwise, the rims are non-trivial cycles in the oriented ℓ-isogeny graph, of size equal to the order of $[\mathfrak{l}] \in \mathrm{Cl}(\mathcal{O})$. We summarize the discussion as follows.

Proposition 3.6. *Let $\mathcal{O}$ be ℓ-fundamental. Let R_ℓ be the order of $[\mathfrak{l}] \in \mathrm{Cl}(\mathcal{O})$, for $\mathfrak{l}$ a prime of $\mathcal{O}$ lying above ℓ. The $\mathcal{O}$-cordillera consists of $\# \mathrm{SS}_{\mathcal{O}} / R_\ell$ volcanoes of rim size R_ℓ.*

3.5. **From oriented isogeny graph to isogeny graph.** There is a graph quotient $\mathcal{G}_K \to \mathcal{G}$ induced by forgetting the orientation.

Proposition 3.7. *Under this quotient, every component of $\mathcal{G}_K$ (i.e. every volcano) covers $\mathcal{G}$.*

Proof. Fix a volcano $\mathcal{V} \subset \mathcal{G}_K$. Choose a vertex $(E, \iota) \in \mathcal{V}$. The image E under the above map lies on $\mathcal{G}$. Since both $\mathcal{V}$ and $\mathcal{G}$ are regular of degree $\ell + 1$ at every vertex, the image of $\mathcal{V}$ must be all of $\mathcal{G}$. ☐

As a corollary, every j-invariant occurs on every volcano infinitely many times. Given p, a result of Kaneko [28, Theorem 2'] implies that the multiple occurrences of a given j-invariant cannot occur too quickly as one descends the oriented ℓ-isogeny volcano. In fact, there is at most one occurrence in the range $|\Delta| < p$ (here Δ is the discriminant corresponding to a certain level in the volcano).

3.6. **Graph statistics and heuristics.** In the ℓ-isogeny graph $\mathcal{G}$, two vertices are at distance d if the shortest path between them in the graph consists of d edges. This is known to be $\leq 2 \log p$ [40, Theorem 1]. In fact, for most pairs of vertices, the distance between them is at most $(1 + \epsilon) \log p$ (see [43, Theorem 1.5] for a precise statement).

We will use the following heuristic to justify the runtimes in the paper. In a follow-up paper [2], we discuss this and some related heuristics in more detail.

Heuristic 3.8. *Let $\mathcal{O}$ be a quadratic order. Consider the finite union $\mathcal{S}$ of $\mathcal{O}'$-cordilleras for all $\mathcal{O}' \supseteq \mathcal{O}$. Fix a j-invariant j_0. Consider the set*

$$\mathcal{J}_{j_0,L} = \{(j_0, \iota) \in \mathcal{S} : \text{appearing at level } \leq L\}.$$

Let $v : \mathcal{J}_{j_0,L} \to \{V : \text{volcano of } \mathcal{S}\}$ be the function taking a vertex to the volcano upon which it lies. Then, as $L \to \infty$, the probability that $v((j_0, \iota)) = V$ for any volcano V is proportional to the number of descending edges from the rim of V.

Briefly, one expects this because a sufficiently long random walk from any rim vertex will visit all vertices with a uniform distribution [25, Theorem 1]. This observation suffices in the case the rims are singletons; other cases should behave similarly.

The following lemma is useful for runtime analyses of our main algorithms (Proposition 8.1 and Proposition 9.8). It states that the Hurwitz class number $H(\mathcal{O})$ (approximately the cardinality of the union of the sets $\mathrm{SS}_{\mathcal{O}}$ involved in $\mathcal{S}$ in Heuristic 3.8) is essentially the same size as the regular class number $h_{\mathcal{O}}$ (approximately the size of the largest $\mathrm{SS}_{\mathcal{O}}$ in the union).

Lemma 3.9. *Let $\mathcal{O}$ be an imaginary quadratic order of discriminant $\Delta < -16$ in some quadratic field K with class number $h_{\mathcal{O}}$ and Hurwitz class number*

$$H_{\mathcal{O}} = \sum_{\mathcal{O} \subseteq \mathcal{O}' \subseteq \mathcal{O}_K} \frac{2}{w'} h_{\mathcal{O}'},$$

where the sum runs over all the quadratic orders $\mathcal{O}'$ containing $\mathcal{O}$ and where $h_{\mathcal{O}'}$ and w' denote the class number and order of the unit group of $\mathcal{O}'$, respectively. Then $H_{\mathcal{O}} \leq h_{\mathcal{O}}(1 + O(1/\log\log|\Delta|))$ as $|\Delta| \to \infty$.

Proof. Let $\mathcal{O}'$ be a quadratic order of discriminant Δ' containing $\mathcal{O}$. Then there exists a positive integer f such that $\Delta = f^2 \Delta'$. By [16, Corollary 7.28], we have

$$h_{\mathcal{O}} = \frac{f h_{\mathcal{O}'}}{w'/w} \prod_{\substack{q \mid f \\ q \text{ prime}}} \left(1 - \left(\frac{\Delta}{q}\right)\frac{1}{q}\right),$$

where $w \in \{2, 4, 6\}$ is the size of the unit group $\mathcal{O}^*$. Thus,

$$h_{\mathcal{O}} \geq \frac{w f h_{\mathcal{O}'}}{w'} \prod_{\substack{q \mid f \\ q \text{ prime}}} \left(1 - \frac{1}{q}\right) = \frac{w}{w'} \varphi(f) h_{\mathcal{O}'},$$

were $\varphi(\cdot)$ denotes Euler's phi function.

The number of quadratic orders $\mathcal{O}'$ containing $\mathcal{O}$ is precisely the number of divisors of f. If follows that

$$H_{\mathcal{O}} \leq \frac{2\sigma(f)}{w\varphi(f)}\, h_{\mathcal{O}} \leq \frac{\sigma(f)}{\varphi(f)}\, h_{\mathcal{O}},$$

where $\sigma(\cdot)$ denotes the sum of divisors function. By [42, Theorem 15] and Robin's Theorem [41], there exist positive constants c_1, c_2 such that

$$\frac{n}{\varphi(n)} > e^{\gamma} \log\log n + \frac{c_1}{\log\log n}\,, \quad \frac{\sigma(n)}{n} < e^{\gamma} \log\log n + \frac{c_2}{\log\log n}$$

for all integers $n \geq 3$, where γ is the Euler-Mascheroni constant. The result now follows. ☐

4. Navigating the K-oriented ℓ-isogeny graph

4.1. **Conjugate orientations and orientations from endomorphisms.** Motivated by our computational goals, we replace the abstract data of an orientation with the more computational data of an endomorphism. Given an element $\theta \in \operatorname{End}(E)$ along with its minimal polynomial $f(x)$, we can infer a unique $\mathbb{Z}[\theta]$-orientation only up to conjugation. Namely, if α is a quadratic irrational root of $f(x)$, then we define $\iota_\theta(\alpha) = \theta$ and extend to a ring homomorphism. The conjugate orientation is defined by $\widehat{\iota_\theta}(\alpha) = \widehat{\theta}$, or equivalently, by $\widehat{\iota_\theta}(\overline{\alpha}) = \theta$. An example in [39, Section 3.1] demonstrates a pair of $\operatorname{Gal}(K/\mathbb{Q})$-conjugate K-oriented curves which are not isomorphic. In other words, given $\varphi \in \operatorname{End}(E)$, one may be in either of two locations in the oriented ℓ-isogeny graph: (E, ι) or $(E, \widehat{\iota})$. However, locally at least, navigating from either location looks the same, in the sense of ascending/descending/horizontal edges and j-invariants.

Lemma 4.1. *The map $(E, \iota) \mapsto (E, \widehat{\iota})$ is a graph isomorphism and an involution, taking $\mathrm{SS}_{\mathcal{O}}$ back to itself for each $\mathcal{O}$. If $\varphi : (E, \iota) \to (E', \iota')$ is a K-oriented ℓ-isogeny, then $\varphi : (E, \widehat{\iota}) \to (E', \widehat{\iota'})$ is a K-oriented ℓ-isogeny, and the type (ascending, descending, or horizontal) is the same.*

Proof. The map is clearly a bijection on vertices. Observe that the dual of $\widehat{\varphi} \circ \iota \circ \varphi$ is $\widehat{\varphi} \circ \widehat{\iota} \circ \varphi$. From this it follows that the map is a graph isomorphism. The observation about type follows from the fact that $\mathrm{SS}_{\mathcal{O}}$ is taken back to itself. ☐

As consequences of this lemma, for two vertices (E, ι) and $(E, \widehat{\iota})$, we have the following:

(1) the j-invariant is the same at both vertices;
(2) both vertices are at the same volcano level;
(3) if the vertices are not at a rim, the ascending isogeny from either vertex is the same;
(4) if the vertices are at the rim, the pair of horizontal isogenies from either vertex is the same;
(5) if we apply any fixed sequence of ℓ-isogenies from both vertices, the sequence of j-invariants appearing on the resulting paths is the same.

For these reasons, it will not, in practice, be necessary for us to know which of two conjugate orientations we are dealing with. Therefore we do not make any choice between the two. In the remainder of the paper, we will not dwell on this distinction, and will work with endomorphisms instead of orientations.

Remark 4.2. It is a natural question to ask when a subset of the four oriented curves (E, ι), $(E^{(p)}, \iota^{(p)})$, $(E, \widehat{\iota})$ and $(E^{(p)}, \widehat{\iota}^{(p)})$ coincide. This question may have importance to a more detailed runtime analysis than we present in this paper, for example. See the thesis of the first author [1].

4.2. **ℓ-primitivity, ℓ-suitability, and direction finding.** Having associated an endomorphism to an orientation, we can now define the following.

Definition 4.3. Let $\theta \in \operatorname{End}(E)$ be an endomorphism and α the corresponding quadratic element (up to conjugation). Then θ (as well as α) is called *ℓ-primitive* if the associated orientations $\iota_\theta : \alpha \mapsto \theta$ and $\widehat{\iota_\theta} : \overline{\alpha} \mapsto \theta$ are ℓ-primitive $\mathbb{Z}[\alpha]$-orientations. Moreover, θ (as well as α) is called *N-suitable*, for an integer N, if α is of the form $f\omega + kN$ where k is some integer, f is the conductor of $\mathbb{Z}[\alpha]$, and $f\omega$ is the generator of $\mathbb{Z}[\alpha]$ as described in the conventions of Section 2.1.

The purpose of this definition is made clear by the following lemma.

Lemma 4.4. *If $\theta \in \operatorname{End}(E)$ is ℓ-suitable, then θ is not ℓ-primitive if and only if $\theta/\ell \in \operatorname{End}(E)$.*

Proof. The endomorphism θ is not ℓ-primitive if and only if there exists a (unique) order $\mathcal{O}' \subseteq \mathrm{End}(E)$ of index $\ell = [\mathcal{O}' : \mathbb{Z}[\theta]]$. But this happens if and only if $\theta/\ell \in \mathrm{End}(E)$, since under the ℓ-suitability hypothesis, $\mathbb{Z}[\theta/\ell]$ is precisely this order $\mathcal{O}'$. □

Lemma 4.5. *Let $\alpha \in O_K \setminus \mathbb{Z}$ with trace t and norm n. Let f be the conductor and Δ_K the fundamental discriminant of $\mathbb{Z}[\alpha]$. Then*

$$\{T \in \mathbb{Z} : \alpha + T \text{ is } N\text{-suitable}\} = \begin{cases} \frac{f-t}{2} + N\mathbb{Z} & \Delta_K \equiv 1 \pmod 4 \\ \frac{-t}{2} + N\mathbb{Z} & \Delta_K \equiv 0 \pmod 4 \end{cases}.$$

In our algorithms, we sometimes choose an optimal T in the sense of the following definition.

Definition 4.6. If $\alpha + T$ has the smallest possible non-negative trace amongst all ℓ-suitable translates of α, we say that $\alpha + T$ is a *minimal ℓ-suitable translate.*

Proposition 4.7. *Suppose $\psi : E \to E'$ is an ℓ-isogeny and $\theta \in \mathrm{End}(E)$ is an ℓ-suitable ℓ-primitive endomorphism. Then*

(1) ψ is ascending if and only if $[\ell]^2 \mid \psi \circ \theta \circ \widehat{\psi}$ in $\mathrm{End}(E')$.
(2) ψ is horizontal if and only if $[\ell] \mid \psi \circ \theta \circ \widehat{\psi}$ but $[\ell]^2 \nmid \psi \circ \theta \circ \widehat{\psi}$ in $\mathrm{End}(E')$.
(3) ψ is descending if and only if $[\ell] \nmid \psi \circ \theta \circ \widehat{\psi}$ in $\mathrm{End}(E')$.

Proof. Let ι' be the induced orientation on E' of $\iota : \alpha \mapsto \theta$ via ψ. Let $\mathcal{O}, \mathcal{O}' \subseteq K$ be two orders such that ι is $\mathcal{O}$-primitive and ι' is $\mathcal{O}'$-primitive. The three cases in the proposition corresponds to the cases when $\mathcal{O} \subsetneq \mathcal{O}'$, $\mathcal{O} = \mathcal{O}'$ and $\mathcal{O} \supsetneq \mathcal{O}'$ respectively. Therefore, ψ is ascending, horizontal and descending correspondingly. □

The previous proposition demonstrates that it is enough to check the action of $\psi\circ\theta\circ\widehat{\psi}$ on $E[\ell]$ to determine whether the isogeny is ascending or descending. However, we can also write down the ascending or horizontal endomorphisms directly by analysing the eigenspaces of θ on $E[\ell]$, as follows. Note that a version of this for Frobenius is used in CSIDH [7] to walk horizontally, earlier used in [29, Section 3.2] and [20, Section 2.3].

Proposition 4.8. *Suppose $\theta \in \mathrm{End}(E)$ is ℓ-suitable and ℓ-primitive. Let $\psi : E \to E'$ be an ℓ-isogeny with kernel $\langle P \rangle \subset E[\ell]$. Then ψ is ascending if and only if $\theta(P) = 0$, and ψ is horizontal if and only if P is an eigenvector of the action of θ on $E[\ell]$ having non-zero eigenvalue. Otherwise ψ is descending.*

Proof. Suppose $\alpha \mapsto \theta$ gives a K-orientation on E, for $K = \mathbb{Q}(\alpha)$. Then for each non-zero eigenvalue $\lambda \in \mathbb{Z}/\ell\mathbb{Z}$ of θ acting on $E[\ell]$, the ideal $\mathfrak{l} := (\alpha - \lambda, \ell)_{\mathcal{O}}$ is an invertible prime ideal above (ℓ) in $\mathcal{O} := \mathbb{Z}[\alpha]$. The isogeny with kernel $E[\mathfrak{l}]$ is horizontal [39, Proposition 3.5] and has kernel $\langle P \rangle$ where $\theta(P) = [\lambda]P$ and $[\ell]P = O_E$. No other ℓ-isogenies are horizontal [39, Proposition 4.1]. (Note that, as usual, [39] only uses the class group action on the image of curves over number fields with CM, but by the more general action including Frobenius described in Proposition 3.3, it holds in our case also.)

Next, suppose that $\lambda = 0$. Then $\mathfrak{l} := (\alpha, \ell)_{\mathcal{O}}$ is a non-invertible ideal, and the corresponding ideal action is ascending [39, Proposition 3.5]. In this case $E[\mathfrak{l}] = \langle P \rangle$ where $[\ell]P = O_E$ and $\theta(P) = 0$. There is only one ascending isogeny [39, Proposition 4.1]. □

5. Representing orientations and endomorphisms

5.1. **Representations and functionality.** We remind the reader that throughout the paper, isogenies and endomorphisms will be assumed separable unless otherwise stated (see Section 2.1). In this section, we discuss two types of representations of an endomorphism. The first is the most basic.

Definition 5.1. A *rationally represented isogeny* is an isogeny given by a rational map. A *rationally represented endomorphism* is an endomorphism which is rationally represented as an isogeny.

We may also represent endomorphisms of large degree (e.g. not polynomial in $\log p$) by writing them as a chain of isogenies of manageable degree.

Definition 5.2. An *isogeny chain isogeny* $\varphi : E_0 \to E_k$ is an isogeny which is given in the form of a sequence of rationally represented isogenies $(\varphi_i : E_{i-1} \to E_i)_{i=1}^k$ which compose to φ, i.e. $\varphi_k \circ \varphi_{k-1} \circ \cdots \circ \varphi_2 \circ \varphi_1 = \varphi$.

Let $B > 0$. Recall that an integer is called *B-smooth* (or *B-friable*) if its largest prime factor is less than B. It is called *B-powersmooth* (or *B-ultrafriable*) if its largest prime power factor is less than B. In order to handle isogeny chain endomorphisms, we will generally *refactor* them, meaning we will replace the chain with another chain representing the same endomorphism, but whose component isogenies have coprime prime power degrees. Moreover, we also fix a powersmooth bound B for the prime power degrees. In Section 5.3.4, we explain our choice of B for the best algorithm runtime.

Definition 5.3. An isogeny chain whose component isogenies have coprime prime power degrees is called a *prime-power* isogeny chain. Moreover, it is called a *B-powersmooth prime-power* isogeny chain if its component isogenies have coprime prime power degrees less than B.

For isogenies represented in any manner, we will need the following functionality:

(1) **Evaluation at ℓ-torsion**: Given $\theta \in \mathrm{End}(E)$, and $P \in E[\ell]$, compute $\theta(P) \in E[\ell]$. (See Lemma 2.4.)
(2) **ℓ-suitable translation**: Given $\theta \in \mathrm{End}(E)$, compute $\theta + [t] \in \mathrm{End}(E)$, for some $t \in \mathbb{Z}$, so that $\theta + [t]$ is ℓ-suitable (Definition 4.3) and again separable. (See Lemma 2.7 for rational representations and Algorithm 5.3 for isogeny chains.) Note that for powersmooth prime power isogeny chains, by computing an ℓ-suitable translation, we always mean that we compute a translate that is a B-powersmooth prime power isogeny chain unless otherwise specified. This is exactly what Algorithm 5.3 does.
(3) **Division by ℓ**: Given $\theta \in \mathrm{End}(E)$ such that $\theta = [\ell] \circ \theta'$, compute $\theta' \in \mathrm{End}(E)$. (See Algorithm 11.2 for rational representations and Algorithm 5.2 for isogeny chains.)
(4) **Waterhouse twisting**: Given $\theta \in \mathrm{End}(E)$ and $\varphi : E \to E'$ an ℓ-isogeny, compute $\varphi \circ \theta \circ \widehat{\varphi} \in \mathrm{End}(E')$. (See Lemma 2.6 for rational representations and Algorithm 5.1 for isogeny chains.)

We have endeavoured to write the paper in a modular fashion, so that these two types of representations — or another unforeseen type of representation, as long as it provides these functionalities — can be used at will. In particular, we write our algorithms (Sections 7.1 onwards) in terms of these functionalities (writing for example $\theta \leftarrow \theta/[\ell]$ for division by ℓ, to be implemented according to the endomorphism representation chosen).

Although isogeny chain endomorphisms may have large degree, we assume that for any type of endomorphism representation, **the overall degree, trace and discriminant are polynomially bounded in p**.

As discussed in Section 2.2, it can be rather involved to compute the trace of an endomorphism. However, the manipulations we perform in our algorithms transform the trace predictably. Therefore, it is to our advantage to attach the trace data to all endomorphisms under consideration and update it as needed. For either rationally represented or isogeny chain endomorphisms, our data type will be the following.

Definition 5.4. A *traced endomorphism* is a tuple of data (E, θ, t, n) where $\theta \in \mathrm{End}(E)$ is either rationally represented or an isogeny chain, and t and n are the reduced trace and norm (degree) of θ, respectively.

5.2. **Functionality for rationally represented endomorphisms.** In the case of a rationally represented endomorphism, we can evaluate at ℓ-torsion directly (Lemma 2.4). We can translate by an integer by adding the rational maps under the group law (Lemma 2.7). We can Waterhouse twist by composing the maps (Lemma 2.6). However, division by ℓ requires a dedicated algorithm. In Section 11, we describe the algorithm of McMurdy [38] for exactly this purpose, and analyse its runtime in greater detail. For the completeness of this section, we record here that the runtime of dividing an isogeny $\varphi : E_1 \to E_2$ of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ (Algorithm 11.2) is $O(\deg^2(\varphi)\mathbf{M}(p))$.

5.3. **Functionality for isogeny chain endomorphisms.** An isogeny chain representation of an endomorphism can be more space efficient than its rational representation, and more efficient to compute with. Computing the Waterhouse twist of an isogeny chain endomorphism is essentially trivial: include the twisting isogenies in the chain. To evaluate at ℓ-torsion, we evaluate the sequence of maps one-by-one (Lemma 2.4); the runtime depends polynomially on the largest degree of their component isogenies.

In this section, we give algorithms for the more onerous tasks of division-by-ℓ and translation by integers. Their runtimes will depend polynomially on the largest prime power appearing in the degree of the endomorphism, which must therefore be kept small for efficiency. To address this problem, which arises when translating to something ℓ-suitable, we use a search step to find a translate of powersmooth degree.

In order to keep the largest prime power in the degree below a certain bound, we will be interested in B-powersmooth prime power isogeny chains. In the last subsection of this section, we balance the runtime considerations by choosing a subexponential powersmoothness bound B for the degree of an isogeny chain endomorphism. Thus, working with a general such endomorphism is a subexponential endeavour.

Although our concern is with endomorphisms, both Algorithm 5.1 and Algorithm 5.2 would work for an isogeny in general.

5.3.1. *Refactoring an isogeny chain.* If an endomorphism is not in the prime power isogeny chain form, we can refactor it. To achieve this, one factors the degree, then builds the new chain from scratch kernel-by-kernel, as described in Algorithm 5.1. In fact, any endomorphism that can be evaluated at arbitrary points on the curve can be converted to an isogeny chain representation using this algorithm.

Remark 5.5. In principle, it is possible to refactor into degrees that are primes as opposed to prime powers. However, this doesn't circumvent the need for powersmoothness (in practice, it would provide some savings, e.g. in Vélu's formulas, but it wouldn't avoid the overall polynomial dependence on the powersmoothness bound). During refactoring, for any prime power factor q^k of the degree, the endomorphism needs to be evaluated on the q^k-torsion, which should therefore be defined over a field of manageable size. See [8, Section 5.2.1] for a nice discussion of this issue in another context.

Algorithm 5.1: Refactoring an isogeny chain

Input: A traced endomorphism (E, θ, t, n) in any form in which it can be evaluated (such as rationally represented or a translation of an isogeny chain), of degree coprime to p.

Output: The same traced endomorphism $(E, \theta, t, n) \in \mathrm{End}(E)$ in prime-power isogeny chain form.

$H \leftarrow []$

$E_0 \leftarrow E$

Write $n = \prod_{j=0}^{u} q_j^{k_j}$ by factoring.

For $j = 0, \ldots, u$ **do**

  Compute a basis for $E[q_j^{k_j}]$.

  Compute $G_j = \ker(\theta) \cap E[q_j^{k_j}]$ by evaluating θ on $E[q_j^{k_j}]$.

  Compute a rationally represented isogeny $\varphi_j : E_j \to E_{j+1}$ given by the kernel $\varphi_{j-1} \circ \ldots \circ \varphi_0(G_j)$, using Velu's formulas.

  Append $(\varphi_j : E_j \to E_{j+1})$ to H.

Return (E, θ, t, n) where θ is given by the isogeny chain H.

Proposition 5.6. *Let B be the largest prime power dividing $\deg \theta$. Then Algorithm 5.1 is correct and has runtime $O(\log \deg \theta)$ times the maximum of the following three runtimes: $O(B^2(\log p))$, $O(B^2(\log B)\boldsymbol{M}(p^{B^2}))$ and the runtime of evaluation of θ on $O(B)$-torsion, and space requirement of $O(B^2 \log p)$. In particular, if θ is an integer translate of an isogeny chain with B-powersmooth degree, then the runtime is $O((\log \deg \theta) B^2 \boldsymbol{M}(p^{B^2}))$.*

Proof. The **For** loop builds an isogeny chain for θ. One can see this by induction: assuming $\theta = \nu' \circ \nu$ where $\nu := \varphi_{j-1} \circ \ldots \circ \varphi_0$, we have by construction that $\nu(G_j)$ vanishes under ν'. Hence θ factors through $\varphi_j \circ \nu$.

To write the factorization of n is at worst $O(B \log^2 B)$ in time (by trial division), but $O(\log n)$ in space. For each prime power factor (so at most $\log n$ times), we must do each of the following: (i) Compute a basis for the torsion subgroup in time and space $O(B^2 \log p)$ by Lemma 2.3. (ii) Evaluate θ on the basis (iii) List the elements of the kernel G_j; this involves computing all linear combinations of the basis images and recording those combinations which vanish; and then computing the corresponding linear combinations of the original torsion points, a total of $B^2 + B$ linear combinations; by Lemma 2.1, this takes time $O(B^2(\log B)\mathbf{M}(p^{B^2}))$. (iv) Apply Vélu's formulas in time $O(B\mathbf{M}(p^{B^2}))$ by Lemma 2.5. Writing down the resulting isogeny takes $O(B)$ coefficients in a subfield of $\mathbb{F}_{p^{12}}$ (Lemma 2.2), hence we use $O(B \log p)$ space for each isogeny of the chain.

If θ is a translate of an isogeny chain whose component degrees are bounded by B, we can further estimate the time taken to evaluate θ on the torsion basis. This involves one evaluation for each component isogeny (at most $\log n$ such). Each evaluation of a component φ_i takes time $O((\deg \varphi_i)\mathbf{M}(p^{B^2}))$ by Lemma 2.4. (Evaluation of the integer translation is of smaller runtime by Lemma 2.1; since the integer is taken modulo the torsion, its size is irrelevant.) □

Remark 5.7. The exponent of the dependence on B can surely be improved here; for example, if B is prime, then our bound on the number of linear combinations on which to evaluate θ is a substantial overestimate.

5.3.2. *Division by ℓ.* In this section, we demonstrate in Algorithm 5.2 how to divide an isogeny chain endomorphism by $[\ell]$.

Algorithm 5.2: Dividing-by-$[\ell]$ for an endomorphism given as a prime-power isogeny chain.

Input: A traced endomorphism (E, θ, t, n) in prime-power isogeny chain form , such that $\theta(E[\ell]) = \{O_E\}$.

Output: A traced endomorphism $(E, \theta', t', n') \in \mathrm{End}(E)$ such that $\theta = [\ell] \circ \theta'$, in prime-power isogeny chain form.

$i \leftarrow$ the index at which the chain has ℓ-power degree.
Modify the chain for θ by replacing φ_i with $\varphi_i/[\ell]$ using Algorithm 11.2.
$t \leftarrow t/\ell$
$n \leftarrow n/\ell^2$.
Return (E, θ, t, n).

Proposition 5.8. *Let B be an upper bound on the degrees of the prime powers in θ. Then Algorithm 5.2 is correct and runs in time $O(B^2 \operatorname{poly}(\log p))$.*

Proof. The runtime is negligible except for the call to Algorithm 11.2. By Proposition 11.6, that algorithm runs in time $O(\deg^2(\varphi_i)\mathbf{M}(p))$ (and we bound $\mathbf{M}(p)$ by $\operatorname{poly}(\log p)$ as discussed in Section 2.1). □

5.3.3. *Finding a B-powersmooth ℓ-suitable translate.* As discussed, we wish to keep the powersmoothness bound B on the degree of an isogeny chain endomorphism low when translating by an integer. Since our goal is to find ℓ-suitable endomorphisms, and translation by ℓ preserves ℓ-suitability, we may search amongst nearby translates for one which is B-powersmooth for our desired bound B.

Algorithm 5.3: Computing a B-powersmooth ℓ-suitable translate in prime-power isogeny-chain form.

Input: A traced endomorphism (E, θ, t, n) in prime-power isogeny chain form , and a powersmoothness bound B (where $B = \infty$ is acceptable).

Output: A traced endomorphism (E, θ', t', n') which satisfies $\mathbb{Z}[\theta'] = \mathbb{Z}[\theta]$ but where θ' is ℓ-suitable, and is given as a separable prime-power isogeny chain, with prime powers $\leq B$.

1 Compute the minimal ℓ-suitable translate T for θ (Lemma 4.5).
2 Try values $n(b) = n + (T + b\ell)t + (T + b\ell)^2$ for small integers b, to find b such that $n(b)$ is B-powersmooth and coprime to p.
3 $\theta' \leftarrow$ a refactored prime-power isogeny chain for $\theta + T + b\ell$, using Algorithm 5.1.
4 $t' \leftarrow t + 2T + 2b\ell$
5 $n' \leftarrow n + (T + b\ell)t + (T + b\ell)^2$.
6 **Return** (E, θ', t', n')

Proposition 5.9. *Algorithm 5.3 is correct, and the runtime is that of Algorithm 5.1 plus the time taken for Step 2.*

Proof. The ℓ-suitability of the output is guaranteed by Lemma 4.5. □

5.3.4. *Choosing a powersmoothness bound B.* In practice, we need to balance the runtimes of the various functionalities of an isogeny chain endomorphism by choosing an appropriate powersmoothness bound B.

The number of B-smooth and B-powersmooth numbers below a bound X is asymptotically the same, provided that $B/\log^2 X \to \infty$ [48] (another reference shows they are asymptotically proportional, provided $\log B/(\log\log X) \to \infty$ [14, Section 3.1]). In our situation, we expect to handle endomorphisms which may have degree as much as exponential in $\log p$. Fortunately, we can, at least heuristically, find subexponentially smooth translates in subexponential time [14, Section 3.1].

Heuristic 5.10. *Given integers n, t, and T, values of the function $f(b) = n + (T + b\ell)t + (T + b\ell)^2$, as $b \to \infty$, are powersmooth with the same probability as randomly chosen integers of the same size.*

This is the powersmooth analogue of the heuristic assumption underlying the quadratic sieve; see [17].

Proposition 5.11. *Assume Heuristic 5.10. Let $\theta \in \operatorname{End}(E)$ have degree d such that $L_d(1/2) > \operatorname{poly}(\log p)$. Then Algorithm 5.3 produces a $L_d(1/2)$-powersmooth prime power isogeny chain of total degree $O(d)$. Furthermore, on $L_d(1/2)$-powersmooth prime power isogeny chains of total degree $O(d)$, the maximum runtime of Algorithm 5.1, Algorithm 5.2 and Algorithm 5.3 is $L_d(1/2)$, and the output of these algorithms is again an $L_d(1/2)$-powersmooth prime power isogeny chain of total degree $O(d)$.*

Proof. We have seen that all the runtimes in Algorithms 5.1 through 5.3 are polynomial in B, $\log\deg\theta$ ($= \operatorname{poly}(\log p)$ by assumption), and $\log p$, with the exception of Step 2 in Algorithm 5.3. Hence, taking $B = L_d(1/2)$, the runtime (except for this step) will be $L_d(1/2)$.

As far as Step 2, under Heuristic 5.10, we can call on [14, Section 3.1] (note that the L-notation in the reference differs from ours here). According to [14, Section 3.1], the probability that a random integer between 1 and d is B-powersmooth is $1/L_d(1/2)$. Testing values of b between 1 and $L_d(1/2)$, we do indeed have $n(b) < d$. Thus, we expect to find a B-powersmooth integer, by Heuristic 5.10. For each b-value, to see whether $n(b)$ is B-powersmooth, we use naïve division in time $O(B\log^2 B)$. Therefore, in total, one will find $L_d(1/2)$-powersmooth integers in time $L_d(1/2)$. □

A few important notes for the remainder of the paper: **we will assume $B = L_{\deg\theta}(1/2)$, where θ is the initial input endomorphism, when dealing with isogeny chains, and that whenever we perform an ℓ-suitable translation on an isogeny chain, we choose a B-powersmooth prime power ℓ-suitable translate.**

Example 5.12 (**Computing an ℓ-suitable translation** via Algorithm 5.3). We continue with our running example, computing an ℓ-suitable translation of a degree-47 endomorphism θ on the curve $E_{1728} : y^2 = x^3 - x$ for $\ell = 2$. Here θ is given as a rational map:

$$\theta(x,y) = \left(\frac{99x^{47} + 22x^{46} + \cdots + 77}{x^{46} + 40x^{45} + \cdots + 77}, \frac{113ix^{69} + 157ix^{68} + \cdots + 63i}{x^{69} + 60x^{68} \cdots + 158} y\right).$$

The traced endomorphism is $(E_{1728}, \theta, 0, 47)$. In Step 1, we compute the minimal 2-suitable translate T using Lemma 4.5. From the traced endomorphism, we compute $\Delta_\theta = t^2 - 4n = 0^2 - 4 \cdot 47 = -188$. This implies that the fundamental discriminant is -47 and the conductor is 2. Therefore the 2-suitable translates are of the form $\theta + T$ for T in $1 + 2\mathbb{Z}$, and the minimal 2-suitable translate is obtained for $T = 1$. In Step 2, we find $b = 0$ produces $n(b) = 2^4 \cdot 3$, which is B-powersmooth for $B = 50$. In Step 3, we factor $\theta + 1$ into an isogeny chain $\theta' = \varphi_{171} \circ \varphi_{1728}$ where $\deg(\varphi_{1728}) = 16$ and $\deg(\varphi_{171}) = 3$, which requires a call to Algorithm 5.1. Here,

$$\varphi_{1728}(x,y) = \left(\frac{x^{16} + (156i + 63)x^{15} + \cdots + 56i + 36}{x^{15} + (156i + 63)x^{14} + \cdots + 10i + 71}, \frac{x^{23} + (55i + 95)x^{22} + \cdots + 105i + 82}{x^{23} + (55i + 95)x^{22} + \cdots + 26i + 87} y\right)$$

and

$$\varphi_{171}(x,y) = \left(\frac{x^3 + (102i + 30)x^2 + (31i + 74)x + 10i + 158}{x^2 + (102i + 30)x + 98i + 130}, \frac{x^3 + (153i + 45)x^2 + (3i + 88)x + 102i + 108}{x^3 + (153i + 45)x^2 + (115i + 32)x + 45i + 174} y\right).$$

The quantities in Steps 4 and 5 can be computed immediately from the values of t, n, T, b, and ℓ, yielding $t' = 2$ and $n' = 48$. The algorithm returns $(E_{1728}, \theta', t', n')$.

5.4. **Poly-rep runtime.** In the last two sections, we computed the runtimes of the basic operations for rationally represented and isogeny chain endomorphisms. We summarize here.

Proposition 5.13. *Suppose θ is an isogeny whose trace t, norm n and discriminant Δ are all at most polynomial in p. If θ is rationally represented, then:*

1. *Evaluating at ℓ-torsion takes time $O(n \operatorname{poly}(\log p))$ (Lemma 2.4).*
2. *Waterhouse twisting by an ℓ-isogeny takes time $\widetilde{O}(n \operatorname{poly}(\log p))$ (Lemma 2.6).*
3. *Dividing by ℓ takes time $O(n^2 \operatorname{poly}(\log p))$ (Proposition 11.6).*
4. *Computing an ℓ-suitable translate takes time $\widetilde{O}(\max\{n, t^2\} \operatorname{poly}(\log p))$ (Lemma 2.7).*

If θ of degree $O(d)$ is represented as a B-powersmooth prime power isogeny chain with $B = L_d(1/2)$ as described in Section 5.3.4, then, assuming Heuristic 5.10 (see Proposition 5.11):

1. *Evaluating at ℓ-torsion takes time $L_d(1/2)$ (Lemma 2.4).*
2. *Waterhouse twisting takes time $L_d(1/2)$ (Proposition 5.6).*
3. *Dividing by ℓ takes time $L_d(1/2)$ (Proposition 5.8).*
4. *Computing a B-powersmooth ℓ-suitable translate takes time $L_d(1/2)$ (Proposition 5.9).*

Of course, in individual situations, these runtimes may be much lower (for example, dividing an isogeny chain by $[\ell]$ may depend only on the power of ℓ if no refactoring is necessary).

In the following algorithms, we will need to call all of these operations many times. It will be convenient to set the following definition.

Definition 5.14. We define the *representation runtime* of a given representation (rationally represented or isogeny chain) to be the maximum runtime of implementing the following operations: evaluating at ℓ-torsion, ℓ-suitable translation, division-by-ℓ, and Waterhouse twisting by an ℓ-isogeny. We say that an algorithm has *poly-rep runtime* if its runtime is bounded above by a constant power of $\log p$ times the relevant representation runtime.

Note that our definition above means that, **throughout the paper** $\operatorname{poly}(\log p) \leq$ **poly-rep**.

6. Orientation-finding for $j = 1728$

For many cryptographic applications, a curve with known endomorphism ring is assumed. Most commonly used is the curve with $j = 1728$, which is supersingular when $p \equiv 3 \pmod 4$. For simplicity, this is the curve we will consider here, but our algorithm can be modified to suit other situations (see below). We will use the model given by $E_{\text{init}} : y^2 = x^3 - x$, which has endomorphism ring

$$\left\langle 1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{k}}{2}, \frac{1+\mathbf{j}}{2} \right\rangle, \quad \mathbf{i}^2 = -1, \mathbf{j}^2 = -p, \mathbf{k} = \mathbf{ij}.$$

In particular, $\mathbf{i}$ is given by $(x, y) \mapsto (-x, \sqrt{-1}y)$ and $\mathbf{j}$ is the Frobenius endomorphism[3] $(x, y) \mapsto (x^p, y^p)$.

Let $\mathcal{O}$ be an imaginary quadratic order of conductor coprime to ℓ such that $\mathcal{O}$ embeds in $B_{p,\infty}$. In this section we give an algorithm for finding an endomorphism $\varphi \in \operatorname{End}(E_{\text{init}})$, generating a suborder $\mathcal{O}' \subseteq \mathcal{O}$ of discriminant $\ell^{2r}\Delta_{\mathcal{O}}$ for the minimal possible r. In other words, we wish to find an ℓ-primitive orientation by a suborder $\mathcal{O}'$ of $\mathcal{O}$. Or, rephrased again, we want to find an orientation for E_{init} placing it at its highest level (nearest to the rims) in the oriented supersingular isogeny graph cordillera with rims at $\mathcal{O}$. Alternatively, the algorithm can be run continuously, to return all ℓ-primitive orientations by suborders of $\mathcal{O}$ in order of increasing r.

The algorithm we provide (Algorithm 6.1) has similarities to [31, Integer Representation, Section 3.2], where the difference arises because we seek a given discriminant instead of a given norm. In fact, this algorithm applies more generally to curves over $\mathbb{F}_p$ satisfying the hypotheses of [31, Section 3.2]; in Section 6.3 we make some comments on adapting this algorithm for other initial curves of known endomorphism ring.

An algorithm for a similar problem appears in [54, Section 4.3]. However, that algorithm finds the 'smallest' quadratic order only: it requires the discriminant be bounded above by $2\sqrt{p} - 1$. We wish to find orientations by more general orders.

---

[3] Note that some papers use the model $y^2 = x^3 + x$, such as [22, Section 5.1]; this model is a quartic twist of ours and under the induced isomorphism of the endomorphism rings, the element which is realized as Frobenius is not preserved. The model we choose for this paper has 2-torsion conveniently defined over $\mathbb{F}_p$. See [47].

6.1. **In terms of 1, i, j, k.** The goal of Algorithm 6.1 is to find such an endomorphism as a linear combination of 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$.

The idea is to solve a norm equation for E_{init} under extra conditions that guarantee that the result is an element of the desired quadratic order. The algorithm depends on Cornacchia's algorithm, which is discussed in [12, Section 1.5.2] and [24, Section 3.1]. It solves the equation $x^2 + y^2 = n$ when a square root of -1 modulo n is known (e.g., such a square root can be found if n is factored).

Algorithm 6.1: Computing an orientation for the initial curve.

Input: A discriminant $\Delta_{\mathcal{O}}$ coprime to p, which is the discriminant of an ℓ-fundamental quadratic order $\mathcal{O}$ that embeds into $B_{p,\infty}$.

Output: (θ, r) where $\theta \in \text{End}(E_{\text{init}})$ is represented as a linear combination of 1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, with $\mathbb{Z}[\theta] = \mathcal{O}' \subseteq \mathcal{O}$ where $[\mathcal{O} : \mathcal{O}'] = \ell^r$. Furthermore, θ is ℓ-primitive. (Here E_{init} and $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are as in the introduction to this section, namely the specified model of $j = 1728$.)

```
r ← −1.
repeat
    r ← r + 1.
    Find the smallest positive x such that x^2 ≡ −Δ_O ℓ^{2r} (mod p).
    While x < sqrt(−Δ_O ℓ^{2r}) do
        D ← (−Δ_O ℓ^{2r} − x^2)/p.
        If D ≡ 1 (mod 4) then
            If D is prime then
                Find a square root of −1 modulo D.
                Use Cornacchia's algorithm to find y and z such that y^2 + z^2 = D.
                If y is odd then
                    Swap y and z.
                If x is even then
                    θ ← 1/2 + (x/2)i + (z/2)j + (y/2)k.
                else
                    θ ← (x/2)i + (y/2)j + (z/2)k.
                break the While loop
        x ← x + p
until θ is defined
c ← 0
While c < r do
    Translate θ to be minimally ℓ-suitable (Lemma 4.5).
    If θ/ℓ ∈ End(E_init) then
        θ ← θ/ℓ.
        c ← c + 1
    else
        break the While loop
Return θ as a linear combination, r − c
```

Remark 6.1. Algorithm 6.1 can be adapted to run continuously, finding many K-orientations of 1728. Simply continue the loops instead of breaking them, returning a solution θ every time one is found.

Remark 6.2. If one wishes to find *all* possible solutions, remove the requirements that D be a prime congruent to $1 \pmod 4$, although this will adversely affect runtime (Cornacchia's algorithm will require factoring D). Furthermore, we must make sure Cornacchia's algorithm returns *all* solutions, and we must include solutions obtained by changing the sign of x on each solution already obtained. We must also be aware that later solutions may fail to be ℓ-primitive; these can be discarded. With these adjustments, every orientation of the form specified will eventually be found by the algorithm (not every θ, but every embedding of $\mathcal{O}'$ into $\mathrm{End}(E_{\mathrm{init}})$ for all $\mathcal{O}'$) – see the proof of Proposition 6.3 for relevant details.

Because of the primality testing step, the algorithm terminates only heuristically. We separately prove its correctness (if it returns) and then give a heuristic runtime.

In what follows, write $\Delta := \Delta_{\mathcal{O}}$ for convenience.

Proposition 6.3. *Any solution returned by Algorithm 6.1 is correct.*

Proof. We attempt to solve the problem for each fixed r increasing from $r = 0$.

If the order $\mathcal{O}'$ of index ℓ^r in $\mathcal{O}$ has even discriminant (namely $\Delta\ell^{2r}$), then we seek an element of reduced trace zero and reduced norm $-\Delta\ell^{2r}/4$. Such an element must generate $\mathcal{O}'$, and $\mathcal{O}'$ must contain a generator of this form. Write the element as $\theta = \frac{x}{2}\mathbf{i} + \frac{y}{2}\mathbf{j} + \frac{z}{2}\mathbf{k}$. Then, simplifying the equation, the norm condition is

$$x^2 + py^2 + pz^2 = -\Delta\ell^{2r}.$$

Any solutions must have $x^2 < \sqrt{-\Delta\ell^{2r}}$, and for a valid x, solutions y and z are found by Cornacchia's algorithm applied to

$$y^2 + z^2 = (-\Delta\ell^{2r} - x^2)/p.$$

In order to be contained in $\mathrm{End}(E_{\mathrm{init}})$, we require $x \equiv z \pmod 2$ and y is even. The variable r is incremented if no solution exists, or if Cornacchia's algorithm is not applied because D is not a prime congruent to 1 (mod 4) (in which case we may miss solutions).

If $\Delta\ell^{2r}$ is odd, we instead seek an element of reduced trace 1 and reduced norm $(-\Delta\ell^{2r} + 1)/4$. Such an element will again necessarily generate $\mathcal{O}'$, and $\mathcal{O}'$ must contain a generator of this form. Writing the element as $\theta = \frac{1}{2} + \frac{x}{2}\mathbf{i} + \frac{y}{2}\mathbf{j} + \frac{z}{2}\mathbf{k}$, after slightly simplifying the norm equation, we must solve the same equation as before:

$$x^2 + py^2 + pz^2 = -\Delta\ell^{2r}.$$

However, in order to lie in $\mathrm{End}(E_{\mathrm{init}})$, such an element must satisfy the conditions that $x \equiv z \pmod 2$ and y is *odd* (note the parity difference). The rest of this case is as above.

If θ is not ℓ-primitive, the algorithm will translate and divide by ℓ until it is. □

For the runtime analysis, and the assertion that the algorithm returns a solution at all, we need a heuristic similar to that used for torsion-point attacks [21, Heuristic 1] and the KLPT algorithm [31, Section 3.2].

Heuristic 6.4. *Fix integers $D > 0$ and $b > 0$, and a prime p coprime to Db that splits in the real quadratic field $\mathbb{Q}(\sqrt{D})$. Ranging through pairs*

$$\left\{(r, x) : 0 < x, x^2 < Db^{2r}, 0 \le r, Db^{2r} - x^2 \equiv 0 \pmod p\right\},$$

consider the value

$$N(r, x) = \frac{Db^{2r} - x^2}{p}.$$

The probability that $N(r, x)$ is a prime congruent to 1 modulo 4 is at least $O(1/(\log D \log N(r, x)))$, where the implied constant is independent of p, D, and b.

We now give a brief justification for this heuristic by passing to the real quadratic field $\mathbb{Q}(\sqrt{D})$. Write $D = f^2 d$ where $d > 0$ is squarefree. We have $N(r, x) = q$ if and only if $\pm pq = N(x + fb^r\sqrt{d})$. Hence we need to estimate the probability, given that $N(x + fb^r\sqrt{d})$ is divisible by p, that it is of the form $\pm pq$ for some other prime q. We analyse instead the probability, for $\alpha \in \mathcal{O}_{\mathbb{Q}(\sqrt{d})}$ (having no assumptions on the form of α), given that $N(\alpha)$ is divisible by p, that it is of the form $\pm pq$ for some prime q. Heuristically, we assume that this will be the same probability.

Given that p splits, we have a prime $\mathfrak{p}$ above p in the maximal order of $\mathbb{Q}(\sqrt{d})$. Hence $N(\alpha)$ has the form $-pq$ if and only if there is a prime ideal $\mathfrak{q}$ of norm q satisfying $\mathfrak{p}\mathfrak{q} = (\alpha)$ (or $\overline{\mathfrak{p}}\mathfrak{q} = (\alpha)$). If $p \mid N(\alpha)$, then replacing $\mathfrak{p}$ with $\overline{\mathfrak{p}}$ if necessary, this occurs if and only if the integral ideal $(\alpha)\mathfrak{p}^{-1} \in [\mathfrak{p}]^{-1}$ has norm q.

Therefore, we estimate the probability that integral elements in $[\mathfrak{p}]^{-1}$ of size X have prime norm. This is bounded below by the probability that integers of size X have a norm which is a prime represented by the class $[\mathfrak{p}]^{-1}$. This in turn is bounded below by $\frac{1}{h \log X}$ where h is the class number of $\mathbb{Q}(\sqrt{d})$. We apply this estimate with $X = N(r, x)$.

Finally, following the Cohen-Lenstra heuristics for real quadratic fields, it may be reasonable to expect the class number $h_{\mathbb{Q}(\sqrt{d})}$ to have an expected value bounded by $O(\log d)$, since the number of prime factors of d is around $\log\log d$ (see [56] for a result for prime discriminants and recall that the 2-part of the class group is controlled by the number of prime factors of d).

Heuristic 6.4 has been confirmed numerically in some small cases; we will consider this heuristic in more detail in [2]. The corresponding heuristic, in the case of the KLPT norm equation, has been verified by Wesolowski [55]; it would be nice to know if similar methods apply here.

Proposition 6.5. *Suppose Heuristic 6.4 holds and Δ is coprime to p. If $|\Delta| < p^2$, then Algorithm 6.1 returns a solution of norm at most $p^2 \log^{2+\epsilon}(p)$ with $r = O(\log p)$ in time $O(\log^{6+\epsilon}(p))$. If instead $|\Delta| > p^2$, then the algorithm will return a solution with $r = O(1)$ and norm $O(|\Delta|)$ in time $O(\sqrt{|\Delta|}\log^{4+\epsilon}(\Delta)(\log p)p^{-1})$.*

Running the algorithm continuously, subsequent solutions should be found in the same runtime, with r expected to increase by 1, and their norms expected to increase by a constant factor of ℓ^2 at each subsequent solution.

Proof. Suppose r is of size at most $u \log_\ell p$, where u is positive (otherwise r is not positive). Then $\sqrt{-\Delta\ell^{2r}} \leq |\Delta|^{1/2}p^u$. Thus, we expect to iterate the **While** loop at Step 5 at most $X(\Delta, u) := \lceil |\Delta|^{1/2}p^{u-1}\rceil + 1$ times. Each time we enter the loop, we obtain a value $D = (-\Delta\ell^{2r} - x^2)/p$ of size $\leq pX(\Delta, u)^2$. The probability that D is prime and $1 \pmod 4$ is heuristically $1/(4\log(p^{1/2}X(\Delta, u)))$ (Heuristic 6.4). Hence we expect to reach Cornacchia's algorithm once u is large enough such that

$$X(\Delta, u) \geq 4\log(p^{1/2}X(\Delta, u)) > 1.$$

Reaching it will terminate the algorithm. This is a mild condition, satisfied asymptotically when $X(\Delta, u) \geq (\log p)^{1+\epsilon}$. In fact, it suffices to take $\sqrt{|\Delta|}p^u \geq p\log^{1+\epsilon}(p)$, or equivalently,

$$u\log p \geq \log p - \frac{1}{2}\log|\Delta| + (1+\epsilon)\log\log p. \tag{3}$$

In particular, $u > 1$ is always enough, and if $|\Delta| > p^{2+\epsilon}$, then any positive value for u will suffice. (An informal explanation of this behaviour: even for a volcano with a trivial rim, distance $(1+\epsilon)\log p$ down its sides is enough to capture all j-invariants. At the same time, if Δ is large enough that the rim likely captures all j-invariants, then we needn't descend the volcano at all.) This shows that the algorithm needs to increase r at most $O(\log p)$ times before it reaches Cornacchia's algorithm.

For $|\Delta| \leq p^{2+\epsilon}$, the optimal value of u is given by (3). However, since u cannot be negative, when $|\Delta| > p^{2+\epsilon}$, the optimal value of u is 0. (Again, informally: the class group will be of size $\approx \sqrt{|\Delta|} > p$, and we will find all $\approx \frac{p}{12}$ supersingular j-invariants already on the rim of an isogeny volcano.)

We first determine the overall runtime in terms of $X(\Delta, u)$ and p. The primality test can be run in time $O(\log^{4+\epsilon} D)$ for example, using the Miller-Rabin algorithm [44, Section 2]. This algorithm is probabilistic, so there is a negligible possibility that Cornacchia's algorithm may fail on false positives.

Once D is a prime congruent to $1 \pmod 4$, we must find a square root of -1 with which to run Cornacchia's algorithm. There is a nice analysis of this exact situation in [24, Section 3.1], which concludes that it takes probabilistic time $\widetilde{O}(\log^2 D)$, which is negligible compared to the primality testing.

Thus, for the final runtime, we increment r at most $O(\log p)$ times, running a primality test of cost $O(\log^{4+\epsilon} D)$ at most $O(X(\Delta, u))$ times for each r, before reaching a point where Cornacchia's algorithm is invoked. Using $D \leq pX(\Delta, u)^2$, this gives runtime $O(X(\Delta, u)(\log p)(\log p + 2\log X(\Delta, u))^{4+\epsilon})$.

In the case of large $|\Delta| > p^{2+\epsilon}$, we put $u = 0$ and obtain $X(\Delta, u) = O(\sqrt{|\Delta|}/p)$ and asymptotically $X(\Delta, u) > p^\epsilon$. This yields a runtime of $O(\sqrt{|\Delta|}\log^{4+\epsilon}(\Delta)(\log p)p^{-1})$. In this case $r = O(1)$ and the norm of the solution is bounded by $O(|\Delta|)$.

In the case of small $|\Delta| \leq p^2$, we optimize u according to (3) and obtain $X(\Delta, u) = O(\log^{1+\epsilon}(p))$ and asymptotically $X(\Delta, u) < p$. This gives $O(\log^{6+\epsilon}(p))$. At the same time, the norm of the solution found is bounded by $|\Delta|\ell^{2r} \leq p^2X(\Delta, u)^2 \leq p^2\log^{2+2\epsilon}(p)$.

Once r has reached $O(\log p)$, we expect solutions for each r with high probability. Therefore, running the algorithm continuously, subsequent solutions should be found in the same runtime as the first, and their sizes should be increasing by an expected constant factor of ℓ^2 at each subsequent solution. □

Example 6.6 (**Computing an orientation for the initial curve** via Algorithm 6.1). We return to our working example $p = 179$, $\Delta = -47$, $\ell = 2$, and $E_{1728} : y^2 = x^3 - x$. Note that $\log_\ell(p) \sim 7.48$, so that we expect the algorithm to succeed reliably once $r = 7$ or 8, if not earlier. Beginning with $r = 0$, in Step 4 we compute the smallest positive x such that $x^2 = 47 \pmod{179}$, namely $x = 88$. As $x = 88$ exceeds $\sqrt{47} \approx 6.9$, we return to Step 3 and increment r to $r = 1$. This reflects the fact that the curve E_{1728} does not admit a $\mathbb{Q}(\sqrt{-47})$-orientation on the rim. Continuing, we find the smallest positive integer x such that $x^2 \equiv 188 \pmod{179}$, namely $x = 3$. As $x = 3 < \sqrt{47 \cdot 4} \approx 13.7$, we define $D = (47 \cdot 4 - 3^2)/179 = 1$ in Step 6. Cornacchia's algorithm returns $1^2 + 0^2 = 1$. We obtain the element $\frac{3\mathbf{i}+\mathbf{k}}{2} \in \mathrm{End}(E_{1728})$. This indicates (correctly) that E_{1728} admits an orientation on level $r = 1$ of the $\mathbb{Q}(\sqrt{-47})$-oriented 2-isogeny volcano, see the node with j-invariant 1728 in Figure 1. If we continue to run the algorithm, looking for pairs (r, θ) for r up to 8, we return three more pairs:

$$\left(r = 7, \theta = \frac{371}{2}\mathbf{i} + 29\mathbf{j} + \frac{13}{2}\mathbf{k}\right), \left(r = 8, \theta = \frac{153}{2}\mathbf{i} + 27\mathbf{j} + \frac{119}{2}\mathbf{k}\right), \left(r = 8, \theta = \frac{511}{2}\mathbf{i} + 41\mathbf{j} + \frac{95}{2}\mathbf{k}\right).$$

6.2. **As an isogeny chain endomorphism.** Since $\mathbf{i}$ and $\mathbf{j}$ are known endomorphisms which can be evaluated at points, any combination of these can also be evaluated at points. Therefore the output of Algorithm 6.1 can be fed into Algorithm 5.3, and an ℓ-suitable isogeny chain endomorphism will result. Thus, in poly-rep time (that is, depending on B, the powersmoothness bound), we can obtain the output of Algorithm 6.1 as an isogeny-chain endomorphism.

6.3. **Curves other than $j = 1728$.** Algorithm 6.1 can be adapted to work for certain curves E_{init} other than the curve with $j = 1728$. In particular, if the endomorphism ring $\mathrm{End}(E)$ of a curve E defined over $\mathbb{F}_p$ is of the form $\mathcal{O} + \mathbf{j}\mathcal{O}$, where $\mathbf{j}$ is the Frobenius endomorphism and $\mathcal{O}$ is a quadratic order, then the adaptation of Algorithm 6.1 is clear, where we use the principal norm form of $\mathcal{O}$ in place of $x^2 + y^2$. As before, this will reduce to Cornacchia's algorithm. Instead of primes that are $1 \pmod 4$, we seek primes that split in the field and are coprime to the conductor of $\mathcal{O}$; this requires a Legendre symbol computation. The runtime is essentially unchanged. This adaptation follows the discussion in [31, Section 3.2].

6.4. **Heuristics.** We now formalize a heuristic about the behaviour of Algorithm 6.1 needed for what follows. This is a version of Heuristic 3.8 specific to the algorithm we use.

Heuristic 6.7. *Let $\mathcal{O}$ be a quadratic order. Let $\mathcal{S}$ be the finite union of $\mathcal{O}'$-cordilleras where $\mathcal{O}' \supseteq \mathcal{O}$. Then Algorithm 6.1 running continuously will (i) eventually produce solutions on every volcano of $\mathcal{S}$, and (ii) produce solutions which are approaching the distribution described in Heuristic 3.8 (i.e. with probabilities proportional to the number of descending edges from the rim).*

If $\mathcal{S}$ has only one volcano, this heuristic is immediate as long as the algorithm produces infinitely many solutions (which happens by Proposition 6.5, under heuristic assumptions from Section 3.6). If Algorithm 6.1 returned *all* orientations of 1728, then this heuristic would follow directly from Heuristic 3.8. The difficulty is that it finds only those solutions where the primality testing step succeeds. In other words, we cannot rule out the unlikely possibility that the primality condition causes all the orientations of 1728 to be missed on some individual volcano. Thus, we seem to require a version of Heuristic 6.4 which asserts that the primality is independent of whether the eventual solution is on any fixed volcano of the cordillera. We consider Heuristic 6.7 more closely in the companion paper [2].

7. Supporting algorithms for walking on oriented curves

7.1. **Computing an ℓ-primitive endomorphism.** Recall from Definition 4.3 that an endomorphism θ is ℓ-primitive if the associated orientation is ℓ-primitive. If θ is chosen to be ℓ-suitable, then equivalently, θ is ℓ-primitive if it cannot be divided by $[\ell]$ in $\mathrm{End}(E)$ (Lemma 4.4). Therefore, given θ, we can translate it to become ℓ-suitable and then divide by $[\ell]$ as often as possible to obtain an ℓ-primitive endomorphism.

Proposition 7.1. *Algorithm 7.1 is correct, and runs in poly-rep time (see Definition 5.14).*

Proof. If $t^2 - 4n$ is ℓ-fundamental, then the conductor of the quadratic order generated by θ is not divisible by ℓ; in this case θ is already ℓ-primitive. In order to check if any order of superindex ℓ contains $\mathbb{Z}[\theta]$ within $\mathrm{End}(E)$, we first translate θ to be ℓ-suitable, and then check whether it is divisible by $[\ell]$ within $\mathrm{End}(E)$. If it is, we divide by ℓ and repeat.

For runtime, the algorithm translates to an ℓ-suitable translate, tests for divisibility by ℓ, and divides by ℓ, at most a polynomial number of times (since we assume that the discriminant of $\mathbb{Z}[\theta]$ is bounded by a power of p; see Section 5.1). ☐

Algorithm 7.1: Computing an ℓ-primitive endomorphism given an endomorphism.

Input: A traced endomorphism (E, θ, t, n) providing the functionality of Section 5.1.
Output: A traced endomorphism (E, θ', t', n') which is ℓ-primitive, and the ℓ-valuation of the index $[\mathbb{Z}[\theta'] : \mathbb{Z}[\theta]]$.

```
If t^2 − 4n is ℓ-fundamental then
    Return (E, θ, t, n) and 0.
(E, θ, t, n) ← an ℓ-suitable translate of (E, θ, t, n)
c ← 0
While [ℓ] | θ do
    (E, θ, t, n) ← (E, θ/[ℓ], t/ℓ, n/ℓ^2)
    c ← c + 1
    If t^2 − 4n is ℓ-fundamental then
        Return (E, θ, t, n) and c.
    (E, θ, t, n) ← an ℓ-suitable translate of (E, θ, t, n)
Return (E, θ, t, n) and c.
```

Example 7.2 (**Computing an ℓ-primitive endomorphism** via Algorithm 7.1). We apply Algorithm 7.1 to the output of Example 5.12, namely $(E_{1728}, \theta', t', n')$ where $\theta' = \varphi_{171} \circ \varphi_{1728}, t' = 2, n' = 48$. This is not at the rim, but is already ℓ-suitable. We find $[2] \nmid \theta'$ by evaluating on $E_{1728}[2]$; hence we return the input unchanged.

7.2. **Rim walking via the class group action.** In the case that an orientation is available, one can walk the rim of the oriented ℓ-isogeny volcano using the class group action. Walking a cycle generated by the class group action was first described in Bröker-Charles-Lauter [6] in the case of ordinary curves, which carry an orientation by Frobenius. This was later used in CSIDH [7], and it was remarked that it extends to orientations by $\mathbb{Q}(\sqrt{-np})$ in Chenu-Smith [9]. In this section we provide a generalization of the same algorithm to arbitrary orientations. The algorithm walks the rim from a specified start curve in an arbitrary direction until it encounters a specified end curve. This path is computed using the action of the class group on the *oriented* curves in the rim of the *oriented* volcano. As such, it requires knowledge of the orientation, so the steps of the algorithm must pull the orientation (i.e. the endomorphism) along with them.

More precisely, the ideal we wish to apply to (E, θ) is given in terms of θ, so that one can use the methods of Bröker-Charles-Lauter [6, Section 3] with θ in place of Frobenius. One can apply the Waterhouse twist of θ, and divide by ℓ to carry along θ in the computation.

The algorithm works by applying the action of $\mathrm{Cl}(\mathcal{O})$ to a rim of elements primitively oriented by a quadratic order $\mathcal{O}$. In fact, using $\mathrm{Cl}(\mathcal{O})$ works just as well if the rim is primitively oriented by $\mathcal{O}' \supseteq \mathcal{O}$, where $\ell \nmid [\mathcal{O}' : \mathcal{O}]$. This allows us to walk on any rim associated to an ℓ-fundamental discriminant Δ, without knowing for sure that the orientation is primitive with respect to Δ. See Proposition 3.4.

Calling Algorithm 7.2 on identical input curves (i.e. $(E_{\mathrm{init}}, \iota_{\mathrm{init}}) = (E_{\mathrm{target}}, \iota_{\mathrm{target}})$ yields the entire rim of the ℓ-oriented isogeny graph.

Proposition 7.3. *Algorithm 7.2 is correct. Each step of the rim walk has poly-rep runtime. The number of steps is bounded $O(h_{\mathcal{O}})$. Furthermore, if θ is in prime-power isogeny chain form with any powersmoothness bound B, then each step of the rim-walk has runtime polynomial in B.*

Proof. If $\ell \mid t^2 - 4n$, then either we are not at the rim, or the field discriminant is not coprime to ℓ. If $j(E_1) = j(E_2)$, we have already completed our task. Assuming neither of those cases, we compute the abstract quadratic order $\mathcal{O}$ generated by θ using its minimal polynomial, and associate an abstract element α_θ to θ. The volcano rim in question is contained in $\mathrm{SS}_{\mathcal{O}'}$ for some $\mathcal{O}' \supseteq \mathcal{O}$, where the index of containment $f = [\mathcal{O}' : \mathcal{O}]$ is coprime to ℓ (by ℓ-primitivity). If ℓ is inert in $\mathcal{O}$, then it is also inert in $\mathcal{O}'$. Hence the rim of the associated volcano is trivial; since $j(E_1) \neq j(E_2)$, this indicates there is no valid path to be found. Otherwise, ℓ is split or ramified in $\mathcal{O}$, so we factor it and compute a and b and τ as in the algorithm. Namely, we have the factorzation $\ell\mathcal{O} = (\ell, \tau)_{\mathcal{O}}(\ell, \overline{\tau})_{\mathcal{O}}$ in $\mathcal{O}$. Then $\ell\mathcal{O}' = (\ell, \tau)_{\mathcal{O}'}(\ell, \overline{\tau})_{\mathcal{O}'}$ in $\mathcal{O}'$. Therefore,

Algorithm 7.2: Walking along the rim of the oriented supersingular ℓ-isogeny graph

Input: An ℓ-primitive traced endomorphism $(E_1, \theta_1, t_1, n_1)$ providing the functionality of Section 5.1, and a target curve E_2.

Output: If E_1 and E_2 are on the same volcano rim in the oriented isogeny graph for the field $\mathbb{Q}(\theta)$, with discriminant coprime to ℓ, the algorithm returns a path of oriented horizontal ℓ-isogenies from $(E_1, \theta_1, t_1, n_1)$ to a vertex with curve E_2. Otherwise returns FAILURE.

1 **If** $\ell \mid t^2 - 4n$ **then**
2   **Return** FAILURE.
3 $H \leftarrow []$.
4 **If** $j(E_1) = j(E_2)$ **then**
5   **Return** H.
6 Compute $\mathcal{O} \cong \mathbb{Z}[\theta]$, the quadratic order generated by θ (using trace and norm), together with an explicit isomorphism given in the form of $\alpha_\theta \in \mathcal{O}$ corresponding to θ.
7 **If** *ℓ is inert in $\mathcal{O}$* **then**
8   **Return** FAILURE.
9 Compute $\tau \in \mathcal{O}$ such that $\mathfrak{l} = (\ell, \tau)_{\mathcal{O}}$ is a prime ideal of $\mathcal{O}$ above ℓ.
10 Compute $a, b \in \mathbb{Z}$ so that $\tau = a + b\alpha_\theta$.
11 $(E, \theta, t, n) \leftarrow (E_1, \theta_1, t_1, n_1)$.
12 **repeat**
13   Compute $E[\ell]$.
14   Compute $E[\mathfrak{l}] \leftarrow E[\ell] \cap \ker(a + b\theta)$ by evaluating $a + b\theta$ on $E[\ell]$.
15   Use Vélu's algorithm to compute the ℓ-isogeny $\nu : E \to E'$ with kernel $E[\mathfrak{l}]$.
16     $(E, \theta, t, n) \leftarrow (E', \nu \circ \theta \circ \widehat{\nu}, t\ell, n\ell^2)$.
17     $(E, \theta, t, n) \leftarrow (E, \theta/[\ell], t/\ell, n/\ell^2)$.
18   Append $(\nu, (E, \theta, t, n))$ to H.
19 **until** $(j(E), \theta, t, n) = (j(E_1), \theta_1, t_1, n_1)$ *or* $j(E) = j(E_2)$
20 **If** $j(E) = j(E_2)$ **then**
21   **Return** H
22 **else**
23   **Return** FAILURE

the isogeny computed is the action of the ideal $\mathfrak{l} \in \mathrm{Cl}(\mathcal{O}')$ lying above ℓ in $\mathcal{O}'$ on $\mathrm{SS}_{\mathcal{O}'}$ as desired, which is thus a horizontal isogeny. The **repeat** clause walks the rim step by step.

We stop if we meet E_2 or return to our (oriented) starting point. The latter occurs only if we have walked the entire rim, which means E_2 was not on that rim.

For runtime, all individual steps are polynomial, except for calls to evaluate at ℓ-torsion points, Waterhouse twist and divide by ℓ. The number of repeats is equal to the path length from E_1 to E_2 along the rim. The size of the rim is $O(h_{\mathcal{O}})$ (Section 3.4).

For the final statement, note that no ℓ-suitable translation is needed in the algorithm. In fact, the norm of the endomorphism remains constant as one walks the rim. □

Example 7.4 (**Walking along the rim of the oriented supersingular ℓ-isogeny graph** via Algorithm 7.2). As before, we have $K = \mathbb{Q}(\sqrt{-47})$. We use Algorithm 7.2 on input $\ell = 2$, $(E_{22}, \theta_{22}, t_{22}, n_{22})$ and target curve E_{22} to compute the entire rim of the oriented 2-isogeny volcano for purposes of demonstration. The endomorphism θ_{22} is a primitive $\mathcal{O}_K$-orientation, so the curve E_{22} lies on the rim of a $\mathcal{O}_K$-oriented isogeny volcano. Step 9 computes the prime ideal $\ell = (2, \omega)_{\mathcal{O}_K}$. In Step 13, we compute $E_{22}[2] = \{\mathcal{O}_{E_{22}}, (2, 0), (156i + 178, 0), (23i + 178, 0)\}$. We obtain $E_{22}[\mathfrak{l}] = \langle (156i + 178, 0) \rangle$ in Step 14. Velu's formulas in Step 15 compute the isogeny $\varphi_{22} : E_{22} \to E_{99i+107}$. The codomain of φ_{22} is $E_{99i+107} : y^2 = x^3 + (26i + 88)x + (141i + 104)$. In Step 16, we compute the traced endomorphism $(E_{99i+107}, \theta_{99i+107}, t_{99i+107}, n_{99i+107})$ with $\theta_{99i+107} := \frac{1}{2}\, \varphi_{22} \circ$

$\theta_{22} \circ \hat{\varphi}_{22}$, an endomorphism of degree 12. Step 18 appends the isogeny φ_{22} and the traced endomorphism $(E_{99i+107}, \theta_{99i+107}, t_{99i+107}, n_{99i+107})$ to H.

In the next rim step, starting with $(E_{99i+107}, \theta_{99i+107}, t_{99i+107}, n_{99i+107})$, we compute the isogeny $\varphi_{99i+107} : E_{99i+107} \to E_{5i+109}$. The isogeny $\varphi_{99i+107}$ and traced endomorphism $(E_{5i+109}, \theta_{5i+109}, t_{5i+109}, n_{5i+109})$ are appended to H in Step 18.

In the next rim step, we find the isogeny $\varphi_{5i+109} : E_{5i+109} \to E_{174i+109}$ and corresponding traced endomorphism $(E_{174i+109}, \theta_{174i+109}, t_{174i+109}, n_{174i+109})$ with $\theta_{174i+109} = \frac{1}{2}(\varphi_{5i+109}) \circ \theta_{5i+109} \circ \hat{\varphi}_{5i+109}$.

A fourth step along the rim produces the isogeny $\varphi_{174i+109} : E_{174i+109} \to E_{80i+107}$ and traced endomorphism $(E_{80i+107}, \theta_{80i+107}, t_{80i+107}, n_{80i+107})$.

The final step along the rim produces the isogeny $\varphi_{80i+107} \to E'_{22}$ with codomain $E'_{22} : y^2 = (125i + 98)x + (84i + 152)$ and induced traced endomorphism $(E'_{22}, \theta'_{22}, t'_{22}, n'_{22})$. The codomain E'_{22} is isomorphic to E_{22} via an isomorphism ρ, and we use the same isomorphism ρ to confirm that E'_{22} and E_{22} are in fact isomorphic as oriented curves by computing $\theta'_{22} = \rho \circ \theta_{22} \circ \rho^{-1}$.

Algorithm 7.2 terminates and returns the rim cycle

$$E_{22} \xrightarrow{\varphi_{22}} E_{99i+107} \xrightarrow{\varphi_{99i+107}} E_{5i+109} \xrightarrow{\varphi_{5i+109}} E_{174i+109} \xrightarrow{\varphi_{174i+109}} E'_{22} \cong E_{22}$$

of length 5 (see the green rim cycle in Figure 1). Indeed, K has class number 5, and the ideal class of $\mathfrak{l}$ generates the class group of K.

7.3. **Ascending to the rim using an orientation.** The other major component of navigating the supersingular ℓ-isogeny graph using an orientation is to walk to the rim. We can use Proposition 4.8 to determine the ascending direction and walk up. This is described in Algorithm 7.3. The number of steps to the rim is expected to be $\log(p)$ in general; see Section 3.6.

Algorithm 7.3: Walking to the rim of the oriented ℓ-isogeny graph.

Input: An ℓ-primitive traced endomorphism (E, θ, t, n) providing the functionality of Section 5.1.
Output: The shortest path from (E, θ, t, n) to the rim of the oriented ℓ-isogeny volcano upon which (E, θ, t, n) lies.

```
1  H ← [].
2  k ← ⌊ ν_ℓ(t² − 4n) / 2 ⌋.
3  If ℓ = 2 and (t² − 4n)/2^{2k} ≢ 1 (mod 4) then
4      k ← k − 1
5  For j = 1, . . . , k do
6      Compute E[ℓ].
7      (E, θ, t, n) ← an ℓ-suitable translate of (E, θ, t, n).
8      Compute a generator P for E[ℓ] ∩ ker(θ).
9      Use Vélu's algorithm to compute the ℓ-isogeny ν : E → E' with kernel ⟨P⟩.
10     (E, θ, t, n) ← (E', ν ∘ θ ∘ ν̂, tℓ, nℓ²)
11     (E, θ, t, n) ← (E, θ/[ℓ²], t/ℓ², n/ℓ⁴)
12     Append (ν, (E, θ, t, n)) to H.
13 Return H
```

Proposition 7.5. *Algorithm 7.3 is correct and has poly-rep runtime times the distance to the rim.*

Proof. The number of steps to the rim is given by the number of times ℓ^2 divides the discriminant of θ (we assume θ is ℓ-primitive); this is k in Step 2. We translate θ to be ℓ-suitable, which implies that $\nu \circ \theta \circ \widehat{\nu}$ can be divided by $[\ell]$ twice when ν is ascending. Since there is no horizontal direction (by the choice of k in Step 2), there exists a non-trivial $P \in E[\ell] \cap \ker(\theta)$. This gives the ascending isogeny by Proposition 4.8. Once we have found the ascending isogeny, we divide the Waterhouse twist of θ by $[\ell]^2$ (Step 11), and the result is ℓ-primitive, in preparation for the next loop iteration. For each iteration of the **For** loop, the work is clearly poly-rep. ∎

Example 7.6 (**Walking to the rim of the oriented ℓ-isogeny graph for rationally represented endomorphisms** via Algorithm 7.3). We apply Algorithm 7.3 to the output of Step 4 of Example 8.3, namely E_{120} and θ_{120} having $t_{120} = 0$, $n_{120} = 188$. We find that we expect to take two steps to the rim. Since θ_{120} is already 2-suitable, we evaluate it on $E_{120}[2]$ and obtain the kernel $\langle(121i + 4, 0)\rangle$ for the ascending isogeny. The codomain is E_{171}. Waterhouse twisting and dividing by [2] twice, we obtain an endomorphism θ' which is not 2-suitable, but Lemma 4.5 shows that $\theta_{171} := \theta' + [1]$ is 2-suitable. The second ascending step is similar; this has kernel $\langle(121i + 131, 0)\rangle$ and codomain E_{5i+109}. The two ascending steps are in blue in Figure 1.

Example 7.7 (**Walking to the rim of the oriented ℓ-isogeny graph for isogeny chain endomorphisms** via Algorithm 7.3). We begin with input $(E_{1728}, \varphi_{171} \circ \varphi_{1728}, 2, 48)$, from Step 8 of Example 8.3. This will require one step to the rim and is already [2]-suitable. Evaluating on $E_{1728}[2]$, we obtain a kernel of $\langle(178, 0)\rangle$ for the ascending isogeny; the codomain is E_{22}. Waterhouse twisting yields an isogeny-chain which is not prime-power refactored, namely $\varphi'_{1728} \circ \varphi_{171} \circ \varphi_{1728} \circ \widehat{\varphi'}_{1728}$ having component degrees 2, 3, 16, 2, respectively. We could apply Algorithm 5.1, but we proceed in a slightly more expedient manner. We rewrite $\varphi'_{1728} \circ \varphi_{171}$, having degrees 2 and 3, respectively, in a form having degrees 3 and 2, respectively. Thus, we evaluate $\varphi'_{1728} \circ \varphi_{171}$ on the 2-torsion to obtain the kernel $\langle(29i+50, 0)\rangle$ determining $\varphi'_{171} : E_{171} \to E_{174i+109}$. Then we apply φ'_{171} to the generator of $\ker(\varphi'_{1728} \circ \varphi_{171}) \cap E_{171}[3] = \langle(128i + 164, 28i + 90)\rangle$ to obtain a kernel for which Vélu gives $\varphi_{174i+109} : E_{174i+109} \to E_{22}$. We obtain the refactored isogeny chain $\varphi_{174i+109} \circ \varphi'_{171} \circ \varphi_{1728} \circ \widehat{\varphi'}_{1728}$. We can then divide the 2-power degree component $\varphi'_{171} \circ \varphi_{1728} \circ \widehat{\varphi'}_{1728}$ by [2] twice and let $\varphi'_{22} := \varphi'_{171} \circ \varphi_{1728} \circ \widehat{\varphi'}_{1728}/[4]$. Replacing this in our isogeny chain above, we now have an isogeny that gives the one step up to the rim (see the red step in Figure 1):

$$(E_{1728}, \varphi_{171} \circ \varphi_{1728}, 2, 48) \xrightarrow{\varphi'_{1728}} (E_{22}, \varphi_{174i+109} \circ \varphi'_{22}, 1, 12).$$

7.4. **Ascending and walking the rim using the endomorphism ring.** When we find an orientation of $j = 1728$, we have more information than just the specified orientation: we also know the endomorphism ring. This extra information allows us to navigate the oriented graph in polynomial time using known algorithms.

Specifically, with Algorithm 7.4 given here, we can walk up the volcano and traverse the rim (being careful not to back-track by comparing to our previous steps), where each step is polynomial in $\log p$ and the length of the representation of θ. To get started, we use E_{init} as the curve defining $B_{p,\infty}$ as in [55], and take the path P to be the trivial path.

Proposition 7.8. *Under GRH, Algorithm 7.4 is correct and runs in expected polynomial time in the following quantities:* $\log p$*, the size of the representation of* θ*, and the length of the path* P*.*

Proof. Each of the cited algorithms runs in the time specified under GRH. We determine which steps are ascending or horizontal by testing whether $\beta/\ell^{s+1}, \beta/\ell^{s+2} \in \mathfrak{O}$, by Proposition 4.7. Since β is represented as a linear combination of a basis of $\operatorname{End}(E')$, this involves dividing the coefficients, which is polynomial time. □

8. Classical path-finding to $j = 1728$

We now present an algorithm which, given a suitable endomorphism on a curve in the supersingular graph, will find a path to the initial curve, under heuristic assumptions. An illustration of the method is given in Figure 1: we walk from the initial endomorphism to its rim; find an orientation of E_{init} and walk from that orientation of E_{init} to its rim; and hope to collide on the same rim.

Proposition 8.1. *Assume GRH, Heuristic 6.4, and the assumptions of Section 5.1. Consider an endomorphism* $\theta \in \operatorname{End}(E)$ *in rationally-represented or prime-power isogeny-chain form as described in Section 5.4, whose discriminant is coprime to* p *and has* ℓ*-fundamental part* Δ *satisfying* $|\Delta| < p^2$*. Write* $\mathcal{O}_\Delta$ *for the order of discriminant* Δ*. Algorithm 8.1 produces a path of length* $O(\log p + h_{\mathcal{O}_\Delta})$ *to* E_{init} *in the supersingular* ℓ*-isogeny graph, under Heuristic 6.7 part (i). The runtime is expected poly-rep times* $O(h_{\mathcal{O}_\Delta})$*, under Heuristic 6.7 part (ii). Furthermore, the following hold:*

1. *If* ℓ *is inert in* K*, then the runtime improves to* $h_{\mathcal{O}_\Delta} \operatorname{poly}(\log p)$*+poly-rep, and the path length improves to* $O(\log p)$*.*
2. *If* ℓ *is inert in* K *and the discriminant of* θ *is already* ℓ*-fundamental, then the runtime improves to* $h_{\mathcal{O}_\Delta} \operatorname{poly}(\log p)$ *and the path length improves to* $O(\log p)$*.*

Algorithm 7.4: Extending a path from E_{init} by an ascending or horizontal step.

Input: A fixed endomorphism $\theta \in \text{End}(E_{\text{init}})$. An elliptic curve E and path P from E_{init} to E, with no descending steps, and s equal to the number of ascending steps in the path P.

Output: For each of the available horizontal or ascending steps $E \to E'$ (with regards to the orientation induced by θ), returns the data (E', P', s'), where P' is the path obtained from P by extending it by the extra step, and s' is the number of ascending steps in the path P'.

```
H ← []
For each ℓ-isogeny ν : E → E' departing E do
    P' ← the path formed by appending ν to P.
    (φ : E_init → E') ← the isogeny associated to the path P'.
    Compute a ℤ-basis of the maximal quaternion order 𝔒 of E' and connecting ideal I between E_init
      and E' using [55, Algorithm 3] from the path P'.
    Compute End(E') together with an isomorphism Ψ : End(E') → 𝔒, using [55, Algorithm 6].
    β ← Ψ(φ ∘ θ ∘ φ̂) (The ability to evaluate Ψ(φ ∘ θ ∘ φ̂) for θ ∈ End(E_init) is also obtained
      when [55, Algorithm 6] is performed in the last step.)
    β ← β + T where T ∈ ℤ is chosen so that β + T is the minimal ℓ^s-suitable translate of φ ∘ θ ∘ φ̂
      using Lemma 4.5.
    If β/ℓ^{s+1} ∈ 𝔒 then
        s' ← s
        If β/ℓ^{s+2} ∈ 𝔒 then
            s' ← s' + 1
        Append (E', P', s') to H.
Return H.
```

Algorithm 8.1: Finding a path to E_{init}.

Input: A traced endomorphism (E, θ, t, n) providing the functionality of Section 5.1, where the discriminant of θ is coprime to p.

Output: A path in the ℓ-isogeny graph between E and E_{init}.

```
(E, θ, t, n) ← (E, θ/[ℓ^k], t/ℓ^k, n/ℓ^{2k}) which is ℓ-primitive, using Algorithm 7.1.
Δ_θ ← t^2 − 4n.
Δ ← the ℓ-fundamental part of Δ_θ.
Call Algorithm 7.3 on input (E, θ, t, n) to produce an ascending path H_2 from (E, θ, t, n) to
  (E_1, θ_1, t_1, n_1) on the rim, i.e. where ℤ[θ_1] ⊆ End(E_1) is ℓ-fundamental.
Call Algorithm 7.2 on input (E_1, θ_1, t_1, n_1) to walk the rim until we encounter E_1 again, storing the
  j-invariants encountered as a list L.
repeat
    Call Algorithm 6.1 on input Δ, to obtain a new solution θ_init = a + bi + cj + dk. (Algorithm 6.1
      can be suspended and then resumed to find subsequent solutions; see Remark 6.1)
    Using the methods of Section 7.4, produce an ascending path H_1 from E_init with endomorphism
      θ_init up to the rim, i.e. to a traced endomorphism (E_0, θ_0, t_0, n_0) having ℓ-fundamental
      ℤ[θ_0] ⊆ End(E_0).
until E_0 ∈ L or E_0^{(p)} ∈ L
Compute H_rim, the path from E_1 to E_0 or E_0^{(p)}, using L.
If H_rim joins E_1 to E_0 then
    H ← H_2 H_rim^{-1} H_1^{-1}, a path from E_init to E.
else
    From H_1, compute the conjugate path H_1^{(p)} from E_init to E_0^{(p)}.
    H ← H_2 H_rim^{-1} (H_1^{(p)})^{-1}, a path from E_init to E.
```

(3) If Δ is a fundamental discriminant, ℓ is split in K and a prime above ℓ generates the class group $\mathrm{Cl}(\mathcal{O}_\Delta)$*, then the dependence on Heuristic 6.7 is removed.*

Proof. Let θ be the input to the algorithm. The pair (E, ι_θ), where $\iota_\theta : K \to \mathrm{End}(E)$ is the orientation given by θ, lies somewhere on the oriented ℓ-isogeny graph associated to K. More specifically, it lies on a volcano of the $\mathcal{O}$-cordillera for some order $\mathcal{O}$ whose discriminant divides the ℓ-fundamental discriminant Δ computed in Step 3. In other words, if we write $\mathcal{O}_\Delta$ for the order of discriminant Δ, then $\mathcal{O} \supseteq \mathcal{O}_\Delta$. Since all endomorphisms throughout the paper are taken to have norm and discriminant at worst polynomial in p, the distance of (E, ι_θ) to the rim is at worst polynomial in $\log p$, and so walking to the rim (Step 4) is poly-rep by Proposition 7.5. Next, we walk around the rim; the runtime depends on the size of the rim and we defer that question to later in the proof.

When Δ is passed on to Algorithm 6.1 in Step 7, the result (which is returned in polynomial time by Proposition 6.5 under Heuristic 6.4) is an endomorphism of $\mathrm{End}(E_{\mathrm{init}})$ which gives an oriented elliptic curve lying somewhere on a volcano in an $\mathcal{O}'$-cordillera, where again $\mathcal{O}' \supseteq \mathcal{O}_\Delta$. (We do not necessarily have $\mathcal{O} = \mathcal{O}'$.) This has norm polynomial in p by Proposition 6.5. By Proposition 6.5 again, the distance to the rim is $O(\log p)$, so walking to the rim is expected polynomial time by Proposition 7.8. Hence each **repeat** iteration has expected polynomial time.

Walking to the rim in Step 8, E_0 lies on the rim of a volcano. This volcano is somewhere in the set of volcanoes $\mathcal{S}$ defined as the finite union of the $\mathcal{O}$-cordilleras for all $\mathcal{O} \supseteq \mathcal{O}_\Delta$. Note that its conjugate $E_0^{(p)}$ also lies on a rim in $\mathcal{S}$. Now E_1 also lies on a rim of $\mathcal{S}$. If E_0 (or $E_0^{(p)}$) and E_1 lie on the same rim, the algorithm will discover this. If not, then one continues the calls to Algorithm 6.1, and another endomorphism will be found. Under Heuristic 6.7 part (i), eventually one of these will produce E_0 or $E_0^{(p)}$ on the same rim as E_1. The algorithm will then succeed.

Let R denote the number of descending edges from the rim containing E_0, referred to in this paragraph as the *adjusted rim size* (which is bounded above and below by a constant multiple of the rim size). The sum of the adjusted rim sizes of all rims of $\mathrm{SS}_\mathcal{O}$ for all $\mathcal{O} \supseteq \mathcal{O}_\Delta$ is $O(H_{\mathcal{O}_\Delta})$ (Propositions 3.3 and 3.6). By Lemma 3.9, this is $O(h_{\mathcal{O}_\Delta})$. By Heuristic 6.7 part (ii), the number of times we must **repeat** is therefore $O(h_{\mathcal{O}_\Delta}/R)$. Each iteration performs Steps 7 and 8 and then checks membership in L. By Proposition 6.5, under GRH, Step 7 runs in polynomial time in $\log p$ and provides a solution θ_{init} of norm at most $p^2 \log^{2+\epsilon} p$. Then θ_{init} can be written as a linear combination of the $\mathbb{Z}$-basis of $\mathrm{End}(E_{\mathrm{init}})$ with integer coefficients of size $O(\log p)$. Hence Step 8 requires a runtime polynomial in $\log p$ by Proposition 7.8; we store the j-invariant of the output for comparison to L. Thus, each iteration is expected polynomial time times $O(R)$ (to check membership in L). The walk to produce L in Step 5 takes at most $O(R)$ steps, each of which is poly-rep. Hence the runtime is poly-rep (for Step 4) plus $O(h_{\mathcal{O}_\Delta}) \cdot \mathrm{poly}(\log p) + O(R) \cdot (\text{poly-rep})$.

This runtime is overall bounded by $O(h_{\mathcal{O}_\Delta})$ times poly-rep. But if ℓ is inert, then E_0 lies on a rim of size 1, so we don't need Step 5, and we have poly-rep plus $h_{\mathcal{O}_\Delta} \mathrm{poly}(\log p)$. If θ is already at the rim, then we don't need Step 4. Combined with inertness, this gives runtime $h_{\mathcal{O}_\Delta} \mathrm{poly}(\log p)$.

Finally, if Δ is a fundamental discriminant, ℓ is split and a prime above ℓ generates $\mathrm{Cl}(\mathcal{O}_\Delta)$, then there is only one volcano, obviating the need for Heuristic 6.7. □

The restriction that $|\Delta| < p^2$ is required to ensure that Algorithm 6.1 is heuristically polynomial time. If $|\Delta|$ is larger, and ℓ is inert, this failure of polynomial time could become the bottleneck. On the other hand, suppose ℓ is split in K. Under the Cohen-Lenstra heuristics, class groups are usually cyclic, and most elements of a cyclic group are generators, so with high probability, Heuristic 6.7 will not be necessary.

It is also possible to use Algorithm 7.3 at Step 4, instead of the methods of Section 7.4. This results in a worse runtime, but removes the dependence on GRH.

Remark 8.2. One might hope to modify Algorithm 8.1 to produce a shorter path along with a square-root runtime improvement, by removing Step 5, and in each **repeat**, attempting to solve a vectorization problem (see Section 9.1) between E_0 and E_{init}. Unfortunately, we cannot: the problem is that we do not know the correct quadratic order $\mathcal{O}$ with respect to which these oriented curves are primitively oriented. To overcome this, one might try to factor Δ and ascend with respect to any square factors, to guarantee that Δ is fundamental. Ascending would be polynomial in the largest squared prime factor of Δ, which could be very costly. An alternative that would usually work may be to try guessing Δ, working backward from the largest (and hence most likely) divisors. Just assuming Δ is fundamental would work much of the time.

Example 8.3 (**Finding a path to** E_{init} via Algorithm 8.1). We again let $p = 179$, $\Delta = -47$, $\ell = 2$, and $E_{\text{init}} = E_{1728} : y^2 = x^3 - x$. As input, we consider the curve $E_{120} : y^2 = x^3 + (7i + 86)x + (45i + 174)$ with $j(E_{120}) = 120$, and a trace endomorphism given as $(E_{120}, \theta_{120}, t_{120}, n_{120})$ with $t_{120} = 20, n_{120} = 2^5 \cdot 3^2$ and

$$\theta_{120}(x, y) = \left(\frac{(122i + 167)x^{288} + (17i + 68)x^{287} + \cdots + 174i + 157}{x^{287} + (78i + 156)x^{286} + \cdots + 16i + 54}, \frac{(69i + 109)x^{431} + (60i + 178)x^{430} + \cdots + 98i + 124}{x^{431} + (146i + 53)x^{430} + \cdots + 44i + 89} \right).$$

We apply Algorithm 8.1 to find a path from E_{120} to E_{1728} (see Figure 1). Step 1 on input $(E_{120}, \theta_{120}, t_{120}, n_{120})$ produces the ℓ-suitable and ℓ-primitive traced endomorphism $\theta_{120} \leftarrow \theta_{120} + [-10]$ with $t_{120} \leftarrow 0$ and $n_{120} \leftarrow 188$. Here $\Delta' = t_{120}^2 - 4n_{120} = -752$ and its ℓ-fundamental part is $\Delta = -47$. Step 4 calls Algorithm 7.3 on input $(E_{120}, \theta_{120}, t_{120}, n_{120})$ to produce the following ascending path H_2 to the rim, see Example 7.6:

$$H_2 : (E_{120}, \theta_{120}, 0, 188) \xrightarrow{\varphi_{120}} (E_{171}, \theta_{171}, 0, 47) \xrightarrow{\varphi_{171}} (E_{5i+109}, \theta_{5i+109}, 1, 12).$$

Now we apply Algorithm 7.2 on input $(E_{5i+109}, \theta_{5i+109}, t_{5i+109}, n_{5i+109})$ to walk the rim in Step 5 as in Example 7.4. The list of all the j-invariants is $L = \{5i + 109, 174i + 109, 80i + 107, 22, 99i + 107\}$. In Step 7, calling Algorithm 6.1 on input Δ, we obtain $\theta_{1728} = (3i + k)/2$ as in Example 6.6. For simplicity in this example, we use Algorithm 7.3 in Step 8, instead of the methods of Section 7.4. We apply Algorithms 5.3 and 7.1 (see Section 6.2) to $(E_{1728}, \theta_{1728}, 0, 47)$ to obtain an ℓ-primitive isogeny-chain endomorphism $\theta'_{1728} = \varphi_{171} \circ \varphi_{1728}$ where $\deg(\varphi_{1728}) = 16$, $\deg(\varphi_{171}) = 3$ and with $t_{1728} = 2$, $n_{1728} = 48$ as in Example 5.12. We call Algorithm 7.3 on input $(E_{1728}, \varphi_{171} \circ \varphi_{1728}, 2, 48)$ to produce the following ascending path (see Example 7.7):

$$H_1 : (E_{1728}, \varphi_{171} \circ \varphi_{1728}, 2, 48) \xrightarrow{\varphi'_{1728}} (E_{22}, \varphi_{174i+109} \circ \varphi'_{22}, 1, 12).$$

Finally, since $j(E_{22}) = 22 \in L$, joining the previous paths, we obtain a path from E_{1728} to E_{120} (see the whole path in Figure 1) as

$$H : E_{1728} \xrightarrow{\varphi'_{1728}} E_{22} \xrightarrow{\varphi_{22}} E_{99i+107} \xrightarrow{\varphi_{99i+107}} E_{5i+109} \xrightarrow{\hat{\varphi}_{171}} E_{171} \xrightarrow{\hat{\varphi}_{120}} E_{120}.$$

9. Quantum algorithms

9.1. **Vectorization.** Since the class group acts on the rim, a problem closely related to walking along the rim is the following, where we use the terminology *vectorization* in analogy with [15] and [9, Section 6.1]. This problem was also recently introduced in [54, Section 3.1].

Problem 9.1 (OrientedVectorization(Δ)). Let $\mathcal{O}$ be the quadratic order of discriminant Δ. Suppose $(E_1, \iota_1), (E_2, \iota_2) \in \mathrm{SS}_{\mathcal{O}}$. Find an ideal class $[\mathfrak{b}] \in \mathrm{Cl}(\mathcal{O})$ such that $[\mathfrak{b}] \cdot (E_1, \iota_1) = (E_2, \iota_2)$.

The following result was implied without details in a more restricted case in [9, Section 6.1]. A variation also appears in [54, Proposition 4].

Heuristic 9.2. *The values of a definite binary quadratic form $f(x, y)$, as $x, y \to \infty$, are powersmooth and coprime to the first N primes with the same probability as randomly chosen integers of the same size.*

Proposition 9.3. *Assume Heuristic 9.2. Suppose (E_1, ι_1) and (E_2, ι_2) are given by $\iota_i := \iota_{\theta_i}$ for some endomorphisms $\theta_i \in \mathrm{End}(E_i)$ which can be evaluated on $E_i(\mathbb{F}_{p^k})$ in time $T_{\theta_i}(k, p) \geq \mathrm{poly}(k \log p)$. Define $T_{\theta_1,\theta_2}(k, p) := max\{T_{\theta_1}, T_{\theta_2}\}$. Let d be the maximal degree of the θ_i. Then* OrientedVectorization($|\Delta|$) *can be reduced to a hidden shift problem and solved in quantum time $T_{\theta_1,\theta_2}(\log^2 d, p)L_{|\Delta|}(1/2)$ under GRH, where, furthermore, the ideal class is $L_{|\Delta|}(1/2)$-smooth and of size $O(\sqrt{|\Delta|})$.*

Proof. The approach is essentially the same as that in Childs-Jao-Soukharev [11], who developed a subexponential means of evaluating the action of the class group (by finding a smooth representative of the needed ideal class), and then applying Kuperberg's algorithm, which requires subexponentially many evaluations. The difference is that we need to apply the class group action, in the form of isogenies, to *oriented* curves, i.e. carry along the orientation.

The reduction to the hidden shift problem is formalized in [33, Theorem 3.3]; the *malleability oracle* in the sense of [34, Definition 3.2], with respect to their notation, is given in terms of $I = G = \mathrm{Cl}(\mathcal{O})$, $O = \mathrm{SS}_{\mathcal{O}}$, and $f : I \to O$ defined by $f([\mathfrak{a}]) = [\mathfrak{a}] \cdot (E_1, \iota_1)$. Then to find $[\mathfrak{b}] \in \mathrm{Cl}(\mathcal{O})$ such that $[\mathfrak{b}] \cdot (E_1, \iota_1) = (E_2, \iota_2)$, we observe that f is malleable, because we can compute $[\mathfrak{a}] \mapsto f([\mathfrak{a}\mathfrak{b}]) = [\mathfrak{a}\mathfrak{b}] \cdot (E_1, \iota_1) = [\mathfrak{a}] \cdot (E_2, \iota_2)$ (this is the malleability oracle at (E_2, ι_2)). See [33].

To evaluate the action of $[\mathfrak{a}]$ on E takes time $\mathrm{poly}(\log p)L_{|\Delta|}(1/2)$ using the methods of [11] or [5] and involves finding an $L_{|\Delta|}(1/2)$-smooth integral representative $\mathfrak{a}$ which can be evaluated as a composition chain of isogenies. Unfortunately, to evaluate the action of $[\mathfrak{a}]$ on θ, we require a powersmooth representative instead. Calling on Heuristic 9.2 and [14, Section 3.1] (similarly to the proof of Proposition 5.11), we can find a representative with norm $L_{|\Delta|}(1/2)$-powersmooth and coprime to the first $\log \deg \theta$ primes, by random search. The time taken is $L_{|\Delta|}(1/2)$, because by Mertens' Theorem, the probability of satisfying the coprimality hypothesis is $\prod_{\substack{p<O(\log \deg \theta) \\ p \text{ prime}}}(1-1/p) \sim O(1/\log\log\deg\theta)$. Having done this, write the result as $\mathfrak{a} := \prod \mathfrak{a}_k$, where the $N(\mathfrak{a}_k)$ are coprime prime powers.

We also need to evaluate the action of $\mathfrak{a}$ on θ in some way that is distinguishable (since isogeny chains are not unique for a given endomorphism). For each j-invariant we choose a fixed model. We replace the data of θ with the data of its linear action on the $O(\log \deg \theta)$ smallest prime-torsion subgroups $E[q]$, as well as all the prime-power $N(\mathfrak{a}_i)$-torsion subgroups. By Chinese Remainder Theorem, this is enough to distinguish different results, since if $\theta - \theta'$ vanishes on all of the prime-power subgroups, then it vanishes on a subgroup (generated by all of the subgroups together), whose size exceeds a fixed multiple of d, which implies that $\theta = \theta'$ (this method is inspired by the Schoof algorithm, as adapted for example in [32, Theorem 81], [22, Lemma 4]).

To compute the action on θ_i, we first need to compute $\varphi_{\mathfrak{a}_k}$. This is done as in Algorithm 7.2, where we consider the linear action of $a + b\theta_i$ on the $N(\mathfrak{a}_k)$-torsion to find the kernel of $\varphi_{\mathfrak{a}_k}$. In order to compute the linear action of $\varphi_{\mathfrak{a}_k} \circ \theta \circ \widehat{\varphi_{\mathfrak{a}_k}}/[N(\mathfrak{a}_k)]$ on the prime or prime-power torsion subgroups $E[q]$ described in the last paragraph, we proceed as follows. If q is coprime to $N(\mathfrak{a}_k)$, then to find this action, we evaluate $\varphi_{\mathfrak{a}_k} \circ \theta \circ \widehat{\varphi_{\mathfrak{a}_k}}$ on $E[q]$ and then apply the action of $[n']$ where $n' \equiv N(\mathfrak{a}_k)^{-1} \pmod q$. Otherwise we store **null** for that value of q (by assumption, this occurs only for q larger than $\log \deg \theta$).

This gives a way to evaluate the function f suitable for quantum computation. Taken together, the time taken for evaluating $[\mathfrak{a}_k]$ is $\mathrm{poly}(\log \deg \theta)$ times the time taken to evaluate θ and $\varphi_{\mathfrak{a}_k}$, namely $T_{\theta_1,\theta_2}(\log^2 d, p) + \mathrm{poly}(\log p)L_{|\Delta|}(1/2)$.

There is a small caveat that the action of Frobenius may take us out of the orbit of $\mathrm{Cl}(\mathcal{O})$, so this will only work when the oriented curves E_1 and E_2 are in the same $\mathrm{Cl}(\mathcal{O})$-orbit. Of course, there are at most two orbits, so in the case of failure, we can apply Frobenius to one of the curves and try again. See Proposition 3.3.

The evaluation algorithm of [11] runs in time $L_{|\Delta|}(1/2)$ under GRH and results in an $L_{|\Delta|}(1/2)$-smooth isogeny of size $O(\sqrt{|\Delta|})$ [11, Proposition 3.2]. Our modification above results in the stated runtime. □

Remark 9.4. If we wish to avoid the coprimality aspect of Heuristic 9.2, then we can take subexponentially many prime power torsion subgroups, at an increased cost in runtime and memory (thanks to Benjamin Wesolowski for this and other helpful observations and corrections to this proof).

Remark 9.5. If we wish to avoid Heuristic 9.2 in Proposition 9.3 entirely, we could first transform θ into a powersmooth isogeny chain (at a runtime cost of $T_\theta(L_{\deg\theta}(1/2), p)$) and then use the method for horizontal stepping of Algorithm 7.2 to evaluate $[\mathfrak{a}]$ prime-by-prime. This allows for the representative $\mathfrak{a}$ to be chosen as smooth, not necessarily powersmooth, but incurs an additional runtime cost to the algorithm as a whole.

9.2. **Primitive orientation computation.** Another useful task is to solve the following problem.

Problem 9.6 (PrimitiveOrientation). Given an elliptic curve E, and an endomorphism $\theta \in \mathrm{End}(E)$, determine the quadratic order $\mathcal{O}$ such that ι_θ is $\mathcal{O}$-primitive.

Our method for solving Problem 9.6 has similarities to that of Proposition 9.3, with a hidden subgroup problem in place of the hidden shift problem. The subexponential runtime in Δ still arises from the need to evaluate the action of the class group.

Proposition 9.7. *Assume Heuristic 9.2. Suppose θ can be evaluated on $E(\mathbb{F}_{p^k})$ in time $T_\theta(k,p)$. Then there is an algorithm to solve* PrimitiveOrientation *in quantum time $T_\theta(\log\deg\theta, p) + \mathrm{poly}(\log p)L_{|\Delta|}(1/2)$.*

Proof. Let $\mathcal{O}_\theta := \mathbb{Z}[\theta]$. Compute $\mathrm{Cl}(\mathcal{O}_\theta)$ as a product of cyclic groups with given generators, using the quantum algorithm [10, Algorithm 10], as described in [11, Proof of Theorem 4.5]. It is possible to solve the PrimitiveOrientation problem by computing the kernel of the map $\mathrm{Cl}(\mathcal{O}_\theta) \to \mathrm{Cl}(\mathcal{O})$ (where we do not a priori know $\mathcal{O}$). This can be done using a hidden subgroup problem. Namely, we consider the action

$\mathrm{Cl}(\mathcal{O}_\theta)$ on $\mathrm{SS}_{\mathcal{O}}$, defining $f([\mathfrak{b}]) = [\mathfrak{b}] \cdot (E, \iota_\theta)$. We evaluate the action of $\mathfrak{b}$ on θ as described in the proof of Proposition 9.3.

Once the kernel G has been computed in the form of generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_n$, one writes each of these ideals as principal $\mathfrak{g}_i = (g_i)$ in the maximal order. Then $\mathcal{O}$ is by definition the order generated from $\mathcal{O}_\theta$ by adjoining the g_i's. One computes the conductor of this order by taking the gcd of the conductors of the $\mathbb{Z}[g_i]$ and $\mathbb{Z}[\theta]$, and hence computing the discriminant $\Delta_{\mathcal{O}}$. These last computations are polynomial in $\log |\Delta_\theta|$. □

An improvement is available: to evaluate the action of $[\mathfrak{b}]$ on E takes time $\mathrm{poly}(\log p) \exp(\widetilde{O}(\log^{1/3} |\Delta_\theta|))$ using the methods of Biasse-Iezzi-Jacobson [5]; they also improve on the computation of $\mathrm{Cl}(\mathcal{O})$.

9.3. **Computation of a smooth isogeny to $j = 1728$.** The problems of computing the endomorphism ring of an elliptic curve E, computing an ℓ-power isogeny to an initial curve (such as $j = 1728$), and computing a smooth isogeny to an initial curve, are all equivalent [55]. In this section, we modify Algorithm 8.1 to find a smooth isogeny, using the algorithms of the previous two sections (Propositions 9.3 and 9.7). The resulting algorithm is Algorithm 9.1.

Algorithm 9.1: Finding a smooth isogeny to E_{init} (quantum)

Input: A traced endomorphism (E, θ, t, n) which can be evaluated on arbitrary points, where the discriminant of θ is coprime to p.

Output: A smooth isogeny $E \to E_{\mathrm{init}}$.

1 $\Delta \leftarrow t^2 - 4n$

2 Choose the smallest prime ℓ so that ℓ^2 does not divide Δ or n.

3 $\Delta' \leftarrow$ the discriminant of the solution to PrimitiveOrientation for (E, θ) via Proposition 9.7.

4 **repeat**

5 | Call Algorithm 6.1 on input Δ', to obtain a new traced endomorphism $(E_{\mathrm{init}}, \theta_{\mathrm{init}}, t_{\mathrm{init}}, n_{\mathrm{init}})$. (Algorithm 6.1 can be suspended and then resumed to find subsequent solutions; see Remark 6.1.)

6 | Walk from $(E_{\mathrm{init}}, \theta_{\mathrm{init}}, t_{\mathrm{init}}, n_{\mathrm{init}})$ to produce an ascending path H_1 from $(E_{\mathrm{init}}, \theta_{\mathrm{init}}, t_{\mathrm{init}}, n_{\mathrm{init}})$ to $(E_0, \theta_0, t_0, n_0)$ on the rim, i.e. where $\mathbb{Z}[\theta_0] \subseteq \mathrm{End}(E_0)$ is ℓ-fundamental (methods of Section 7.4).

7 | $\Delta'' \leftarrow$ the discriminant of the solution to PrimitiveOrientation for (E_0, θ_0) via Proposition 9.7.

8 **until** $\Delta' = \Delta''$

9 Use a quantum computer to solve OrientedVectorization(Δ') as described in Proposition 9.3, to find an ideal class $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O}_{\Delta'})$ such that $[\mathfrak{a}](E_1, \iota_{\theta_1})$ is (E_0, ι_{θ_0}) or $(E_0^{(p)}, \iota_{\theta_0}^{(p)})$ (try both).

Proposition 9.8. *Assume GRH, Heuristic 6.4, Heuristic 6.7, Heuristic 9.2, and the assumptions of Section 5.1. Suppose θ can be evaluated on $E(\mathbb{F}_{p^k})$ in time $T_\theta(k, p) \geq \mathrm{poly}(k \log p)$. Let $d = \max\{\deg \theta, |\Delta|\}$. Suppose $|\Delta| < p^2$ and Δ is coprime to p. Algorithm 9.1 is correct and succeeds in heuristic expected time $T_\theta(\log^2 d, p) L_{|\Delta|}(1/2)$. The resulting $L_{|\Delta|}(1/2)$-smooth isogeny has norm $O(\sqrt{|\Delta|})$.*

Proof. The algorithm determines Δ' so that ι_θ is $\mathcal{O}_{\Delta'}$-primitive. In the **repeat** loop, it finds an orientation of $j = 1728$ and a path from that oriented curve to an oriented curve (E_0, θ_0) which is primitive with respect to the same order. Thus vectorization applies, and finds a smooth isogeny between (E, θ) and (E_0, θ_0). Combining the path and isogeny, we find a smooth isogeny between E and the initial curve.

The first two steps take time $O(\log |\Delta|)$. The third step takes time $T_\theta(\log \deg \theta, p) + \mathrm{poly}(\log p) L_{|\Delta|}(1/2)$ by Proposition 9.7. Steps 5 and 6 take polynomial time in $\log p$ and $\log |\Delta|$ by Proposition 6.3 and Proposition 7.8. Step 7 again takes time $T_\theta(\log \deg \theta, p) + \mathrm{poly}(\log p) L_{|\Delta|}(1/2)$. To determine how often we must **repeat**, we compute that the probability that $\Delta' = \Delta$ is equal to $h_{\mathcal{O}}/H_{\mathcal{O}}$ (by consideration of the sizes of $\mathrm{SS}_{\mathcal{O}}$ (Propositions 3.3 and 3.6) and using Heuristic 6.7), which by Lemma 3.9 is $O(1)$.

Note that the endomorphism found by Algorithm 6.1 is of norm $O(|\Delta|)$. Therefore the rim endomorphism θ_0 is also of norm $O(|\Delta|)$. Thus, OrientedVectorization in Step 9 takes time $T_\theta(\log^2 d, p) L_{|\Delta|}(1/2)$ (Proposition 9.3). Note that the evaluation time for θ_0 on small torsion is $O(\log p)$ since we have expressed θ_0 as a linear combination of basis elements, each of which can be evaluated via the chain down to $j = 1728$. □

10. Proofs of Main Theorems and Special Cases

10.1. **Proof of Theorems 1.1 and 1.3.**

Proof of Theorem 1.1. Suppose θ is such an endomorphism. Then set $B = L_d(1/2)$. We can apply Algorithm 5.3 (having Algorithm 5.1 as a subroutine) to θ, whose runtime depends on the evaluation of θ on inputs in a field $\mathbb{F}_{p^{O(B^2)}}$. The runtime for this conversion is therefore $T_\theta(L_d(1/2), p)$. The result is a prime-power isogeny-chain representation of θ. We can then use Algorithm 8.1, with the representation runtime being $L_d(1/2)$, by Proposition 5.13. The classical runtime follows from Proposition 8.1. □

Proof of Theorem 1.3. For the quantum case, we use Algorithm 9.1, with no need to pre-process θ. The quantum result follows from Proposition 9.8. □

10.2. **Special cases.** In this section, we refer to an endomorphism as *insecure* if access to such an endomorphism allows for a polynomial time path-finding algorithm. Endomorphisms of small size are known to be insecure [36]. We obtain a version of this from our methods also.

Theorem 10.1. *Assume the situation of Theorem 1.1. In the following special cases, the runtime and path length of Algorithm 8.1 is polynomial in* $\log p$*:*

(1) The input endomorphism is rationally represented in polynomial space.
(2) $h_{\mathcal{O}_\Delta} = \mathrm{poly}(\log p)$ *and* ℓ *is coprime to* Δ *and inert in* K*. In this case, the endomorphism is not even needed as input; only its existence, trace and norm are needed.*

Proof. The second case is a consequence of Algorithm 8.1 and Proposition 8.1, in which the hypotheses imply Steps 4 and 5 are unnecessary. The first is a consequence of the observation that such endomorphisms have polynomially sized discriminants and class numbers. □

The following result demonstrates the existence of non-small endomorphisms which are insecure.

Theorem 10.2. *Suppose* $\Delta = f^2\Delta'$ *where* Δ' *is a discriminant of* $\mathrm{poly}(\log p)$ *size,* f *is* $\mathrm{poly}(\log p)$*-smooth, and* θ *is* f*-suitable with* $\mathrm{poly}(\log p)$*-powersmooth norm, and represented in some fashion so that it can be evaluated in* $\mathrm{poly}(\log p)$ *time on points of* $\mathrm{poly}(\log p)$ *size. Then there is an algorithm to find an* $O(\log p)$*-powersmooth isogeny to* E_{init} *in time* $\mathrm{poly}(\log p)$*.*

Proof. The dependence on ℓ throughout the paper has been suppressed by assuming $\ell = O(1)$, but it is at worst polynomial throughout. We refactor θ in $\mathrm{poly}(\log p)$ time (this is possible by Proposition 5.6 and the evaluation runtime assumption), to obtain an isogeny chain. Taking each prime ℓ dividing f in turn, we ascend as for as possible on the oriented ℓ-isogeny volcano. By f-suitability, we can ascend without any further translation or refactoring. Having ascended, we obtain an endomorphism of discriminant Δ' of $\mathrm{poly}(\log p)$ size and trace zero, and hence call on Theorem 10.1 with respect to some suitable ℓ. □

The following corollary guarantees that every elliptic curve has an insecure endomorphism. Recall that most curves do not have small endomorphisms. It is known that there are curves having no endomorphisms of norm smaller than $p^{2/3-\epsilon}$ (see [35, Proposition B.5], [23, Section 4], [57, Proposition 1.4]). Therefore the endomorphisms guaranteed by the following corollary are frequently large.

Corollary 10.3. *Let* p *be such that* $p \equiv 3 \pmod 4$*, and let* E *be any supersingular elliptic curve over* $\mathbb{F}_{p^2}$*. The endomorphism ring* $\mathrm{End}(E)$ *contains an endomorphism which can be presented in* $\mathrm{poly}(\log p)$ *space and evaluated in* $\mathrm{poly}(\log p)$ *time, and knowledge of that endomorphism allows for a classical* $\mathrm{poly}(\log p)$*-time algorithm to find a path to* $j = 1728$*.*

Proof. Consider the Gaussian field $\mathbb{Q}(i)$. Let $L = \prod_i \ell_i$ be a product of the first $O(\log p)$ odd primes. We claim that $\mathrm{End}(E)$ contains $\mathbb{Z}[Li]$. To see this, we use [25, Theorem 1], which asserts that E can be reached by a random walk from E_{init} of $j = 1728$ (which exists since $p \equiv 3 \pmod 4$) with degree L. Then $\mathrm{End}(E)$ must contain $\mathbb{Z}[Li]$ (in fact, it may contain a strictly larger order, if the steps are not all descending with respect to the Gaussian field). Taking the element Li, represented as a $\mathrm{poly}(\log p)$-powersmooth isogeny chain, we apply Theorem 10.2. □

This proof is not constructive, and it is indeed not easy to find such an endomorphism. Examples of such endomorphisms exist in any field with $\mathrm{poly}(\log p)$ discriminant; indeed one can take any element of the form $L(\omega + k)$ for $k \in \mathbb{Z}$ and a $\mathrm{poly}(\log p)$-powersmooth L such that $N(\omega + k)$ is $\mathrm{poly}(\log p)$-powersmooth.

Finally, we remark on one more special case. When the norm of θ is well-behaved, and we are already at the rim with respect to ℓ (perhaps by choosing ℓ judiciously), then we have improved dependence on p. Note that in the following theorem, there is no requirement on the factorization of Δ.

Theorem 10.4. *Suppose the norm of θ has powersmoothness bound $B(p)$, and suppose that Δ is coprime to ℓ. Then there is an algorithm to find an ℓ-isogeny path of length $O(\log p + h_{\mathcal{O}})$ to E_{init} in time $h_{\mathcal{O}}\, \mathrm{poly}(B(p) \log p)$.*

Proof. Use Algorithm 8.1. By the assumption on Δ, we need not ascend with θ (that is, we skip Step 4). We only walk horizontally, and those steps are polynomial in $B(p)$ by Proposition 7.3. □

11. Division by $[\ell]$

We conclude with a detailed description and analysis of McMurdy's algorithm (Algorithm 11.2) which can be used to divide any *isogeny* (not just an endomorphism) by $[\ell]$ if it is a multiple of $[\ell]$. Given a rationally represented traced endomorphism, we apply Algorithm 11.2 and then adjust the trace and norm accordingly.

We follow the notation of McMurdy [38]. Let E_1 and E_2 be two supersingular elliptic curves given by respective short Weierstrass equations

$$E_1 : y^2 = W_1(x), \qquad E_2 : y^2 = W_2(x).$$

with $W_1(x), W_2(x) \in \mathbb{F}_{p^2}[x]$. Denote by $\psi_{E_1,\ell}$ the ℓ-division polynomial of E_1, made monic, and let $X_i(x)$ and $Y_i(x)$ be the rational functions representing the multiplication-by-ℓ map on E_i, i.e. $[\ell]_{E_i}(x, y) = (X_i(x), Y_i(x)y)$ for $i = 1, 2$. For a polynomial $P(x) = (x - r_1) \cdots (x - r_n)$ with coefficients in some field $\mathbb{F}$ whose roots r_i lie in some field extension $\mathbb{F}'$ of $\mathbb{F}$, and a rational function $T(x)$ over FF', define

$$P(x)\big|T := (x - T(r_1)) \cdots (x - T(r_n))\,.$$

Given $[\ell]\varphi : E_1 \to E_2$ as a pair of rational maps, where $\varphi : E_1 \to E_2$ is an isogeny, the rational maps of φ are obtained as follows.

Proposition 11.1 ([38, Proposition 2.6])**.** *Suppose that $\varphi : E_1 \to E_2$ is a separable isogeny such that $([\ell]\varphi)(x, y) = (F(x), G(x)y)$ for rational functions $F(x), G(x)$. Write $F(x)$ in lowest terms, i.e. as either $\frac{c_F \cdot P(x)}{W_1(x)Q(x)}$ when $\ell = 2$ or $\frac{c_F \cdot P(x)}{\psi_{E_1,\ell}(x)^2 Q(x)}$ when $\ell \neq 2$, with monic polynomials $P(x), Q(x)$. Set*

$$p(x) = P(x)\big|X_1, \;\; q(x) = Q(x)\big|X_1.$$

Then $p(x) = p_0(x)^{\ell^2}$ and $q(x) = q_0(x)^{\ell^2}$ for monic polynomials $p_0(x), q_0(x)$. Moreover, we have $\varphi(x, y) = (f(x), g(x)y)$, where $f(x) = c_F \ell^2 \cdot \frac{p_0(x)}{q_0(x)}$ and $g(x) = \frac{G(x)}{Y_2(f(x))}$.

Algorithm 11.1 computes the polynomials $p(x)$ and $q(x)$ as given in Proposition 11.1. The main division-by-$[\ell]$ process (Algorithm 11.2) then calls Algorithm 11.1 twice.

Division by $\ell = 2$ has been implemented by McMurdy [38] (code available at [37]). Division by odd primes $\ell > 2$ is complicated by the non-vanishing of the y-coordinates of the ℓ-torsion points. Fix an odd prime $\ell > 2$. In order to compute $p(x) = P(x)\big|X_1$ and $q(x) = Q(x)\big|X_1$ in Steps 3 and 4 of Algorithm 11.2, we compute the rational map $N_P = \prod_i P(\mathbf{x}_i)$ as a function of the variable x only. In contrast to the case of 2-torsion points, the ℓ-torsion points on E_1 have non-zero y-coordinates, so some $\mathbf{x}_i$ depend not only on x (as in the case $\ell = 2$) but also on y and y_i for $i \leq (\ell^2 - 1)/2$. As a consequence, N_P also depends on these variables. To overcome this obstruction, we employ a new technique presented in Steps 5–11 of Algorithm 11.1. In these steps, we compute the products $\mathbf{x}_i \cdot \bar{\mathbf{x}}_i$, and hence the products $P(\mathbf{x}_i) \cdot P(\bar{\mathbf{x}}_i)$. Each product $P(\mathbf{x}_i) \cdot P(\bar{\mathbf{x}}_i)$ is a rational map in x, y^2, and y_i^2 ($i \leq (\ell^2 - 1)/2$) by Lemma 11.4. We replace y^2 (respectively y_i^2) with $W_1(x)$ (respectively $W_1(x_i)$) to obtain rational maps in the variable x only.

Example 11.2 (**Computing the polynomial** $P(x)\big|X_1$ via Algorithm 11.1). Let $\ell = 3$, $p = 179$, and $E_{1728} : y^2 = x^3 - x$ the supersingular elliptic curve over $\overline{\mathbb{F}}_p$ with $j = 1728$. Let $X_1(x)$, $Y_1(x)$ be associated to multiplication-by-3, i.e.

$$[3]_{E_{1728}}(x, y) = (X_1(x), Y_1(x)y) \quad \text{where} \quad X_1(x) = \frac{20x^9 + 61x^7 + 63x^5 + 175x^3 + x}{x^8 + 175x^6 + 63x^4 + 61x^2 + 20}\,.$$

Algorithm 11.1: Computing the polynomial $P(x)\big|X_1$

Input: An elliptic curve E_1, a monic polynomial $P(x)$ defined over $\mathbb{F}_{p^m}$, and the rational map $X_1(x)$ associated to E_1.
Output: $P(x)\big|X_1$.

1 Compute a root ζ of X_1.
2 Compute the x-coordinates x_i of the points $S_i = (x_i, y_i) \in E_1[\ell]$, indexed by $i = 1, \ldots, \ell^2 - 1$ so that $x_{i+\frac{\ell^2-1}{2}} = x_i$, using the ℓ-th division polynomial (note that we do not compute the y_i here). Let $S_0 = O_{E_1}$.
3 Compute the x-coordinates $\mathbf{x}_i(x, y, y_i)$ for $1 \le i \le \frac{\ell^2-1}{2}$ of the maps representing point addition $(x, y) + S_i$ on E_1, using the values of x_i computed in step 2 but leaving y_i's as indeterminates. Set $\bar{\mathbf{x}}_i(x, y, y_i) = \mathbf{x}_i(x, y, -y_i)$ which is the x-coordinate of the point addition $(x, y) + (-S_i)$.
4 $N(x) \leftarrow P(x)$ and $D(x) \leftarrow 1$.
5 **For** $i = 1, \ldots, \frac{\ell^2-1}{2}$ **do**
6 Compute $P(\mathbf{x}_i(x, y, y_i))$ and $P(\bar{\mathbf{x}}_i(x, y, y_i))$ (as rational functions in x, y and y_i) using Horner's algorithm.
7 Compute the numerator N_i and denominator D_i of $P(\mathbf{x}_i)P(\bar{\mathbf{x}}_i)$ as polynomials in x, y and y_i.
8 Replace y^2 with $W_1(x)$ and y_i^2 with $W_1(x_i)$ in N_i. Denote the result by $N_i(x)$, as no y's or y_i's should remain.
9 Replace y_i^2 with $W_1(x_i)$ in D_i. Denote the result by $D_i(x)$, as no y's or y_i's should remain.
10 $N(x) \leftarrow N(x) \cdot N_i(x)$, and $D(x) \leftarrow D(x) \cdot D_i(x)$.
11 $N_P(x) \leftarrow \frac{N(x)}{D(x)}$, $i \leftarrow 0$, $p(x) \leftarrow 0$.
12 **For** $i = 0, \ldots, \deg(P(x))$ **do**
13 $a_i \leftarrow N_P(\zeta)$.
14 $p(x) \leftarrow p(x) + a_i x^i$.
15 $N_P(x) \leftarrow N_P(x) - a_i x^i$.
16 $N_P(x) \leftarrow N_P(x)/X_1(x)$.
17 **Return** $p(x)$.

Algorithm 11.2: Division by $[\ell]$.

Input: Elliptic curves E_1, E_2, rational maps $F(x)$ and $G(x)$ where $([\ell]\varphi)(x, y) = (F(x), G(x)y)$ for some isogeny $\varphi : E_1 \to E_2$.
Output: Rational maps $f(x)$ and $g(x)$ such that $\varphi(x, y) = (f(x), g(x)y)$.

1 Determine c_F, and the monic polynomials $P(x)$ and $Q(x)$ such that $F(x) = \frac{c_F \cdot P(x)}{W_1(x) \cdot Q(x)} (\ell = 2)$ or $F(x) = \frac{c_F \cdot P(x)}{(\psi_{E_1,\ell}(x))^2 \cdot Q(x)} (\ell \neq 2)$.
2 Compute $X_1(x)$ and $Y_2(x)$.
3 Compute $p(x) \leftarrow P(x)\big|X_1$ using Algorithm 11.1 on input $E_1, P(x), X_1(x)$.
4 Compute $q(x) \leftarrow Q(x)\big|X_1$ using Algorithm 11.1 on input $E_1, Q(x), X_1(x)$. In this step we can skip Steps 1–4 in Algorithm 11.1 since they were already performed in Step 3 of this algorithm.
5 Compute $p_0(x) \leftarrow p(x)^{1/\ell^2}$ and $q_0(x) \leftarrow q(x)^{1/\ell^2}$ using a truncated variant of Newton's method.
6 $f(x) \leftarrow c_F \ell^2 \cdot \frac{p_0(x)}{q_0(x)}$, $g(x) \leftarrow \frac{G(x)}{Y_2(f(x))}$.
7 **Return** $f(x), g(x)$.

Let $P(x) = x^{18} + 122x^{16} + 136x^{14} + 65x^{12} + 29x^{10} + 150x^8 + 114x^6 + 43x^4 + 57x^2 + 178$. We compute $p(x) = P(x)\big|X_1$ using Algorithm 11.1 as follows.

In Steps 1 and 2, we may choose $\zeta = 0$. Let $\mathbb{F}_{p^4}$ be generated by $\mathbf{a}$ having minimal polynomial $x^4 + x^2 + 109x + 2$. We obtain $S_0 = O_{E_{1728}}, S_1 = (103, y_1), S_2 = (76, y_2), S_3 = (24\mathbf{a}^3 + 39\mathbf{a}^2 + 119\mathbf{a} + 102, y_3), S_4 = (155\mathbf{a}^3 + 140\mathbf{a}^2 + 60\mathbf{a} + 77, y_4), S_5 = -S_1, S_6 = -S_2, S_7 = -S_3, S_8 = -S_4$. In Steps 3, we compute $\mathbf{x}_i(x, y, y_i)$

and $\bar{\mathbf{x}}_i(x,y,y_i)$ as $\mathbf{x}_0 = x, \bar{\mathbf{x}}_i = \mathbf{x}_i(x,y,-y_i), \forall i, 1 \le i \le 4$ where

$$\mathbf{x}_1(x,y,y_1) = \frac{-x^3+y^2-2yy_1+y_1^2-76x^2+48x+68}{x^2-27x+48},$$
$$\mathbf{x}_2(x,y,y_2) = (-x^3+y^2-2yy_2+y_2^2+76x^2+48x-68)/(x^2+27x+48),$$
$$\mathbf{x}_3(x,y,y_3) = \frac{-x^3+y^2-2yy_3+y_3^2+(24\mathbf{a}^3+39\mathbf{a}^2-60\mathbf{a}-77)x^2-46x+(30\mathbf{a}^3+4\mathbf{a}^2-75\mathbf{a}+38)}{(x^2+(-48\mathbf{a}^3-78\mathbf{a}^2-59\mathbf{a}-25)x-46)},$$
$$\mathbf{x}_4(x,y,y_4) = \frac{-x^3+y^2-2yy_4+y_4^2+(-24\mathbf{a}^3-39\mathbf{a}^2+60\mathbf{a}+77)x^2-46x+(-30\mathbf{a}^3-4\mathbf{a}^2+75\mathbf{a}-38}{x^2+(48\mathbf{a}^3+78\mathbf{a}^2+59\mathbf{a}+25)x-46}.$$

In Steps 4–11: We compute the norm $N_P(x)$ of $P(x)$ by first computing $P(\mathbf{x}_i)\cdot P(\bar{\mathbf{x}}_i) = \frac{N_i}{D_i}, 1 \le i \le 4$. We then have $N(x) = P(x)\prod_i N_i = 14x^{162}+157x^{160}+\cdots+22x^2+165$ and $D(x) = \prod_i D_i = x^{144}+107x^{142}+\cdots+90x^2+75$. Hence $N_P(x) = \frac{N(x)}{D(x)}$. Finally, we compute all the coefficients of $p(x)$ by repeating Steps 13–16. The result is

$$p(x) = x^{18}+170x^{16}+36x^{14}+95x^{12}+126x^{10}+53x^8+84x^6+143x^4+9x^2+178.$$

Example 11.3 (**Division by** $\ell = 3$ via Algorithm 11.2). As before, let $p = 179$ and $E_{1728} : y^2 = x^3 - x$ the supersingular elliptic curve over $\overline{\mathbb{F}}_p$ of j-invariant $j(E_{1728}) = 1728$ as in Example 11.2. Then the endomorphism ring of E_{1728} contains the endomorphism $[i]$ defined as $[i](x,y) := (-x, iy)$ with $i \in \mathbb{F}_{p^2}$ and $i^2 = -1$.

The map $\theta = 1 + [i]$ is a separable endomorphism and we have $([3]\theta)(x,y) = \left(\frac{F_1(x)}{F_2(x)}, \frac{G_1(x)}{G_2(x)}y\right)$, defined over $\mathbb{F}_{p^2}$, with

$$F_1(x) = 169ix^{18}+33ix^{16}+72ix^{14}+66ix^{12}+68ix^{10}+111ix^8+113ix^6+107ix^4+146ix^2+10i$$
$$F_2(x) = x^{17}+8x^{15}+45x^{13}+124x^{11}+110x^9+124x^7+45x^5+8x^3+x$$
$$G_1(x) = (58i+58)x^{26}+(170i+170)x^{24}+\cdots+(170i+170)x^2+58i+58,$$
$$G_2(x) = x^{26}+12x^{24}+2x^{22}+66x^{20}+128x^{18}+44x^{16}+171x^{14}+44x^{12}+128x^{10}+66x^8+2x^6+12x^4+x^2.$$

We apply Algorithm 11.2 to divide $[3]\theta$ by 3 to obtain $\theta = [f(x), g(x)y]$ as follows.

In Step 1, we write $F(x) = \frac{c_F \cdot P(x)}{(\psi_{E_{1728},3}(x))^2 \cdot Q(x)}$ where $c_F = 169i$, $\psi_{E_{1728},3}(x) = x^4+177x^2+119$ and

$$P(x) = x^{18}+122x^{16}+\cdots 57x^2+178, \qquad Q(x) = x^9+12x^7+30x^5+143x^3+9x.$$

In Step 2, we compute X_1 and Y_2 using the formula for multiplication by 3 map on E_{1728}. Here, X_1 is as given in Example 11.2 and

$$Y_2 = \frac{126x^{12}+92x^{10}+153x^8+136x^6+139x^4+63x^2+159}{x^{12}+173x^{10}+11x^8+175x^6+56x^4+59x^2+53}.$$

Then we compute $p(x) = P(x)\big|X_1$ and $q(x) = Q(x)\big|X_1$ in Steps 3 and 4 using Algorithm 11.1 to obtain $p(x) = x^{18}+170x^{16}+\cdots+9x^2+178$, and $q(x) = x^9$. In Step 5, computing 9-th roots of $p(x)$ and $q(x)$ yields $p_0(x) = x^2+178$ and $q_0(x) = x$. The final output is

$$f(x) = c_F\ell^2 \cdot \frac{p_0(x)}{q_0(x)} = \frac{89ix^2+90i}{x}, \quad g(x) = \frac{G(x)}{Y_2(f(x))} = \frac{(134i+134)x^2+134i+134}{x^2}.$$

To determine the complexity of Algorithm 11.1, we first prove the following lemma which is needed in the proof of Proposition 11.5.

Lemma 11.4. *Fix $0 \le i \le \frac{\ell^2-1}{2}$, the products $\mathbf{x}_i\bar{\mathbf{x}}_i$ and $P(\mathbf{x}_i)P(\bar{\mathbf{x}}_i)$ are rational functions in x, y^2, and y_i^2.*

Proof. By direct computation, both $\mathbf{x}_i + \bar{\mathbf{x}}_i$ and $\mathbf{x}_i\bar{\mathbf{x}}_i$ are rational functions in x, y^2, and y_i^2. As a symmetric polynomial in $\mathbf{x}_i$ and $\overline{\mathbf{x}}_i$, the quantity $P(\mathbf{x}_i)P(\overline{\mathbf{x}}_i)$ is a polynomial in $\mathbf{x}_i + \bar{\mathbf{x}}_i$ and $\mathbf{x}_i\bar{\mathbf{x}}_i$, hence also a rational function in x, y^2 and y_i^2. ∎

Proposition 11.5. *Algorithm 11.1 is correct and has runtime $O(\deg^2(P)\boldsymbol{M}(p^m))$.*

Proof. Algorithm 11.1 is correct by [38, Pages 8–9] and Lemma 11.4. Steps 1-3 are negligible because they require a fixed number of operations in an extension of $\mathbb{F}_{p^2}$ of degree $O(\ell^2)$. Since $P(x) \in \mathbb{F}_{p^m}[x]$ and $E_1[\ell]$ is defined over an extension of $\mathbb{F}_{p^2}$ of degree at most ℓ^2 by Lemma 2.3, all the arithmetic in the remaining steps takes place in a field extension of $\mathbb{F}_{p^2}$ of degree $\mathrm{lcm}(\ell^2, m) = O(m)$.

In the first loop (steps 5-10), the most costly steps are 7 and 10 which both require $O(\deg^2(P))$ operations; the remaining steps are linear in $\deg P$ when Horner's algorithm is used. In the second loop (steps 12-11), $p(x)$ is computed as described in [38, Page 9]. Step 13 requires $O(\deg P)$ field operations using Horner's algorithm again. Since X_1 has degree $O(\ell^2)$, step 11 also takes $O(\deg P)$ operations. Hence the second loop takes $O(\deg^2(P))$ field operations. □

Proposition 11.6. *Algorithm 11.2 is correct and has runtime* $O(\deg^2(\varphi)\boldsymbol{M}(p))$.

Proof. The correctness of Algorithm 11.2 follows from [38, Proposition 2.6]. By Lemma 2.2, φ is defined over $\mathbb{F}_{p^{12}}$, so all the rational functions appearing in the algorithm belong to $\mathbb{F}_{p^{12}}(x)$. We also note that $P(x)$ and $Q(x)$ have degree $O(\deg \varphi)$, hence so do $p(x)$, $q(x)$, $p_0(x)$ and $q_0(x)$.

Since $\psi_{E_1,\ell}(x)$ and $W_1(x)$ have fixed degree, step 1 requires $O(\deg \varphi)$ field operations. Steps 5 and 6 take $\widetilde{O}(\deg \varphi)$ operations using fast polynomial arithmetic; see [26, Theorem 1.2]. Here, to extract an ℓ^2-th root of $p(x)$, we apply a truncated variant of Newton's method (see [52, Sections 9.4 and 9.6]) to the polynomial $H(y) = y^{\ell^2} - p(x)$ and compute the sequence of polynomials

$$f_0(x) = x^{\deg p}\ , \qquad f_{i+1}(x) = f_i(x) - \left\lfloor \frac{H(f_i(x))}{H'(f_i(x))} \right\rfloor \quad (i \geq 0)$$

to obtain $p_0(x)$ after at most $\lceil \log_2(\deg p) \rceil$ iterations; similarly for $q_0(x)$.

The runtime of Algorithm 11.2 is thus dominated by steps 3 and 4, which have runtime $O(\deg^2(\varphi)\mathbf{M}(p^{12})) = O(\deg^2(\varphi)\mathbf{M}(p))$. □

References

[1] S. Arpin. PhD thesis, University of Colorado Boulder. In preparation for May 2022.

[2] S. Arpin, M. Chen, K. E. Lauter, R. Scheidler, K. E. Stange, and H. T. N. Tran. Orientations and cycles in supersingular isogeny graphs, 2022. In preparation.

[3] S. Arpin, M. Chen, K. E. Lauter, R. Scheidler, K. E. Stange, and H. T. N. Tran. Win5 github repository, 2022. `https://github.com/SarahArpin/WIN5`.

[4] E. Bank, C. Camacho-Navarro, K. Eisenträger, T. Morrison, and J. Park. Cycles in the supersingular l-isogeny graph and corresponding endomorphisms. In *Research Directions in Number Theory*, pages 41–66. Springer, 2019.

[5] J.-F. Biasse, A. Iezzi, and M. J. Jacobson. A note on the security of csidh. In D. Chakraborty and T. Iwata, editors, *Progress in Cryptology – INDOCRYPT 2018*, pages 153–168, Cham, 2018. Springer International Publishing.

[6] R. Bröker, D. Charles, and K. Lauter. Evaluating large degree isogenies and applications to pairing based cryptography. In *Pairing-based cryptography—Pairing 2008*, volume 5209 of *Lecture Notes in Comput. Sci.*, pages 100–112. Springer, Berlin, 2008.

[7] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in cryptology—ASIACRYPT 2018. Part III*, volume 11274 of *Lecture Notes in Comput. Sci.*, pages 395–427. Springer, Cham, 2018.

[8] D. X. Charles, K. E. Lauter, and E. Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

[9] M. Chenu and B. Smith. Higher-degree supersingular group actions. *Mathematical Cryptology*, 2021.

[10] K. K. H. Cheung and M. Mosca. Decomposing finite abelian groups. *Quantum Info. Comput.*, 1(3):26–32, Oct 2001.

[11] A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Math. Cryptol.*, 8(1):1–29, 2014.

[12] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[13] L. Colò and D. Kohel. Orienting supersingular isogeny graphs. *J. Math. Cryptol.*, 14(1):414–437, 2020.

[14] J.-S. Coron and D. Naccache. Security analysis of the gennaro-halevi-rabin signature scheme. In *EUROCRYPT*, 2000.

[15] J.-M. Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. `https://ia.cr/2006/291`.

[16] D. A. Cox. *Primes of the form* $x^2 + ny^2$. Pure and Applied Mathematics (Hoboken). John Wiley & Sons, Inc., Hoboken, NJ, second edition, 2013. Fermat, class field theory, and complex multiplication.

[17] E. Croot, A. Granville, R. Pemantle, and P. Tetali. On sharp transitions in making squares. *Annals of Mathematics*, 175(3):1507–1550, 2012.

[18] L. De Feo. Mathematics of isogeny based cryptography. 2017. `https://arxiv.org/abs/1711.04062`.

[19] L. De Feo, C. Delpech de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, and B. Wesolowski. *Séta: Supersingular Encryption from Torsion Attacks*, pages 249–278. Advances in Cryptology – ASIACRYPT 2021. Springer International Publishing, Cham, 2021.

[20] L. De Feo, J. Kieffer, and B. Smith. Towards practical key exchange from ordinary isogeny graphs. In *Advances in cryptology—ASIACRYPT 2018. Part III*, volume 11274 of *Lecture Notes in Comput. Sci.*, pages 365–394. Springer, Cham, 2018.

[21] V. de Quehen, P. Kutas, C. Leonardi, C. Martindale, L. Panny, C. Petit, and K. E. Stange. *Improved Torsion-Point Attacks on SIDH Variants*, pages 432–470. Advances in Cryptology – CRYPTO 2021. Springer International Publishing, Cham, 2021.

[22] K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison, and C. Petit. Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In *Advances in cryptology—EUROCRYPT 2018. Part III*, volume 10822 of *Lecture Notes in Comput. Sci.*, pages 329–368. Springer, Cham, 2018.

[23] N. D. Elkies. The existence of infinitely many supersingular primes for every elliptic curve over $\mathbf{Q}$. *Invent. Math.*, 89(3):561–567, 1987.

[24] F. Fité and A. V. Sutherland. Sato-Tate groups of $y^2 = x^8 + c$ and $y^2 = x^7 - cx$. In *Frobenius distributions: Lang-Trotter and Sato-Tate conjectures*, volume 663 of *Contemp. Math.*, pages 103–126. Amer. Math. Soc., Providence, RI, 2016.

[25] S. D. Galbraith, C. Petit, and J. Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *J. Cryptology*, 33(1):130–175, 2020.

[26] D. Harvey and J. van Der Hoeven. Polynomial multiplication over finite fields in time $O(n \log n)$. 2019. `https://hal.archives-ouvertes.fr/hal-02070816/document`.

[27] S. Ionica and A. Joux. Pairing the volcano. In *Algorithmic number theory*, volume 6197 of *Lecture Notes in Comput. Sci.*, pages 201–208. Springer, Berlin, 2010.

[28] M. Kaneko. Supersingular j-invariants as singular moduli mod p. *Osaka J. Math.*, 26(4):849–855, 1989.

[29] J. Kieffer. Accelerating the couveignes rostovtsev stolbunov key exchange protocol. Master's thesis, l'Université Paris IV, 2018. `https://arxiv.org/pdf/1804.10128.pdf`.

[30] D. E. Knuth. *The art of computer programming. Vol. 2*. Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley Publishing Co., Reading, Mass., second edition, 1981. Seminumerical algorithms.

[31] D. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 2014.

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

[33] P. Kutas, S.-P. Merz, C. Petit, and C. Weitkaemper. One-way functions and malleability oracles: hidden shift attacks on isogeny-based protocols. In A. Canteaut and F. Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, Lecture Notes in Computer Science, pages 242–271. Springer, June.

[34] P. Kutas, S.-P. Merz, C. Petit, and C. Weitkämper. *One-Way Functions and Malleability Oracles: Hidden Shift Attacks on Isogeny-Based Protocols*, pages 242–271. Advances in Cryptology – EUROCRYPT 2021. Springer International Publishing, Cham, 2021.

[35] J. Love and D. Boneh. Supersingular curves with small non-integer endomorphisms, 2020. `https://arxiv.org/abs/1910.03180`.

[36] J. Love and D. Boneh. Supersingular curves with small noninteger endomorphisms. In *ANTS XIV—Proceedings of the Fourteenth Algorithmic Number Theory Symposium*, volume 4 of *Open Book Ser.*, pages 7–22. Math. Sci. Publ., Berkeley, CA, 2020.

[37] K. McMurdy. `https://phobos.ramapo.edu/~kmcmurdy/research/SAGE_ssEndos/`. Accessed Jan 10, 2022.

[38] K. McMurdy. Explicit representation of the endomorphism rings of supersingular elliptic curves. `https://phobos.ramapo.edu/~kmcmurdy/research/McMurdy-ssEndoRings.pdf`, 2014.

[39] H. Onuki. On oriented supersingular elliptic curves. *Finite Fields and their Applications*, 69, 2021.

[40] A. K. Pizer. Ramanujan graphs and Hecke operators. *Bull. Amer. Math. Soc. (N.S.)*, 23(1):127–137, 1990.

[41] G. Robin. Grandes valeurs de la fonction somme des diviseurs et hypothèse de Riemann. *J. Math. Pures Appl. (9)*, 63(2):187–213, 1984.

[42] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6:64–94, 1962.

[43] N. T. Sardari. Diameter of Ramanujan graphs and random Cayley graphs. *Combinatorica*, 39(2):427–446, 2019.

[44] R. Schoof. Four primality testing algorithms. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 101–126. Cambridge Univ. Press, Cambridge, 2008.

[45] D. Shumow. Isogenies of elliptic curves: a computational approach. Master's thesis, University of Washington, 2009. `https://arxiv.org/abs/0910.5370`.

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

[47] K. E. Stange. Frobenius and the endomorphism ring of $j = 1728$, 2021. `http://math.colorado.edu/~kstange/papers/1728.pdf`.

[48] G. Tenenbaum. On ultrafriable integers. *Q. J. Math.*, 66(1):333–351, 2015.

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

[50] J. Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.

[51] J. Voight. *Quaternion algebras*, volume 288 of *Graduate Texts in Mathematics*. Springer, Cham, [2021] ©2021.

[52] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, third edition, 2013.

[53] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992.

[54] B. Wesolowski. Orientations and the supersingular endomorphism ring problem. Cryptology ePrint Archive, Report 2021/1583, 2021. `https://iacr.org/2021/1583`.

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

[56] H. C. Williams and H. te Riele. New computations concerning the Cohen-Lenstra Heuristics. *Experimental Mathematics*, 12(1):99 – 113, 2003.

[57] T. Yang. Minimal cm liftings of supersingular elliptic curves. *Pure and applied mathematics quarterly*, 4(4):1317–1326, 2008.

Department of Mathematics, University of Colorado, Campus Box 395, Boulder, Colorado 80309-0395
Email address: `Sarah.Arpin@colorado.edu`

Department of Mathematics, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0112
Email address: `mic181@ucsd.edu`

Facebook AI Research, Meta, Seattle, WA
Email address: `klauter@fb.com`

Department of Mathematics and Statistics, University of Calgary, 2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
Email address: `rscheidl@ucalgary.ca`

Department of Mathematics, University of Colorado, Campus Box 395, Boulder, Colorado 80309-0395
Email address: `kstange@math.colorado.edu`

Department of Mathematical and Physical Sciences, Concordia University of Edmonton, 7128 Ada Blvd NW, Edmonton, AB T5B 4E4, Canada
Email address: `hatran1104@gmail.com`