Algorithms for solving the isogeny problem using orientations

Maria Corte-Real Santos[1,2], Arthur Herlédan Le Merdy[3], Joseph Macula[4,5], Michael Meyer[6], Travis Morrison[7] and Eli Orvis[5,8]

[1] École Normale Supérieure de Lyon, France
[2] CNRS, France
[3] COSIC, KU Leuven, Belgium
[4] Wake Forest University, USA
[5] University of Colorado Boulder, USA
[6] University of Regensburg, Germany
[7] Virginia Tech, USA
[8] University of Calgary, Canada

Abstract. We introduce WayFinder, a framework for generalizing the Delfs–Galbraith and SuperSolver algorithms for the supersingular isogeny problem. Our framework extends the search for elliptic curves with an orientation by an order containing $\mathbb{Z}[\ell\sqrt{-p}]$ to more general orders, and we derive a heuristic cost model for such generalisations. Our cost model not only works in a more general context, but also provides more accurate predictions when applied to SuperSolver. We instantiate WayFinder for orders containing $\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ where ℓ_i are 1 or primes such that the modular curve $X_0(\ell_2)$ has genus 0. We then introduce a low-storage algorithm for computing an isogeny between two oriented supersingular elliptic curves, even when the curves are oriented by distinct orders. Together, these provide an algorithm that improves on the state of the art for solving the isogeny problem, and a cost model with potential applications to parameter selection in isogeny-based cryptography.

Keywords: Isogeny-based cryptography · Delfs–Galbraith algorithm · Elliptic curves

1 Introduction

The most general form of the *supersingular isogeny problem* asks, given a prime p and supersingular elliptic curves E_1 and E_2 defined over $\mathbb{F}_{p^2}$, to compute an isogeny $\varphi : E_1 \to E_2$. The security of most isogeny-based cryptographic schemes relies on the presumed hardness of this problem. In particular, the supersingular isogeny problem is equivalent to the *endomorphism ring problem* [Wes22b, MW25], making the best algorithms to solve the isogeny problem essential to understanding the security of isogeny-based schemes such as SQIsign [AAA+25] (in Round 3 of NIST's alternate call for signature schemes).

The current state-of-the-art algorithm for solving the isogeny problem is SuperSolver, introduced in 2022 by Corte-Real Santos, Costello, and Shi [CRSCS22]. SuperSolver provides concrete improvements on the 2016 $\tilde{O}(p^{1/2})$ algorithm of Delfs and Galbraith [DG16] by using a *fast subfield root detection* algorithm. The authors predict that this improvement provides an 8 to 13 times speed-up over Delfs–Galbraith for cryptographically sized primes.

E-mail: maria.corte_real_santos@ens-lyon.fr (Maria Corte-Real Santos), arthur.herledanlemerdy@esat.kuleuven.be (Arthur Herlédan Le Merdy), maculaj@wfu.edu (Joseph Macula), michael@random-oracles.org (Michael Meyer), tmo@vt.edu (Travis Morrison), eli.orvis@ucalgary.ca (Eli Orvis)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2026-11-22.

In both the original Delfs–Galbraith algorithm and SuperSolver, navigating to the subgraph consisting of the curves defined over $\mathbb{F}_p$ is the bottleneck step. This subgraph can be defined in the language of *orientations*, as the subgraph of curves that admit an embedding $\mathbb{Z}[\sqrt{-p}] \hookrightarrow \mathrm{End}(E)$. In this light, it is natural to ask whether one can replace the $\mathbb{F}_p$-subgraph with the subgraph of curves admitting an embedding by a different order. An idea in this vein was first suggested in 2021 by Chenu and Smith [CS21]. Specifically, they consider replacing the $\mathbb{F}_p$-subgraph with the subgraph consisting of those curves with an orientation by $\mathbb{Z}[\sqrt{-dp}]$ for a finite set S of squarefree integers d coprime to p. The resulting generalized Delfs–Galbraith algorithm's first step would consist of taking random walks in the 2-isogeny graph until one finds curves E_1 and E_2 with respective orientations $\mathbb{Z}[\sqrt{-d_1 p}]$ and $\mathbb{Z}[\sqrt{-d_2 p}]$ for $d_1, d_2 \in S$.

Chenu and Smith conclude that this approach likely does not offer an asymptotic speed-up to the Delfs–Galbraith algorithm, identifying two difficulties. First, one must devise a method to determine whether a given curve E in the supersingular isogeny graph is a vertex in the desired subgraph, i.e., whether E has an orientation by $\mathbb{Z}[\sqrt{-dp}]$ for some $d \in S$. For the case $d = 1$ this is trivial: simply check whether the j-invariant is in $\mathbb{F}_p$. Chenu and Smith note that the orders $\mathbb{Z}[\sqrt{-dp}]$ can be detected using modular polynomials, but conclude that the computational cost outweighs any potential speed-ups. Second, one must now find an isogeny between two elliptic curves E_1 and E_2 with knowledge of orientations on each by potentially different orders.

Contributions. The first main contribution of this paper is a new generalisation of the Delfs–Galbraith algorithm, called WayFinder, that overcomes the difficulties laid out by Chenu and Smith. It is parameterised by a set $\mathcal{R}$ of quadratic orders R. Given supersingular E_1, E_2 defined over $\mathbb{F}_{p^2}$, it works in two steps:

1. Find isogenies $\varphi_1 : E_1 \to E_1'$ and $\varphi_2 : E_2 \to E_2'$, with E_1', E_2' oriented by $R_1, R_2 \in \mathcal{R}$ using depth-first search, terminating in expected time $\widetilde{O}(p^{1/2})$.

2. Find an isogeny $\psi : E_1' \to E_2'$ by exploiting the extra information given by the orientations. This terminates in expected time $\widetilde{O}(p^{1/4})$.

Framing the first step of the Delfs–Galbraith algorithm as finding paths to curves oriented by an order containing $\mathbb{Z}[\sqrt{-p}]$ and the first step of SuperSolver as finding paths to curves oriented by an order containing $\mathbb{Z}[\ell\sqrt{-p}]$, we can immediately cast these as a special case of WayFinder. We detail the first step in Section 3, and the second step in Section 4.

The key challenge of the first step is to find the set $\mathcal{R}$ that minimises the cost of the algorithm for a given p. This involves balancing the improvements obtained by taking orders with larger discriminants against the increased cost of detection. We analyse the situation for general orders in Section 3. Using this cost analysis, we reformulate the analogous optimisation problem in [CRSCS22] as the problem of choosing an optimal set of orientations depending only on p. We define an objective function that, when specialised to orientations by orders containing $\mathbb{Z}[\ell\sqrt{-p}]$, is similar to the SuperSolver objective function. However, there are important differences. Not only does our cost model allow us to analyse these more general algorithms, it also more accurately predicts the improvement of SuperSolver over Delfs–Galbraith.

The first key difference between our cost model and that of [CRSCS22] is that we model the global behaviour of the algorithm, giving a precise estimate of the total number of nodes visited in terms of Hurwitz class numbers, rather than relying on the "nodes revealed" metric of SuperSolver, a local metric that is in many ways overly optimistic. A second key difference is that our study of depth-first search for structured subgraphs predicts the impact of *clustering* on the complexity of the first stage in SuperSolver and WayFinder. In [CRSCS22] it was observed that the number of nodes visited in depth-first search is, on average, larger by a constant factor than what one would expect using random walks,

and this constant factor depends on the congruence class of p modulo 8. Our analysis in Section 3 predicts this constant factor for each residue class modulo 8, and our experiments in Appendix A confirm these predictions.

In Section 4, we present a low-storage algorithm for solving the following problem:

Problem 1. *Let E_1 and E_2 be supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ (primitively) oriented by orders R_1 and R_2. Find an isogeny $\phi : E_1 \to E_2$.*

For the case $R_1 = R_2 = \mathbb{Z}[\sqrt{-p}]$, in [DG16] Delfs and Galbraith suggest to leverage the birthday paradox to solve this problem in time $\tilde{O}(|\operatorname{disc}(R_1)|^{1/4})$, following a meet-in-the-middle approach in the graph of R_1-oriented curves with edges corresponding to a generating set of the ideal class group $\mathrm{Cl}(R_1)$. They also suggest a low-storage Pollard-ρ approach. Since then, the asymptotic complexity of this algorithm has never been improved. To the best of our knowledge, the case $R_1 \neq R_2$ has only been discussed in [CS21]. While their solution has a similar asymptotic complexity $\tilde{O}(\max \sqrt{|\operatorname{disc}(R_1)|}, \sqrt{|\operatorname{disc}(R_2)|})$, it has limitations; see Section 6.2. We compute an isogeny $E_1 \to E_2$ by computing collisions between the given oriented curves and special curves with known endomorphism rings. This allows us to efficiently compute the isogeny corresponding to each path, and to derive an isogeny between the special curves, by taking advantage of the Deuring correspondence. Our algorithm requires only polynomial storage.

In Sections 5 and 6 we give a concrete instantiation of WayFinder, which overcomes the two obstructions identified by Chenu and Smith. Section 5 details how to detect supersingular elliptic curves that are oriented by $R \supseteq \mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ where ℓ_i are 1 or primes such that the modular curve $X_0(\ell_2)$ has genus 0. These restrictions will become natural in Section 5.2. When $\ell_2 = 1$, this is SuperSolver, and when $\ell_1 = 1$ this can be done using modular polynomials. When $\ell_1, \ell_2 > 1$, we use resultants and the theory of modular curves to construct a polynomial whose rational factors identify the correct j-invariants. We then apply the algorithm RationalFactor from [CRSCS22] to $R_{\ell_1,\ell_2}(x)$. In Section 6, we instantiate our cost model for these orders and thus predict the optimal set $\mathcal{R}$ on which to run our algorithm. After performing a depth-first-search to find curves with orientations in $\mathcal{R}$, we apply our novel algorithm for finding an isogeny between oriented curves.

In Section 7 we discuss several experiments on the performance of our instantiation of WayFinder. Our results for small primes up to 44 bits show that our cost model achieves very high accuracy and is much more precise than the previous SuperSolver model. Furthermore, by allowing for pairs (ℓ_1, ℓ_2) with $\ell_2 \neq 1$ we obtain up to 2.7 times speed-up compared to SuperSolver. We remark that the observed speed-up depends significantly on the specific prime. Indeed, in some cases we obtain smaller speed-ups, and in some we confirm that SuperSolver is the optimal approach. Our cost model accurately predicts all of these cases. We also use our cost model to predict the concrete hardness of the isogeny problem for cryptographically sized primes, for instance for SQIsign parameters, and observe that WayFinder consistently outperforms SuperSolver in these cases. Our cost prediction can serve as a hardness estimator for the isogeny problem for concrete parameters used in isogeny-based schemes. Our confidence in these estimates stems from:

1. Our cost model, which predicts the main term in the expected nodes visited during depth-first search for a tree-like subgraph S, confirmed experimentally;
2. Our estimate of the size of S as a function of class numbers, again predicting the main term and confirmed via experiments;
3. The development and analysis of novel, efficient tests for orientability that lead to concrete speed-ups of the first stage;
4. Our estimates for the expectation of relevant class numbers, which allow us to accurately estimate the cost of the algorithm in parameter ranges where class numbers can no longer be easily computed, again confirmed via experiments.

Finally, our new cost model for precomputation is very flexible, in the sense that it is independent of traversing the search tree in the 2-isogeny graph or the particular orders detected. It can therefore be applied to previous works such as [CRSCS22] and [CD25] to refine their estimates, as we will see in Remark 11. We provide an accompanying implementation of all steps of our algorithm, written in SageMath [S+23] and Python. It can be found at

https://github.com/mariascrs/WayFinder

Concurrent work. During the preparation of this work, we were made aware of concurrent work on the concrete hardness of the isogeny problem [PRSU26]. At a high level, their work introduces a novel method to explore subtrees of the $\{2,3\}$-isogeny graph while incorporating ideas of Corte-Real Santos, Costello, and Shi and of Chenu and Smith. The authors use these ideas to produce a highly optimised implementation of SuperSolver and the second stage of the Delfs–Galbraith algorithm. Our work instead opens the aperture, introducing a framework to study generalisations of the Delfs–Galbraith algorithm as depth-first search for orientations, along with an accurate model to analyse its cost. We further make stage two compatible with our modified first stage, where the curves may have different orientations. We expect that our cost model could incorporate their method for traversing the search tree, and that our tests for orientability would be compatible with their algorithm. We expect that these orthogonal contributions could (with some effort) be combined in future work to produce a faster algorithm and thus more accurate predictions for the bit-security of isogeny-based schemes.

Later, upon completion of this work, Wesolowski announced a new attack against the supersingular isogeny problem that runs in time and memory $p^{1/3+o(1)}$ [Wes26]. In contrast, our algorithm is memory-free, but runs in time $\widetilde{O}(p^{1/2})$. However, the concrete performance of the new method and its impact on the security of isogeny-based schemes is still yet to be established due to the significant memory costs. If the memory requirements remain infeasible with current technology, our framework may still find use in choosing secure parameters.

Acknowledgements. This project started at the 2025 Banff workshop on isogeny-based cryptography, and we want to thank the organisers and BIRS for a successful workshop. We also thank David Jao for his ideas on using modular curves for detecting orientations of the form $\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ for $\ell_1 > 1$, and for useful discussions throughout the project. The authors further thank Benjamin Wesolowski for helpful discussions, particularly on the second step of WayFinder. We also thank Sebastian Spindler for pointing out a typo in the statement of Proposition 1. This work is supported by the HQI initiative (ANR-22-PNCQ-0002), by the PEPR quantique France 2030 programme (ANR-22-PETQ-0008), by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (under grant agreement ISOCRYPT - No. 101020788, and grant agreement AGATHA CRYPTY - No. 101116169), by the Research Council KU Leuven grant C14/24/099, by CyberSecurity Research Flanders with reference number VOEWICS02, by the German Federal Ministry of Research, Technology and Space (BMFTR) under the project QUORYPTAN (16KIS2034), and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 577658121 (IMPACT). Travis Morrison was supported by the National Science Foundation under Grant No. 2340564. Part of this work was done while Arthur Herlédan Le Merdy was at ENS de Lyon.

2 Preliminaries

Throughout, we fix a prime $p > 3$.

Modular polynomials. For any positive integer m, there exists a two variable polynomial $\Phi_m(X, Y) \in \mathbb{Z}[X, Y]$ called the *classical modular polynomial of level* m (or simply *modular polynomial*), which gives a singular plane model of the modular curve $X_0(m)$. More specifically, for any algebraically closed field k in which m is invertible, the polynomial $\Phi_m(X, Y)$ vanishes on input $j_1, j_2 \in k$ if and only if any elliptic curve with j-invariant j_1 has a cyclic degree m-isogeny to any elliptic curve with j-invariant j_2. The polynomial Φ_m is available for most m up to 1,000 by work of Bröker, Lauter, and Sutherland [BLS12].

Isogeny graphs and orientations. Let $\ell > 0$ be an integer with $p \nmid \ell$. We write $G(p, \ell)$ for the supersingular ℓ-isogeny graph, whose vertex set $V(G)$ is the set of j-invariants of supersingular elliptic curves defined over $\overline{\mathbb{F}_p}$ and whose edges are ℓ-isogenies defined over $\overline{\mathbb{F}_p}$. Note that, as we consider only supersingular elliptic curves E, every vertex $j(E)$ is defined over $\mathbb{F}_{p^2}$. We have that $\#V(G) = \lfloor p/12 \rfloor + \varepsilon$ vertices, where $\varepsilon \in \{0, 1, 2\}$ depending on the value of p modulo 12. When ℓ is prime, $G(p, \ell)$ is connected, and is an $(\ell + 1)$-regular Ramanujan graph. Due to the existence of dual isogenies, we can further think of $G(p, \ell)$ being undirected, except for a small number of special vertices. The set $V(G)$ also contains the set $V(\mathbb{F}_p)$ of j-invariants that are defined over the subfield $\mathbb{F}_p$. We have $\#V(\mathbb{F}_p) = \tilde{O}(\sqrt{p})$. An *orientation* of a supersingular elliptic curve E by a quadratic order R with fraction field K is an embedding $\iota : K \hookrightarrow \mathrm{End}(E) \otimes \mathbb{Q}$ such that $\iota(R) \subset \mathrm{End}(E)$. The orientation is said to be *primitive* if $\iota(K) \cap \mathrm{End}(E) = \iota(R)$. An isogeny $\varphi\colon E \to E'$ between two curves with orientations ι, ι' by R is *oriented* or an R-isogeny if $\varphi \circ \iota(\alpha) = \iota'(\alpha) \circ \varphi$ for all $\alpha \in R$. We let $\mathrm{SS}(R)$ denote the set of isomorphism classes of R-oriented supersingular elliptic curves. For more on orientations we refer the reader to [CK20, Onu21, ACL+24].

Isogeny Volcanoes. Kohel first showed that ordinary elliptic curves and ℓ-isogenies form *volcanoes*: graphs with a single cycle with trees descending from this "rim" [Koh96]. Fouquet and Morain showed that these graphs have computational applications in [Fou01, FM02]. The concept was then generalised to the context of oriented supersingular elliptic curves in [CK20]: the connected components of the graph whose vertex set is $\mathrm{SS}(R)$ and whose edges are ℓ-oriented isogenies are ℓ-volcanoes. For more details on isogeny volcanos, we refer the reader to [Sut13].

The Delfs–Galbraith attack. The main idea of the Delfs–Galbraith algorithm for solving the supersingular isogeny problem is to reduce the problem of finding an isogeny over $\mathbb{F}_{p^2}$ to an isogeny finding problem over the subfield $\mathbb{F}_p$. Given E_1, E_2 with j-invariants defined over $\mathbb{F}_{p^2} \backslash \mathbb{F}_p$, the first step of the algorithm attempts to find isogenies to curves defined over $\mathbb{F}_p$, i.e. isogenies $\varphi_1 : E_1/\mathbb{F}_{p^2} \to E_1'/\mathbb{F}_p$ and $\varphi_2 : E_2/\mathbb{F}_{p^2} \to E_2'/\mathbb{F}_p$. The second step of the algorithm solves the isogeny finding problem between the subfield curves E_1' and E_2', by finding $\psi : E_1' \to E_2'$. Then, the isogeny $\varphi : E_1 \to E_2$ with $\varphi = \hat{\varphi}_2 \circ \psi \circ \varphi_1$ solves the original isogeny finding problem. In terms of complexity, the first step is the bottleneck of the algorithm, and has an asymptotic complexity of $\tilde{O}(p^{1/2})$. In contrast, the second step of finding the isogeny ψ between subfield curves has a complexity of $\tilde{O}(p^{1/4})$.

Solver and SuperSolver. Corte-Real Santos, Costello, and Shi [CRSCS22] analysed how to implement and optimize the first step of the Delfs–Galbraith algorithm. Their first algorithm, called Solver, is a direct instantiation of the approach outlined by Delfs and Galbraith, using a depth-first search in the 2-isogeny graph from $E_1/\mathbb{F}_{p^2}$ until finding a subfield curve $E_1'/\mathbb{F}_p$. For efficiency reasons, Solver performs steps in the 2-isogeny graph via computing the j-invariants of neighbouring 2-isogenous curves through modular polynomials. Each step in the non-backtracking walk reveals both 2-isogenous neighbours, which are stored such that Solver explores a binary tree of fixed depth until finding a subfield curve.

SuperSolver extends this approach by choosing a set of degrees ℓ, and additionally checking if curves that appear during the 2-isogeny walk have ℓ-isogenous neighbours

defined over the subfield $\mathbb{F}_p$, without explicitly computing these neighbours. The authors achieve this by using an efficient gcd computation to check whether $\Phi_\ell(j, x)$ has roots defined over $\mathbb{F}_p$, for each j encountered during the walk. For primes ℓ, this reveals the field of definition of the $\ell + 1$ neighbouring j-invariants for a cost of $O(\ell^2)$ multiplications in $\mathbb{F}_p$. For a given prime p, SuperSolver precomputes which set of (not necessarily prime) ℓ to use for the detection, such that the cost per curve that is checked is minimised.

Finding rational factors of a polynomial. Detecting $\mathbb{F}_p$-rational factors of $\Phi_\ell(j, x)$ can be done by computing $\gcd(x^{p^k} - x, \Phi_\ell(j, x))$ for $k = 1, 2$, but the expected cost of $\tilde{O}(\ell^2 \log p)$ operations in $\mathbb{F}_p$ is prohibitively expensive in SuperSolver. Instead, SuperSolver uses an efficient gcd approach: Given $f(X) = a_n X^n + a_{n-1} X^{n-1} + \ldots a_1 X + a_0 \in \mathbb{F}_{p^2}[X]$ and $\beta \in \mathbb{F}_{p^2}$ such that $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$ and $\beta^2 \in \mathbb{F}_p$, we define $\mathrm{Re}(a + b\beta) = a$ and $\mathrm{Im}(a + b\beta) = b$, analogously to the notation of complex numbers. We also define the polynomials $f_1(X) = \mathrm{Re}(a_n)X^n + \cdots + \mathrm{Re}(a_1)X + \mathrm{Re}(a_0)$ and $f_2(X) = \mathrm{Im}(a_n)X^n + \cdots + \mathrm{Im}(a_1)X + \mathrm{Im}(a_0)$. Following [CRSCS22], f then has an $\mathbb{F}_p$-rational factor if and only if $\deg(\gcd(f_1, f_2)) \geq 1$. Note that this avoids the usage of $\mathbb{F}_{p^2}$-operations, and that the gcd can be computed efficiently without requiring inversions using the InvFreeGCD function from [CRSCS22]. We call this function RationalFactor. In [CRSCS22], InvFreeGCD was used to test if the largest $\mathbb{F}_p$-rational factor of $\Phi_\ell(j(E), Y)$ had degree 1. However, as the proposition below shows, if E is supersingular and ℓ is small relative to p, then the polynomial $\Phi_\ell(j(E), Y)$ has an $\mathbb{F}_p$-rational factor of positive degree if and only if E or one of its ℓ-neighbours is an $\mathbb{F}_p$-curve. Since we usually will use this test on a j-invariant not in $\mathbb{F}_p$, this determines whether the vertex labeled j in $G(p, \ell)$ has a neighbour in $\mathbb{F}_p$, so we call this algorithm NeighbourInFp.

Proposition 1. *Let E be supersingular and ℓ prime with $\ell^2 < p/4$. The following are equivalent:*

1. *$\Phi_\ell(j(E), Y)$ has a divisor in $\mathbb{F}_p[x]$ of positive degree;*
2. *E has an orientation by $\mathbb{Z}[\ell\sqrt{-p}]$;*
3. *either E or a curve ℓ-isogenous to E has j-invariant in $\mathbb{F}_p$.*

Proof. Suppose $\Phi_\ell(j(E), Y)$ has a $\mathbb{F}_p$-rational factor f of positive degree. The roots of $\Phi_\ell(j(E), Y)$ and hence of f are supersingular j-invariants, so these roots are elements of $\mathbb{F}_{p^2}$ and hence all have minimal polynomials of degree 1 or 2. Thus we can assume that f is irreducible over $\mathbb{F}_p[x]$ and has degree 1 or 2. If $\deg f = 1$ then a root j_0 of f is in $\mathbb{F}_p$ and hence there is an elliptic curve E_0 defined over $\mathbb{F}_p$ with j-invariant j_0 and ℓ-isogenous to E, say $\varphi\colon E \to E_0$ has degree ℓ. Then define $\theta = \widehat{\varphi} \circ \pi_{E_0} \circ \varphi$ where π_{E_0} is the Frobenius endomorphism of E_0. If instead f has degree 2 then its (distinct) roots are j_0 and j_0^p, say, and E is ℓ-isogenous to both E_0 and $E_0^{(p)}$. There are ℓ-isogenies $\varphi\colon E \to E_0$ and $\psi\colon E \to E_0^{(p)}$, so we define $\theta = \widehat{\psi} \circ \pi \circ \varphi$. The endomorphism θ has degree $\ell^2 p$ and is inseparable since E is supersingular. Thus the trace of θ must be 0 modulo p. Moreover the trace of θ is bounded by $2\sqrt{\deg \theta} = 2\ell\sqrt{p}$, so the assumption $\ell^2 < p/4$ implies the trace of θ is 0.

Now suppose E has an orientation $\iota\colon \mathbb{Z}[\ell\sqrt{-p}] \to \mathrm{End}(E)$. Let $\alpha = \iota(\ell\sqrt{-p})$. If $j(E) \in \mathbb{F}_p$ we are done so suppose $j(E) \notin \mathbb{F}_p$; we show E is isogenous to a curve with j-invariant in $\mathbb{F}_p$. Since $j(E) \notin \mathbb{F}_p$ we observe that E does not have an orientation by $\mathbb{Z}[\sqrt{-p}]$ so the orientation ι must be primitive at ℓ. There is a unique "ascending" ℓ-isogeny $\varphi\colon E \to E_0$ to a curve E_0 orientable by $\mathbb{Z}[\sqrt{-p}]$, so $j(E_0) \in \mathbb{F}_p$ as desired.

Finally suppose either E or an ℓ-neighbour of E has j-invariant in $\mathbb{F}_p$. In either case, E has an endomorphism α of degree $\ell^2 p$. Then $\ker \alpha$ contains a subgroup of order ℓ so choose one, call it K (there may be more than one), let $\varphi\colon E \to E_0$ have kernel K, and factor $\alpha = \psi \circ \varphi$. Now ψ has degree ℓp and may be factored as $\psi_{sep} \circ \pi$ where $\psi_{sep}\colon E_0^{(p)} \to E$ is

separable of degree ℓ and $\pi\colon E_0 \to E_0^{(p)}$ is the Frobenius isogeny. We see that j_0 and j_0^p are roots of $\varphi_\ell(j(E), Y)$. If $j_0 \in \mathbb{F}_p$ then $\Phi_\ell(j(E), Y)$ has the rational factor $Y - j_0$, and if not then it has the rational factor $(Y - j_0)(Y - j_0^p)$. □

Precomputation. To minimise the cost of running SuperSolver, the algorithm runs a precomputation to determine the optimal list $\mathcal{L}$ of ℓ's for which to run the detection. For this, SuperSolver models the cost of the search as follows. Let C_ℓ denote the cost of deciding whether $\Phi_\ell(j, Y)$ has an $\mathbb{F}_p$-rational root. Let C denote the cost of taking a step in the 2-isogeny graph; this is the cost of computing a square root in $\mathbb{F}_{p^2}$. Let $P = \{\ell_1, \ell_2, \ldots, \ell_n\}$ be a set of primes. SuperSolver chooses $\mathcal{L}$ to minimise the following:

$$\frac{C + \sum_{\ell \in S} C_\ell}{2 + \sum_{\ell \in S}(\ell + 1)}.$$

This represents the ratio of the amount of work done at each node by the total number of nodes inspected at that node (there are 2 nodes revealed when computing the neighbours in the 2-isogeny graph, and for each ℓ there are $\ell + 1$ neighbours in the ℓ-isogeny graph).

Cost metric. Since the goal of our work is to optimize the practical cost of solving the isogeny problem, we will follow [CRSCS22] regarding the cost metric: instead of considering asymptotic complexities in the O-notation, we will measure the costs in terms of $\mathbb{F}_p$-multiplications. As is standard, (e.g., see [CEMR24]), we will count $\mathbb{F}_p$-squarings as 0.8 multiplications, and disregard $\mathbb{F}_p$-additions due to their relatively small cost compared to multiplications. Naturally, our algorithms also require $\mathbb{F}_{p^2}$-operations. We follow the standard of implementing those by calling $\mathbb{F}_p$-operations (see, e.g., [CRSCS22, CEMR24]). Therefore, we count 3 $\mathbb{F}_p$-multiplications per $\mathbb{F}_{p^2}$-multiplication and 2 $\mathbb{F}_p$-multiplications per $\mathbb{F}_{p^2}$-squaring, while ignoring $\mathbb{F}_{p^2}$-additions since they only require $\mathbb{F}_p$-additions.

With this metric, we avoid dependence on the actual implementation of the underlying field arithmetic, whose efficiency may vary depending on the shape and size of p. However, it is easy to convert our count of field operations to other metrics like bit operations, cycle counts, or similar, or to a different ratio between the cost of $\mathbb{F}_p$-operations.

3 WayFinder: A generalised Delfs–Galbraith algorithm

The Delfs–Galbraith algorithm finds an isogeny between two given supersingular elliptic curves E_1 and E_2 by first finding isogenies $\varphi_i\colon E_i \to E_i'$ where E_i' are defined over $\mathbb{F}_p$. After taking an ascending step in the 2-volcano if necessary, we can assume both E_1' and E_2' are oriented by the ring of integers R of $\mathbb{Q}(\sqrt{-p})$. One then uses the action of the class group $\mathrm{Cl}(R)$ of R to compute an $\mathbb{F}_p$-isogeny $\psi\colon E_1' \to E_2'$, and $\widehat{\varphi_2} \circ \psi \circ \varphi_1\colon E_1 \to E_2$ solves the isogeny problem on input E_1 and E_2.

With this perspective, for any fixed set $\mathcal{R}$ of imaginary quadratic orders such that p is inert or ramified in each $R \in \mathcal{R}$, there is a generalisation of the Delfs–Galbraith algorithm (which we call WayFinder or $\mathcal{R}$-WayFinder) for solving the isogeny problem that works as follows. Given E_1 and E_2 supersingular, the *first stage* of the algorithm is to find paths from E_1 to E_1' and E_2 to E_2' in the 2-isogeny graph via depth-first search, where E_1' and E_2' are orientable by some $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$, respectively. In the *second stage*, we use the orientations of E_1' and E_2' by R_1 and R_2 to find an isogeny $\psi\colon E_1' \to E_2'$. Observe that if $\mathcal{R}$ is the set of imaginary quadratic orders containing $\mathbb{Z}[\sqrt{-p}]$, then the $\mathcal{R}$-WayFinder algorithm is just the Delfs–Galbraith algorithm, and that SuperSolver can be viewed as $\mathcal{R}$-WayFinder where $\mathcal{R}$ is the set of orders containing $\mathbb{Z}[\ell\sqrt{-p}]$ for some finite set of odd positive integers ℓ.

We focus in this section on analyzing the cost of the bottleneck, which is the first step as in Delfs–Galbraith. In particular, we will assume in this section that we have

the following subroutines, all of which are required to implement $\mathcal{R}$-WayFinder: (1) an efficient test for orientability by $R \in \mathcal{R}$; (2) an efficient method for finding an orientation by R on an R-orientable curve E; and (3) a method for solving the problem of computing an isogeny between an R-curve E and an R'-curve E'. In Sections 4, 5, and 6, we will consider details of particular implementations of these subroutines, including one which applies to more general orders than Delfs–Galbraith or SuperSolver.

3.1 The cost model for $\mathcal{R}$-WayFinder

Let $V(\mathcal{R})$ denote the set of isomorphism classes of supersingular elliptic curves that have an orientation by some order in $\mathcal{R}$ and let V denote the set of all supersingular isomorphism classes. For d sufficiently large, a random walk of length d in the 2-isogeny graph $G(p, 2)$ beginning at E will terminate in $V(\mathcal{R})$ with probability approximately $\#V(\mathcal{R})/\#V$. Since $G(p, 2)$ is a Ramanujan graph, taking the length d to be approximately $2 \log p$ suffices; see [BCC+23, Theorem 11]. This gives an algorithm for finding an isogeny from E to a curve in $V(\mathcal{R})$ with complexity $\widetilde{O}(\#V/\#V(\mathcal{R}))$: repeatedly take a random walk of length d from E until terminating in $V(\mathcal{R})$. In practice, it can be advantageous to use depth-first search rather than random walks to avoid recomputing the same initial segment of a branch and retesting for orientability of curves near E in $G(p, 2)$. We therefore analyse depth-first search with depth $\log p \leq d \leq 2 \log p$ for a curve in $V(\mathcal{R})$.

While the algorithm is now deterministic, we still refer to the expectation of various parameters of the algorithm, viewing the expectation as being taken over random inputs. The expected cost of the resulting algorithm is the work done per node multiplied by the number of nodes visited. The search tree is a rooted 3-regular tree of depth d. We model a branch of the search tree as a random non-backtracking walk of length d. While this model has flaws (e.g., the leaves are highly correlated since "nearby" leaves have short paths between them), and relies on several heuristics, our experiments confirm that our analysis is still sufficiently accurate.

SuperSolver aimed to minimise the ratio of the work done per node by the number of "nodes revealed". When running SuperSolver with a set $\mathcal{L}$, the number of "nodes revealed" is $2 + \sum(\ell + 1)$, where the sum runs over $\ell \in \mathcal{L}$. The main idea is that the two children in the search tree in the 2-isogeny graph are computed, and running NeighbourInFp for $\ell \in \mathcal{L}$ checks all $\ell + 1$ neighbours of the given node E in the ℓ-isogeny graph; these neighbours are the "nodes revealed". However, this model is overly optimistic. Indeed: This model ignores the fact that the $\mathbb{F}_p$-rational vertices can induce a highly structured subgraph in $G(p, \ell)$. For example, when ℓ splits in $\mathbb{Q}(\sqrt{-p})$, this subgraph consists of connected components that are (generically) cycles, where the length is often exponential in $\log p$ as it is the order of the prime above ℓ in $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$. This structure means that an $\mathbb{F}_p$-rational j-invariant may only be adjacent to $\ell - 1$, rather than $\ell + 1$, irrational j-invariants.

We replace the "nodes revealed" metric with an estimate for the total complexity of SuperSolver (and more generally for the $\mathcal{R}$-WayFinder algorithm). We then seek to minimise the ratio of that cost with the cost of Delfs–Galbraith. This reframing not only results in more accurate predictions for the total cost of the Delfs–Galbraith and SuperSolver algorithms, but also allows us to develop and analyse WayFinder variants that test for orientability by imaginary quadratic fields beyond $\mathbb{Q}(\sqrt{-p})$. We carry this out in Section 6.

Suppose now $\mathcal{R}$ is a finite set of imaginary quadratic orders such that, for a given curve E, we can test whether E is orientable by at least one $R \in \mathcal{R}$ at least as fast as we can compute the children of E in the search tree. More precisely, testing for orientability by at least one $R \in \mathcal{R}$ is no slower than computing a square root of a square in $\mathbb{F}_{p^2}$ (see the discussion in §3 of [CRSCS22]). We also assume that we can compute an orientation of any R-orientable curve. An example of a family of such orders are those containing $\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ where ℓ_1, ℓ_2 are either 1 or small primes. These examples will come up again in Section 6

in our concrete instantiation of $\mathcal{R}$-WayFinder, where we propose an orientability-checking algorithm for orders of the form $\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$.

Let $C_{\mathcal{L}}$ denote the cost per node, and $N_{\mathcal{L}}$ the number of nodes visited. The total cost of the algorithm is $M_{\mathcal{L}} := C_{\mathcal{L}} \cdot N_{\mathcal{L}}$. The cost per node depends on the algorithms used to detect orientability. In the following subsection, we estimate the total number of nodes visited for an arbitrary order R.

3.2 Random walks, depth-first search, and clustering

Let $V(R)$ be the set of j-invariants orientable by R, a subset of V. For simplicity of our model, we assume the conductor of R has 2-adic valuation at most 1. An initial estimate above for the number of nodes visited in a depth-first search for a node in $V(R)$ is $\#V/\#V(R)$. However, this estimate is rather crude. As observed in [CRSCS22, Remark 5], because $V(R)$ induces a highly structured subgraph in V, this structure can cause the average number of nodes visited to differ from $\#V/\#V(R)$ by a constant factor.

In this section, we refine this initial estimate by accounting for the structure of oriented subgraphs.

3.2.1 Hitting time and depth-first search.

Fix $d \geq 0$ so that a random non-backtracking 2-isogeny walk of length d is approximately stationary. We wish to estimate N, the average number of nodes visited during depth-first search of the 2-isogeny graph to depth d to find exactly one node in S, a subset of V. This is related to the cumulative distribution function of the hitting time for S (from vertex v_0): Let $(v_t)_{t\geq 0}$ be a non-backtracking random walk beginning at v_0, and define the hitting time to be the random variable $\tau_S = \min_t\{v_t \in S\}$. Then $\Pr[\tau_S \leq d]$ is the probability that a randomly selected branch has a node in S. Let $Q_k = \Pr[\tau_S = k]$, the probability that a random non-backtracking walk first enters S after exactly k steps. We will treat $G(p, 2)$ as if it were a simple graph in the following analysis; the number of loops and multiple edges are both bounded by small constants, independent of p. Additionally, we make the following assumption.

Assumption 1. $\mathbb{E}[\textit{nodes visited in DFS for } S \textit{ to depth } d] \approx \Pr[\tau_S = d]^{-1} = Q_d^{-1}$.

In a moment, we will justify Assumption 1 with another assumption we will use again:

Assumption 2. *$Q_{d-r} \approx Q_d$ for $r \leq R$, and Q_{d-r} is negligible otherwise.*

A more rigorous analysis could instead estimate $\Pr[\tau_S \leq d] = \sum_{k\leq d} Q_k$ where $Q_k = \Pr[\tau_S = k]$ and incorporate a geometric decay $Q_{d-r} \approx 2^{-r/2}Q_d$ for r sufficiently large, where the factor of $2^{-1/2}$ comes from the mixing time of the non-backtracking random walk. For our purposes, we find the above heuristic holds reasonably well in experiments with small primes: as the prime p varies, the average length of a path to the $\mathbb{F}_p$-rational vertices found via depth-first search with depth $d = \lfloor \log_2 p \rfloor + 1$, for example, is often between $d - 1$ and d.

Let us return to justifying Assumption 1. We suppose we explore N nodes in the DFS tree so that on average, of the N nodes, only one is in S. Let N_k denote the number of depth k nodes visited, so $Q_k N_k$ is the probability that the node in S in the explored search tree occurs at depth k. Then using Assumption 2, we have

$$1 \approx \sum_{k\geq 0} Q_k N_k \approx Q_d \sum_{k\geq 0} N_k \approx Q_d N.$$

Thus $N \approx Q_d^{-1}$, justifying Assumption 1 under Assumption 2.

We now must calculate Q_d^{-1}. We will show that

$$Q_d^{-1} = c_S \frac{\#V}{\#S}$$

for a constant c_S that encodes a correlation or clustering penalty. We calculate c_S by counting branches with paths contained in S.

Definition 1. Let $W = v_0 \cdots v_d$ be a branch of the search tree and let $0 \leq r \leq d$. We say *W ends with a length r run in S* if $v_i \notin S$ for $0 \leq i < d - r$ and $v_i \in S$ for $d - r \leq i \leq d$.

Let ∂S denote the **boundary** of S, the set of vertices of S with at least one neighbor not in S.

Assumption 3. *For all but a negligible number of $v \in \partial S$ and for all $r \geq 0$, there are exactly Z_r non-backtracking paths contained in S, where Z_r depends on r but not on v.*

In other words, we are assuming that for fixed d and for $0 \leq r \leq d$, the number of branches that end in length r runs in S with fixed initial segment $v_0 \cdots v_{d-r}$ depends only on r, except for negligible collection of bad branches. We now estimate $c_S = (\Pr[\tau_S = d])^{-1}/(\#V/\#S)$, the correlation factor.

Proposition 2. *If $d \approx \log p$ and given Assumptions 1 to 3, $c_S \approx \sum_{r \geq 0} Z_r 2^{-r}$.*

Proof. Every leaf node corresponds to a branch in the search tree, a tree of depth d. If the leaf node is in S, there is some depth $d - r$ such that the branch first entered S. Let E_r be the event $\{\tau_S = d - r \text{ and } v_{d-r}, v_{d-r+1}, \ldots, v_d \in S\}$, that is, the walk first enters at depth $d - r$ and then ends with a length r run in S. Ignoring the branches that enter, leave, and then return to S, we denote the proportion of leaf nodes in S by P_d and approximate it by

$$P_d \approx \sum_{r=0}^{d} \Pr[E_r].$$

We now write $\Pr[E_r] = \Pr[v_i \in S, d - r \leq i \leq d \mid \tau_S = d - r] \Pr[\tau_S = d - r]$. Since there are 2^r branches beginning at a vertex v_i at depth $d - r$, Assumption 3 implies that $\Pr[v_i \in S, d - r \leq i \leq d \mid \tau_S = d - r] = Z_r 2^{-r}$. Thus $\Pr[E_r] = Q_{d-r} Z_r 2^{-r}$. Since d is near the mixing length, the proportion of non-backtracking walks of length d terminating in S is $\approx \#S/\#V$. We calculate using Assumption 2

$$\frac{\#S}{\#V} \approx P_d = \sum_{r=0}^{d} Q_{d-r} Z_r 2^{-r} \approx Q_d \sum_{r=0}^{R} Z_r 2^{-r},$$

where $Q_{d-r} \approx Q_d$ for $r \leq R$ and Q_{d-r} is negligible for $r > R$ as in Assumption 2. The result follows from Assumption 1 and approximating $\sum_{r \geq 0} 2^{-r} Z_r$ by $\sum_{r=0}^{R} 2^{-r} Z_r$:

$$\mathbb{E}[\text{nodes visited}] \approx (\Pr[\tau_S = d])^{-1} \approx Q_d^{-1} \approx \left(\sum_{r=0}^{R} Z_r 2^{-r} \right)^{-1} \frac{\#V}{\#S}.$$

□

The estimate for c_S depends on the structure of S. We consider three possibilities:

1. **ramified**: connected components are pairs of vertices connected by a single edge.
2. **inert**: components of S are 3-volcanoes of depth 1 with a rim of length 0.

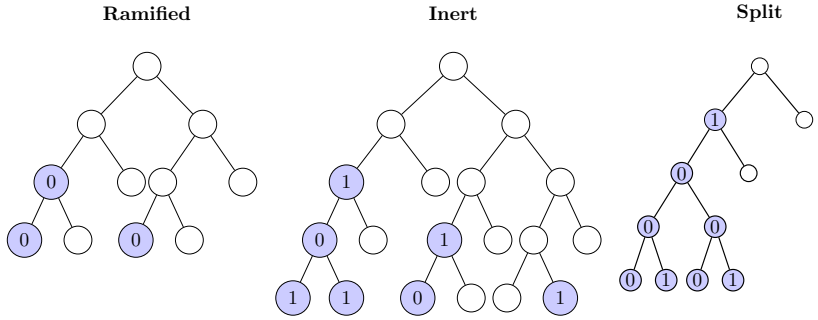


Figure 1: How a branch in the DFS tree can enter S in each case: ramified, inert, and split. Labels correspond to the depth of the vertex in the volcano.

3. **split**: components of S are volcanoes of depth 1 with rim of length h.

We can now calculate Z_r in each case above.

Corollary 1. *If S is ramified, inert, or split then it satisfies Assumption 3. Assume Assumptions 1 and 2. Define $c_S = \sum_{r=0}^{R} Z_r 2^{-r}$. Then if S is of ramified, inert type we have $c_S = 3/2$ or 2 respectively. If S is if split type then $c_S \approx 3$. Consequently*

$$\mathbb{E}[\textit{nodes visited}] \approx \begin{cases} \frac{3}{2}\frac{\#V}{\#S} & \textit{if } S \textit{ is ramified} \\ 2\frac{\#V}{\#S} & \textit{if } S \textit{ is inert} \\ 3\frac{\#V}{\#S} & \textit{if } S \textit{ is split} \end{cases}.$$

Proof. In each case, we count the number of paths of length r in S that begin at a vertex at the boundary of S. We then apply Proposition 2 for the second claim. In the **ramified** case, we have $Z_0 = 1$, $Z_1 = 1$, and $Z_r = 0$ for $r > 1$, because the longest run of edges in S is 1. So $c_S = \left(1 \cdot 2^0 + 1 \cdot 2^{-1}\right) = \frac{3}{2}$.

In the **inert** case, we have $Z_0 = 1$, $Z_1 = 1$ because each floor vertex is adjacent to exactly one rim vertex, $Z_2 = 2$ because now v_{d-2} is a floor vertex, the rim vertex is at depth $d-1$, and there are two child nodes of the rim vertex in S. For $r > 2$ there can not be a nonbacktracking walk of length r contained in S so $Z_r = 0$ for $r > 2$. Thus $c_S = \left(1 \cdot 2^0 + 1 \cdot 2^{-1} + 2 \cdot 2^{-2}\right) = 2$.

Now consider the **split** case. We can only enter the graph at a depth 1 vertex in the volcano, and then there are two branches along the rim to depth d contained in the search tree. From each depth $d-1$ vertex in the rim there will be two depth d vertices in the rim and two depth d vertices at depth 1 in the volcano. We have $Z_0 = Z_1 = 1$, $Z_2 = 2$, and $Z_r = 4$ for $r \geq 3$. Thus $c_S = \left(1 + \frac{1}{2} + \frac{2}{4} + \sum_{r=3}^{d} 2^{-r+2}\right) \approx 3$. The final claim follows from $\mathbb{E}[\text{nodes visited}] \approx (\Pr[\tau_S \leq d])^{-1} \approx c_S \frac{\#V}{\#S}$. □

Remark 1. This analysis matches the experimental data in [CRSCS22]. Let p be an odd prime larger than 3 and let $K = \mathbb{Q}(\sqrt{-p})$. When $p \equiv 1 \pmod 4$, the $\mathbb{F}_p$-subgraph is *of ramified type* since 2 ramifies in K (each supersingular $E/\mathbb{F}_p$ is 2-isogenous to exactly one supersingular $\mathbb{F}_p$-rational curve). When $p \equiv 3 \pmod 8$ the $\mathbb{F}_p$-subgraph is *of inert type* since 2 is inert in K; if E has CM by R_K then it is 2-isogenous to exactly three curves over $\mathbb{F}_p$ each with CM by $\mathbb{Z} + 2R_K$. When $p \equiv 7 \mod 8$ the $\mathbb{F}_p$-rational subgraph is *of split type* since 2 splits in R_K: if E has CM by R_K then it is 2-isogenous to exactly two curves with CM by R_K and one curve with CM by $\mathbb{Z} + 2R_K$. A connected component is a 3-volcano of depth 1: the rim corresponds to the CM action by the class of $\mathfrak{p}$ above

2 and its length is the order of $\mathfrak{p}$ in $\mathrm{Cl}(R_K)$. From every rim vertex there is exactly one descending isogeny to a floor vertex.

There are some pathologies when passing from the $\mathbb{F}_p$-rational 2-isogeny graph to the subgraph induced in $G(p,2)$ by taking $\overline{\mathbb{F}_p}$-isomorphism classes; see [ACNL+23] and [HAS26]. The analysis we provide primarily concerns only the local geometry of the induced subgraph of the oriented isogeny graph, and for large p the local neighbourhood of a typical vertex in $V(\mathbb{F}_p)$ or $V(R)$ will be indistinguishable from its neighbourhood in the oriented isogeny graph.

For an imaginary quadratic order R, define

$$c_2(R) = c_{V(R)}^{-1} = \begin{cases} \frac{2}{3} & : \left(\frac{\mathrm{disc}\, R}{2}\right) = 0 \\ \frac{1}{2} & : \left(\frac{\mathrm{disc}\, R}{2}\right) = -1 \\ \frac{1}{3} & : \left(\frac{\mathrm{disc}\, R}{2}\right) = 1 \end{cases}.$$

Define $V(R)$ to be the set of j-invariants that are primitively orientable by R. We have modeled the probability that a branch first enters set of orders orientable by R at depth d by $c_2(R)\frac{\#V(R)}{\#V}$.

We now assume an additional heuristic:

Assumption 4. *$\#(V(R) \cap V(R'))/\#V(R)$ is negligible for $R \neq R' \in \mathcal{R}$.*

With this assumption and under our model of branches in the DFS tree as random walks of length d, we estimate the probability that a branch first enters $V(\mathcal{R})$ at depth d is

$$\sum_{R \in \mathcal{R}} c_2(R) \frac{\#V(R)}{\#V}$$

so the expectation of the number of nodes visited is

$$\frac{\#V}{\sum_{R \in \mathcal{R}} c_2(R) \#V(R)}.$$

We demonstrate the clustering impact on the complexity of depth-first search for subfield curves, and verify the clustering constants above experimentally in Appendix A.

3.3 A new cost model with clustering

We now revisit Delfs–Galbraith, SuperSolver, and their cost models. Suppose we wish to test for orientability by a specific set of orders $\mathcal{R}$. First, we specify how the depth-first search tree is explored. We will represent nodes in the isogeny graph, and hence in the search tree, as j-invariants (but we may refer to a node j as a curve E with $j(E) = j$ when convenient). Let $j(E)$ be the root of the tree. If $j(E) \in \mathbb{F}_p$, we can terminate, so we assume $j(E) \notin \mathbb{F}_p$. Suppose we are at a node j in the DFS tree whose parent node is j_{prev} at depth r. We test if $j \in \mathbb{F}_p$ and if $E(j)$ is R-orientable for each $R \in \mathcal{R}$. Finally, if $r < d$ we compute the two children of j in the search tree. These are the 2 (generically distinct) roots of $\Phi_2(j, Y)/(Y - j_{\mathrm{prev}})$. We append both to a stack (with pointers to their parent j and depth $r+1$) and continue, popping a node from the stack, testing for $\mathbb{F}_p$-rationality and membership in $V(\mathcal{R})$, and proceed until we have found an orientable curve.

Let C denote the cost of computing the two children of a node in the search tree and $C(R)$ denote the cost of testing for orientability of a node by the order R. Assume that $\mathcal{R}$ does not contain any of the orders containing $\mathbb{Z}[2\sqrt{-p}]$. Indeed, we already test for orientability by orders containing $\mathbb{Z}[\sqrt{-p}]$, and testing for orientability by $\mathbb{Z}[2\sqrt{-p}]$ is equivalent to testing whether E is 2-isogenous to an $\mathbb{F}_p$-curve. We now estimate the total cost of $\mathcal{R}$-WayFinder. The total number of nodes visited T will be approximately $2L$ where

L is the number of leaf nodes visited. We run the tests for orientability by some $R \in \mathcal{R}$ at all T vertices, and we compute children at L vertices. We estimate

$$T \approx \frac{\#V}{c_2(\mathbb{Z}[\sqrt{-p}])\#V(\mathbb{F}_p) + \sum_{R\in\mathcal{R}} c_2(R)\#V(R)}.$$

The cost per node is

$$C\frac{T}{2} + \left(\sum_{R\in\mathcal{R}} C(R)\right) T = \left(\frac{C}{2} + \sum_{R\in\mathcal{R}} C(R)\right) T$$

so the total cost is estimated to be

$$\left(\frac{C}{2} + \sum_{R\in\mathcal{R}} C(R)\right) \cdot \frac{\#V}{c_2(\mathbb{Z}[\sqrt{-p}])\#V(\mathbb{F}_p) + \sum_{R\in\mathcal{R}} c_2(R)\#V(R)}.$$

3.3.1 Estimated cost of Delfs–Galbraith

We now estimate the cost of the Delfs–Galbraith algorithm in this model. We only test for $\mathbb{F}_p$-rationality, so $\mathcal{R}$ is empty. Testing a node for $\mathbb{F}_p$-rationality is essentially free. The cost per node is just the cost of computing its children if it is a non-leaf node. Recall that L is the expected number of leaf nodes visited before finding an $\mathbb{F}_p$-rational vertex, and T is the total number of nodes visited, and $2L \approx T$. Since we assume $2L \approx T$, there are L leaf nodes and L non-leaf nodes, so the total cost is approximately $CT/2$. We have $T \approx \#V/\#V(\mathbb{F}_p)$ and $\#V(\mathbb{F}_p) = \frac{1}{2}H(-4p)$ where $H(R) = \sum_{R'\supseteq R} h(R)$ is the Hurwitz class number of R. The total cost is estimated to be

$$C\frac{\#V}{c_2(\mathbb{Z}[\sqrt{-p}])H(-4p)}.$$

3.3.2 Estimated cost of SuperSolver

Let us now specialize to SuperSolver. At each node, we run the NeighbourInFp algorithm for a finite set of odd primes $\mathcal{L}$, which tests whether the node is orientable by any $\mathbb{Z}[\ell\sqrt{-p}]$ for $\ell \in \mathcal{L}$. Since we only run this test if the node is not $\mathbb{F}_p$-rational, we are testing whether the node has an ℓ-primitive orientation by $\mathbb{Z}[\ell\sqrt{-p}]$. The number of ℓ-primitive orientations by $\mathbb{Z}[\ell\sqrt{-p}]$ is $H(-4\ell^2 p) - H(-4p) = \left(\ell - \left(\frac{-4p}{\ell}\right)\right) H(-4p)$. We approximate the number of isomorphism classes of curves that can be ℓ-primitively oriented by $\mathbb{Z}[\ell\sqrt{-p}]$ by $\frac{1}{2}\#\operatorname{Cl}(R)$; this will be justified for our concrete instantiation. The number of curves we detect with this test is thus approximately $\frac{1}{2}\left(\ell - \left(\frac{-4p}{\ell}\right)\right) H(-4p)$.

For $\ell = 1$ or $\ell \in \mathcal{L}$, the constant $c_2(\mathbb{Z}[\ell\sqrt{-p}])$ is $2/3, 1/2$, or $1/3$ according to whether or not 2 is ramified, inert, or split in $\mathbb{Q}(\sqrt{-p})$. In particular, it does not depend on ℓ, and both it and $H(-4p)$ disappear in the ratio so neither appears in our objective function. More precisely, we wish to minimise

$$\frac{\left(C/2 + \sum_{\ell\in\mathcal{L}} C_\ell\right) c_2(\mathbb{Z}[\sqrt{-p}])H(-4p)/2}{(C/2)\left(c_2(\mathbb{Z}[\sqrt{-p}])H(-4p)/2 + \sum_{\ell\in\mathcal{L}} c_2(\mathbb{Z}[\ell\sqrt{-p}])\left(\ell - \left(\frac{-4p}{\ell}\right)\right) H(-4p)/2\right)},$$

which simplifies to

$$\frac{C + 2\sum_{\ell\in\mathcal{L}} C_\ell}{C\left(1 + \sum_{\ell\in\mathcal{L}}\left(\ell - \left(\frac{-4p}{\ell}\right)\right)\right)}. \tag{1}$$

After multiplying by C, which depends only on p, we get an objective function very similar to

$$\frac{C + \sum_{\ell \in \mathcal{L}} C_\ell}{1 + \sum_{\ell \in \mathcal{L}} (\ell + 1)},$$

which is the objective function from [CRSCS22].

We observe that our objective function penalizes including a given prime ℓ in two ways: it contributes a cost $2C_\ell$ rather than C_ℓ in the numerator, and if ℓ is not inert in $\mathbb{Q}(\sqrt{-p})$, its contribution to the denominator shrinks (relative to the SuperSolver objective function) from $\ell + 1$ to ℓ or $\ell - 1$.

3.3.3 Advantages of the new cost model

By viewing the first stages of the Delfs–Galbraith SuperSolver algorithms as a search for curves with certain orientations, we solve multiple previously discussed issues with the "nodes visited" metric simultaneously. We now view the work done at a node as a test for orientability of $E(j)$ by the order $\mathbb{Z}[\ell\sqrt{-p}]$. This properly casts it as an event entirely about the node j. The second issue was that it was overly optimistic. This is fixed by seeing the "benefit" of using the factor of $\ell - \left(\frac{-4p}{\ell}\right)$ in the denominator, rather than $\ell + 1$. Finally, our cost model lets us analyse generalisations of the algorithm, which we take up in Section 6. Once we have an optimal set $\mathcal{R}$ of orders, we get a more accurate cost model, since we can incorporate the impact of clustering. This allows us to accurately predict the (average) cost of any instantiation of WayFinder on input some prime p. This is invaluable for cryptanalysis: we can compute the concrete security of isogeny-based cryptographic schemes relying on the hardness of the isogeny problem (see Section 7).

4 Solving the isogeny problem between oriented curves

The second stage in the Delfs–Galbraith algorithm consists of finding an isogeny between two supersingular elliptic curves over $\mathbb{F}_p$ with $\mathbb{F}_p$-rational endomorphism rings isomorphic to the same quadratic order R, see [DG16, Section 3]. The authors suggest solving this problem with a meet-in-the-middle algorithm in the graph of elliptic curves with the same property, where edges are given by a set of ideals generating the class group $\mathrm{Cl}(R)$. Heuristically, by the birthday paradox, this approach takes $\tilde{O}(\sqrt{\#\,\mathrm{Cl}(R)})$ steps in the graph before finding a collision. This result was proven rigorously, conditionally on the Generalized Riemann Hypothesis, in [HW25] using the more general framework of R-orientations. Nevertheless, this method has two main limitations.

- First, the meet-in-the-middle algorithm requires exponential space, as it needs to store an exponential number of curves to exploit the birthday paradox. As mentioned in both previously cited papers, see [DG16, Section 3, Remark 4] and [HW25, Remark 6], this limitation can be addressed by using the well-known Pollard-ρ approach.
- Second, the two oriented curves need to lie on the same graph. This *de facto* excludes the case of elliptic curves oriented by different number fields.

In this section, we outline an algorithm that addresses both issues by following the approaches of Galbraith–Hess–Smart and Galbraith–Stolbunov [GHS02, GS13] to obtain a low-storage variant, and by using [Wes22a, Lemma 3] to connect sets of elliptic curves oriented by different orders via the Deuring correspondence. The algorithm we present for stage 2 of WayFinder relies on the following subprocedures that can be instantiated in several ways.

- OrientedSpecialCurve: Given a prime p and a quadratic order R such that there exist R-oriented supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, return a primitively

R-oriented elliptic curve (E, ι) together with an isomorphism $\varepsilon : \mathrm{End}(E) \xrightarrow{\sim} \mathcal{O}$, where $\mathcal{O}$ is a maximal order in $B_{p,\infty}$ such that an embedding $R \hookrightarrow \mathcal{O}$ is known. Under GRH, and given the factorisation of the discriminant of R, this can be done in polynomial time in the length of the input via [Wes22a, Lemma 3].

- SmoothAction: Given $B > 0$, a primitively R-oriented elliptic curve (E, ι) defined over $\mathbb{F}_{p^2}$, and an invertible R-ideal $\mathfrak{a}$ of B-smooth norm, return the action $\mathfrak{a} \star (E, \iota)$. Note that we do not need to return the corresponding oriented isogeny $\varphi_{\mathfrak{a}}$. Under GRH, this computation can be done in polynomial time in the length of the input and in B using [HW25, Corollary 3]. However, that result makes use of higher-dimensional isogenies, which makes the algorithm completely impractical without further hypotheses on the characteristic of the ground field. This is a serious limitation for obtaining a generic implementation; see Remark 3.

- SpecialAction: Given a primitively R-oriented elliptic curve (E, ι) defined over $\mathbb{F}_{p^2}$, together with an explicit isomorphism $\varepsilon : \mathrm{End}(E) \xrightarrow{\sim} \mathcal{O}$ where $\mathcal{O}$ is a maximal order in $B_{p,\infty}$ such that an embedding $R \hookrightarrow \mathcal{O}$ is known, and an invertible R-ideal $\mathfrak{a}$, return the isogeny $\varphi_{\mathfrak{a}} : (E, \iota) \to \mathfrak{a} \star (E, \iota)$. Note that the isogeny itself is required, not only its codomain. This computation can be done by translating the ideal $\mathfrak{a}$ to the corresponding quaternion ideal and then applying KLPT-like algorithms [KLPT14, Wes22b], the latter being proven under GRH to run in polynomial time.

- OrientedIsomorphism: Given two primitively R-oriented elliptic curves (E_1, ι_1) and (E_2, ι_2), return whether the two oriented elliptic curves are isomorphic. One can for instance use Wesolowski's unique encoding to check for isomorphism between oriented elliptic curves efficiently, see [Wes22a, Section 7].

The references provided above to instantiate these procedures are very generic and can be optimized depending on the given context. In Section 6.2, we described the concrete algorithms that can be used in our specific case, namely for orientations by orders of the form $\mathbb{Z}[\sqrt{-dp}]$ for d a square-free positive integer. First, we generalize a result of [HW25] to curves oriented by different orders.

Proposition 3. *Assume the Generalised Riemann Hypothesis (GRH). Let R and R' be two imaginary quadratic orders in which a prime p is not split. Let $\Delta := \max(|\operatorname{disc}(R)|, |\operatorname{disc}(R')|)$. Given two supersingular elliptic curves E and E', defined over $\mathbb{F}_{p^2}$, primitively oriented by orders R and R' respectively, one can compute an isogeny from E to E' in time $O(l^{O(1)} \sqrt[4]{\Delta})$, where l denotes the length of the input $(E, \iota\colon R \hookrightarrow \mathrm{End}(E))$ and $(E', \iota\colon R' \hookrightarrow \mathrm{End}(E')$.*

Proof. We start by connecting the set of R-oriented elliptic curves to the set of R'-oriented elliptic curves. We first compute two elliptic curves, denoted E_0 and E_0', respectively, that are primitively oriented by the orders R and R', along with their endomorphism rings. Under GRH, by [Wes22a, Lemma 4], this takes time polynomial in the length of the input l and in $\log \Delta$ after we factor $\operatorname{disc} R$ and $\operatorname{disc} R'$. Since we know the endomorphism rings of E_0 and E_0', it is possible to find an isogeny $\psi : E_0 \to E_0'$ in polynomial time (under GRH) using KLPT-like algorithms [Wes22b]. On the other hand, by [HW25, Theorem 7], an isogeny $\varphi : E \to E_0$ (resp. $\varphi' : E' \to E_0'$) can be computed, under GRH, in time $O(l^{O(1)} \sqrt[4]{\Delta})$ by leveraging the knowledge of primitive orientations. The cost of computing these isogenies dominates the cost of factoring the discriminants. Hence, we obtain an isogeny $\hat{\varphi}' \circ \psi \circ \varphi : E \to E'$ in the claimed time complexity. □

From the proof of the above proposition, we see it suffices to connect (E, ι) to (E_0, ι_0) by an R-oriented isogeny and (E', ι') to (E_0', ι_0') by an R'-oriented isogeny, where E_0 and E_0' have known endomorphism ring. As shown in the proof, the curves E_0, E_0', their endomorphism rings, and the orientations by R and R' can all be computed in polynomial

time after we factor $\operatorname{disc} R$ and $\operatorname{disc} R'$. Moreover, they can all be computed efficiently in practice. So we have reduced to the case that we have two orientations by the same order, and we know the endomorphism ring of one curve. We now assume we are in this setting.

Let $R = \mathbb{Z}[\theta]$ be an imaginary quadratic order with p not split and let $(E, \iota), (E', \iota')$ be two R-oriented curves. Assume $\operatorname{End}(E)$ is known. We wish to construct an (oriented) isogeny $E \to E'$. We adapt the low-storage algorithm of Galbraith–Hess–Smart and Galbraith–Stolbunov [GHS02, GS13]. Let X be the set of primitively R-oriented elliptic curves. The group $\operatorname{Cl}(R) \times C_2$ acts freely and transitively on X if p is inert in R, and the group $\operatorname{Cl}(R)$ acts freely and transitively on X when p ramifies (see, for example, [Onu21, Proposition 3.3, Theorem 3.4], [Wes22a, Theorem 1] [ACL+24, Theorem 4.4]).

We now restrict our attention to the latter case. An ideal class $[\mathfrak{a}] \in \operatorname{Cl}(R)$ acts via the class group action $[\mathfrak{a}](E, \iota) = (E/E[\mathfrak{a}], (\varphi_{\mathfrak{a}})_* \iota)$. The GHS algorithm works as follows. Fix a function $v \colon X \to L$ where L is a set of split primes of R that (under GRH) generate $\operatorname{Cl}(R)$. Then v determines a function $f \colon X \to X$ via $f(x) = [v(x)]x$. Let $x_0 = (E, \iota)$ and let $y_0 = (E', \iota')$. Let $\mathfrak{a} = \mathfrak{b} = R$. Given $(x_i, \mathfrak{a})$ define $x_{i+1} = f(x_i)$ and update $\mathfrak{a}$ via $\mathfrak{a} \leftarrow I$ where I is the reduced ideal equivalent to $v(x_i)\mathfrak{a}$. Let $T \approx \sqrt{h_R}$ and compute $(x_T, \mathfrak{a})$. Similarly define the sequence $y_1, y_2, \ldots$ and ideal $\mathfrak{b}$. After an expected $\approx T$ iterations we find $y_n = x_T$. Then we have ideals $\mathfrak{a}, \mathfrak{b}$ such that $\mathfrak{a} \cdot (E, \iota) = \mathfrak{b} \cdot (E', \iota')$. Then $\mathfrak{c} = \mathfrak{b}^{-1}\mathfrak{a}$ corresponds to an oriented isogeny $(E, \iota) \to (E', \iota')$. When p is inert in R, there are two orbits in X under the $\operatorname{Cl}(R)$ action, permuted by conjugation [Onu21]. By testing for a collision with the forward orbit of $\overline{(E, \iota)}$, this approach extends to the inert case.

It remains to compute the isogeny $E \to E'$. The Bisson–Sutherland algorithm [BS12] can compute a representation of the ideal $\mathfrak{c}$ as a low-weight product of prime ideals in time $O(\Delta^{1/4+\epsilon})$ and polynomial storage. It can be computed efficiently with Clapotis [PR23a] if, say, there is large rational 2-power torsion on E. However, we see a second benefit of using an oriented curve with known endomorphism ring: We can simply convert the quadratic ideal $\mathfrak{c}$ into the left ideal $I = \operatorname{End}(E)\mathfrak{c}$ and convert I into an isogeny $E \to E/E[I] \simeq E'$. These last steps run in polynomial time in theory and efficiently in practice.

In Algorithm 1, we give pseudocode for stage 2 of WayFinder in the special case that p ramifies in the two orders. We address how to extend it to the inert case in Remark 2.

Complexity analysis.

Let us now discuss the correctness and complexity of Algorithm 1. We assume the inputs N and M are chosen so that after N iterations it is possible to find a collision, and so that after M iterations it is likely that we would have already detected a collision, if one existed. In practice one could pick $N \approx 1.5\sqrt{\#\operatorname{Cl}(R)}$ and $M \approx 3\sqrt{\#\operatorname{Cl}(R)}$. We first focus on the meet-in-the-middle part of the algorithm, namely between Line 5 and Line 21, that is done twice to connect each E_i to a special cure $E_i^{(\mathrm{spe})}$. This is the adaptation of the GHS algorithm [GHS02] from the setting of computing $\mathbb{F}_p$-rational isogenies of elliptic curves to R-oriented elliptic curves that we discussed in Section 4.

Thus it takes $\tilde{O}(\sqrt{\Delta})$ group action computations to find the isogenies $\varphi_i : E_i^{(\mathrm{spe})} \to E_i$, where Δ denotes $\max(|\operatorname{disc}(R_1)|, |\operatorname{disc}(R_2)|)$. During this process, it is required to store 8 oriented elliptic curves and 6 ideal classes. An ideal class can be represented by a reduced representative of the class, which will have polynomial size. Hence, the memory cost is polynomial in $\log p$ and $\log \Delta$.

Then to conclude, it is only required to compute an isogeny between the two curves $E_1^{(\mathrm{spe})}$ and $E_2^{(\mathrm{spe})}$. Since the endomorphism rings of the two curves are known, this can be done in polynomial time using KLPT-like algorithms [KLPT14, Wes22b].

Finally, following the description and the suggested algorithmic instantiation of the algorithms in the comments in Algorithm 1, the overall time complexity is $\tilde{O}(\mathsf{len}(\texttt{input}) \cdot \sqrt{\Delta})$, and the memory complexity is polynomial in the length of the input.

Algorithm 1 WayFinder Stage 2

Input: Integers $N > 0$ and $M > 0$, two primitively oriented elliptic curves (E_1, ι_1) and (E_2, ι_2) defined over $\mathbb{F}_{p^2}$ respectively in $\mathrm{SS}(R_1)$ and $\mathrm{SS}(R_2)$, two functions $f_i : \mathrm{SS}(R_i) \to S_i$ where S_i is a set of B-smooth ideals generating $\mathrm{Cl}(R_i)$, for some $B > 0$

Output: An isogeny $\varphi : E_1 \to E_2$

1: **for** $i \in \{1, 2\}$ **do**
2: $(E_i^{(\mathrm{spe})}, \iota_i^{(\mathrm{spe})}), \mathrm{End}(E_i^{(\mathrm{spe})}) \overset{\varepsilon}{\simeq} \mathcal{O}_i^{(\mathrm{spe})} \leftarrow$ OrientedSpecialCurve(p, R_i)
3: **repeat**
4: $[\mathfrak{r}_i] \leftarrow$ a random smooth R_i-ideal class ▷ *Randomize starting point*
5: $(E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})}) \leftarrow [\mathfrak{r}_i] \star (E_i, \iota_i)$
6: $[\mathfrak{a}_i] \leftarrow [\mathfrak{r}_i]$
7: **for** $j \in \{1, \ldots, N\}$ **do**
8: $[\mathfrak{t}] \leftarrow f_i(E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})})$
9: $[\mathfrak{a}_i] \leftarrow [\mathfrak{a}_i] \cdot [\mathfrak{t}]$
10: $(E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})}) \leftarrow [\mathfrak{t}] \star (E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})})$ ▷ *Using* SmoothAction
11: $(E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})}) \leftarrow (E_i^{(\mathrm{spe})}, \iota_i^{(\mathrm{spe})})$
12: $[\mathfrak{b}_i] \leftarrow$ the trivial R_i-ideal class $[1]$
13: $k \leftarrow 0$
14: **while** $(E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})}) \neq (E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})})$ **and** $k < M$ **do** ▷ *Using* OrientedIsomorphism
15: $[\mathfrak{t}] \leftarrow f_i(E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})})$
16: $[\mathfrak{b}_i] \leftarrow [\mathfrak{b}_i] \cdot [\mathfrak{t}]$
17: $(E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})}) \leftarrow [\mathfrak{t}] \star (E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})})$ ▷ *Using* SmoothAction
18: $k \leftarrow k + 1$
19: **until** $(E_i^{(\mathrm{temp})}, \iota_i^{(\mathrm{temp})}) = (E_i^{(\mathrm{mid})}, \iota_i^{(\mathrm{mid})})$ ▷ *Using* OrientedIsomorphism
20: $[\mathfrak{c}_i] \leftarrow [\mathfrak{b}_i \bar{\mathfrak{a}}_i]$
21: $\varphi_i \leftarrow \varphi_{\mathfrak{c}_i} : E_i^{(\mathrm{spe})} \to E_i$ ▷ *Using* SpecialAction
22: $I \leftarrow$ a connecting ideal between $\mathcal{O}_1^{(\mathrm{spe})}$ and $\mathcal{O}_2^{(\mathrm{spe})}$
23: $\psi \leftarrow \varphi_I : E_1^{(\mathrm{spe})} \to E_2^{(\mathrm{spe})}$
24: **return** $\varphi_2 \circ \psi \circ \hat{\varphi}_1 : E_1 \to E_2$

Remark 2. If p is inert in R_i, then there are two orbits in the set of isomorphism classes of primitively R_i-oriented elliptic cures under the action of $\mathrm{Cl}(R_i)$. The action of complex conjugation permutes these two orbits, as mentioned earlier. Thus we can simply modify Step 4 in Algorithm 1 to instead act by a random element $([\mathfrak{r}], b)$ of $\mathrm{Cl}(R) \times \mathbb{Z}/2\mathbb{Z}$, where the bit b indicates whether we follow the action of $[\mathfrak{r}]$ by complex conjugation or not.

Remark 3. Following the SIDH attacks [CD23, MMP+23, Rob23], several efficient algorithms relying on higher-dimensional isogenies have been introduced in the field, notably [PR23b], to compute in polynomial time the action on oriented elliptic curves in full generality. Nevertheless, in practice, the formulas and implementations developed for higher-dimensional isogenies concern only chains of 2-isogenies and require a large accessible 2-torsion, see for instance [Dar25]. Thus, such algorithms are only usable over certain finite fields where supersingular elliptic curves have a large rational 2-torsion, for example prime characteristics of the form $p = c \cdot 2^f - 1$, where f and c are integers and c is small. While it is theoretically possible to efficiently rely on higher-dimensional isogenies over any finite field by exploiting its specific rational torsion, it would require developing dedicated formulas and implementations. By computing the action of $\mathfrak{c}_i$ on a special curve, see SpecialAction, we avoid relying on higher-dimensions, and thus ensure a practical deployment for any

characteristic p. To the best of our knowledge, there is currently no method to avoid higher-dimensional isogenies for SmoothAction in plain generality when an exponential amount of smooth ideal actions is computed, see the remark "Computing the action" in [Wes22a]. This is why we can only present an instantiation free from higher-dimensional isogenies in our specific case of orientation by $\mathbb{Z}[\sqrt{-dp}]$ in our concrete instantiation of stage two of $\sqrt{-\star p}$-WayFinder in Section 6.2. We avoid the issue via practical algorithms for computing orientations, oriented isogenies, and pushforward orientations.

5 Detecting Orientations

We exhibit a method of detecting orientations to instantiate $\mathcal{R}$-WayFinder, introduced in Section 3, for a certain set $\mathcal{R}$ of quadratic orders. We present two new algorithms:

- ConjugateFinder: for a fixed positive integer ℓ, detects if a given supersingular elliptic curve is ℓ-isogenous to its Frobenius conjugate. This detects $\mathbb{Z}[\sqrt{-\ell p}]$-orientations.
- OrientationFinder: for fixed positive integers ℓ_1, ℓ_2, detects whether a given supersingular elliptic curve is oriented by $\mathbb{Z}[\alpha]$ with $\alpha \in D \cup \ell_1 D$ where

$$D = \left\{ \sqrt{-p}, \frac{1+\sqrt{-p}}{2}, \sqrt{-\ell_2 p}, \frac{1+\sqrt{-\ell_2 p}}{2} \right\}.$$

5.1 Detecting conjugate neighbours

The following algorithm detects whether a given supersingular elliptic curve E is ℓ-isogenous to its conjugate $E^{(p)}$ by evaluating $\Phi_{\ell,p}(j, j^p)$ efficiently. Recall that for $\ell < p/4$ the curve E is orientable by $\mathbb{Z}[\sqrt{-\ell p}]$ if and only if $\Phi_{\ell,p}(j, j^p) = 0$, since an endomorphism of E of degree ℓp factors as Frobenius followed by an ℓ-isogeny $E^{(p)} \to E$.

Lemma 1. *Let $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$ such that $\beta^2 \in \mathbb{F}_p$. On input a j-invariant $j \in \mathbb{F}_{p^2}$ of a supersingular curve E, and a positive integer ℓ, ConjugateFinder from Algorithm 2 outputs* `true` *if and only if E is ℓ-isogenous to $E^{(p)}$. Furthermore, using our cost metric, it costs at most $3\ell^2 + \ell + \frac{2}{5}$ $\mathbb{F}_p$-multiplications.*

Proof. In Line 6, we compute the powers $j^2, \ldots, j^{\ell+1}$ with ℓ multiplications in $\mathbb{F}_{p^2}$. In Lines 4 and 8, we obtain $j^p, (j^p)^2, \ldots, (j^p)^{\ell+1}$ by conjugation, which incurs no $\mathbb{F}_p$-multiplications. Then, in Line 9, we compute $j \cdot j^p$ using two squarings and one multiplication in $\mathbb{F}_p$. We use this to compute $(j \cdot j^p)^k$ for $k = 2, \ldots, \ell$ in Line 11 with $\ell - 1$ multiplications in $\mathbb{F}_p$, noting that $j \cdot j^p \in \mathbb{F}_p$. Finally, from Lines 12 to 18, we compute $r = \Phi_{\ell,p}(j, j^p)$. Indeed, in Line 12, $r = j^{\ell+1} + (j^p)^{\ell+1}$, and then, for each $k = 1, \ldots, \ell - 1$ and $i = 1, \ldots, \ell - k$, we add $c_{k+i,k}\, j^{k+i}(j^p)^k + c_{k,k+i}\, j^k (j^p)^{k+i}$, with 6 multiplications in $\mathbb{F}_p$, since multiplying an $\mathbb{F}_{p^2}$-element by an $\mathbb{F}_p$-element costs 2 $\mathbb{F}_p$-multiplications each. In total, Lines 12 to 18 cost $3\ell(\ell-1)$ $\mathbb{F}_p$-multiplications. Summing up, and counting one $\mathbb{F}_{p^2}$-multiplication as 3 $\mathbb{F}_p$-multiplications, Algorithm 2 costs $3\ell^2 + \ell$ multiplications and 2 squarings in $\mathbb{F}_p$. Then, E with j-invariant j is ℓ-isogenous to $E^{(p)}$ if and only if $r = 0$. □

5.2 Generators for $X_0(\ell)$ and $X_0(\ell)^+$

Our method to detect orientations uses the modular curves $X_0(\ell)$ and $X_0^+(\ell)$ for values of ℓ such that these curves have genus 0. Therefore, we give a short primer on modular curves before introducing the main algorithm. For more background on modular curves see [DS05, DR73, KM85].

Modular curves. We will work with *modular curves*, which are complex curves whose points represent elliptic curves, enhanced by torsion data, up to the appropriate notion of

Algorithm 2 ConjugateFinder

Input: $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$ with $d = \beta^2 \in \mathbb{F}_p$, the j-invariant $j \in \mathbb{F}_{p^2}$ of a supersingular curve E, and a positive integer ℓ.
Output: A boolean `bool` indicating whether E is ℓ-isogenous to $E^{(p)}$.

Fetch the modular polynomial $\Phi_{\ell,p}(X,Y) = X^{\ell+1} + Y^{\ell+1} + \sum_{0\le i,j\le \ell} c_{i,j} X^i Y^j$
$x_1 \leftarrow j$
$a_1 + \beta b_1 \leftarrow x_1$ ▷ $a_1, b_1 \in \mathbb{F}_p$
$y_1 \leftarrow a_1 - \beta b_1$
for $k = 2, \dots, \ell + 1$ **do**
    $x_k \leftarrow x_{k-1} \cdot j$
    $a_k + \beta b_k \leftarrow x_k$ ▷ $a_k, b_k \in \mathbb{F}_p$
    $y_k \leftarrow a_k - \beta b_k$ ▷ $y_k = (j^p)^k$
$z_1 \leftarrow a_1^2 + d b_1^2$ ▷ $z_1 = jj^p$
for $k = 2, \dots, \ell$ **do**
    $z_k \leftarrow z_{k-1} \cdot z_1$ ▷ $z_k = (jj^p)^k$
$r \leftarrow x_{\ell+1} + y_{\ell+1}$
for $k = 1, \dots, \ell - 1$ **do**
    for $i = 1, \dots, \ell - k$ **do**
        $x_{i,k} \leftarrow z_k \cdot x_i$ ▷ $x_{i,k} = (jj^p)^k j^i$
        $a_{i,k} + \beta b_{i,k} \leftarrow x_{i,k}$
        $x_{i,k}^p \leftarrow a_{i,k} - \beta b_{i,k}$
        $r \leftarrow r + c_{i+k,k} x_{i,k} + c_{k,i+k} x_{i,k}^p$
return $(r \stackrel{?}{=} 0)$ ▷ $\Phi_{\ell,p}(j, j^p) \stackrel{?}{=} 0$

isomorphism. In particular, we will work with the curve $X_0(\ell)$, whose non-cuspidal points parameterise cyclic ℓ-isogenies of elliptic curves.

The modular curve $X_0(\ell)$ has an involution called the *Fricke involution*, ω_ℓ, defined on $\mathcal{H} = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$ by $\tau \mapsto -1/\ell\tau$. We denote by $X_0^+(\ell)$ the quotient of $X_0(\ell)$ by this involution. The non-cuspidal points of $X_0^+(\ell)$ represent equivalence classes of ℓ-isogenies ϕ between elliptic curves over $\mathbb{C}$, under the equivalence relation $\phi \sim \hat{\phi}$.

Both $X_0(\ell)$ and $X_0^+(\ell)$ have models defined over $\mathbb{Z}[\frac{1}{\ell}]$, and so can be reduced modulo p for any prime $p \neq \ell$; see [DS05, Theorem 8.6.1]. We denote by $\tilde{X}_0(\ell)_p$ and $\tilde{X}_0^+(\ell)_p$ the reduction modulo p of these curves. Suppose ℓ is a prime such that $X_0(\ell)$ has genus 0, so $X_0(\ell)^+$ has genus 0 too. In this case, the points on $X_0^+(\ell)$ can be identified by a single parameter. Supersingular points on $X_0(\ell)$ are defined over $\mathbb{F}_{p^2}$, so the same is true for its quotient $X_0(\ell)^+$. Thus, if the point P on $\tilde{X}_0^+(\ell)$ is supersingular and represented by the parameter value h, then $h \in \mathbb{F}_{p^2}$.

We use the modular curves $X_0^+(\ell)$ to find polynomials in $\mathbb{Z}[X,Y]$ that allow us to detect the presence of orientations by orders in $D \cup \ell_1 D$, analogously to how the modular polynomials $\Phi_\ell(X,Y)$ allowed us to detect the presence of orientations by $\mathbb{Z}[\sqrt{-\ell p}]$ in Algorithm 2. We construct these polynomials from algebraic relations between certain generators of the function fields of modular curves. It is therefore necessary for us to work with primes ℓ such that $X_0^+(\ell)$ and $X_0(\ell)$ have genus zero, since this guarantees that their function fields are generated over $\mathbb{C}$ by a single transcendental element.[1] In what follows, we assume we are always working with such a prime ℓ.

A generator for the function field $\mathbb{C}(X_0(\ell))$ of $X_0(\ell)$ parameterizes (up to isomorphism) ℓ-isogenies ϕ, while a generator for $\mathbb{C}(X_0^+(\ell))$ parameterizes (up to isomorphism) equivalence classes $[\phi]$ of ℓ-isogenies, where the equivalence relation is given by $\phi \sim \hat{\phi}$. As shown in

[1] $X_0(\ell)$ is genus 0 for $\ell \in \{1, 2, 3, 5, 7, 13\}$, and $X_0^+(\ell)$ is genus 0 for *supersingular primes*: $\ell \in \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59, 71\}$.

[Elk98], for $\ell \in \{2,3,5,7,13\}$, the function $t_\ell := (\eta(q)/\eta(q^\ell))^{24/(\ell-1)}$ is a generator for $\mathbb{C}(X_0(\ell))$, where $\eta(q)$ is the Dedekind eta function, and $h_\ell := t_\ell + \omega_\ell(t_\ell)$ is a generator for $\mathbb{C}(X_0^+(\ell))$. The definitions of these two generators for $\mathbb{C}(X_0(\ell))$ and $\mathbb{C}(X_0^+(\ell))$, respectively, give us crucial algebraic relations we use to construct our desired polynomials. The remaining necessary relations relate t_ℓ to the j-function and can be found in Section 4 of [Elk98].

Example 1. Let $\ell = 2$. Then, $t_2(q) = (\eta(q)/\eta(q^2))^{24}$ and we have the following Hauptmoduln for $X_0(2)$ and $X_0^+(2)$:

$$\begin{aligned} X_0(2) &\to X(1), & X_0(2) &\to X_0^+(2) \\ t_2 &\mapsto j := \frac{(t_2+256)^3}{t_2^2}, & t_2 &\mapsto h_2 := t_2 + \omega_2(t_2) = t_2 + \frac{2^{12}}{t_2} \end{aligned}$$

Moreover, $j(q^2) = (t_2(q)+16)^3/t_2(q)$ generates the function field of $X_0(2)$ over the function field of $X(1)$. The pair $(j(q), j(q^2))$ gives, for any choice of q, a pair of 2-isogenous j-invariants.

5.3 Detecting orientations

We now construct the analogues of $\Phi_\ell(X,Y)$ that allow us to detect orientations by orders in $D \cup \ell_1 D$ for a pair (ℓ_1, ℓ_2). These polynomials are constructed from iterated resultants involving the relations between generators for $X(1), X_0(\ell)$, and $X_0^+(\ell)$ explored in the previous subsection.

Our resultant polynomial. Let $\ell \in \{2,3,5,7,13\}$.[2] For ease of notation, we denote t_ℓ by T, h_ℓ by H, and the j-function by J. The definitions of T and H above, together with the relation between T and J in [Elk98] give us covering maps

$$\begin{aligned} X_0(\ell) &\to X(1) & X_0(\ell) &\to X_0(\ell)^+ \\ T &\mapsto J := \frac{f_1(T)}{g_1(T)}, & T &\mapsto H := \frac{f_2(T)}{g_2(T)} \end{aligned}$$

Here $f_1, f_2, g_1, g_2 \in \mathbb{Z}[X]$, i.e., J and H are given by rational functions evaluated at T. For a concrete example we refer to Example 1. We let

$$\begin{aligned} r_{1,\ell}(T,J) &:= g_1(T)J - f_1(T), \\ r_{2,\ell}(T,H) &:= g_2(T)H - f_2(T). \end{aligned}$$

We define the resultant polynomial

$$R_\ell(J,H) := \mathrm{Res}_T(r_{1,\ell}(T,J), r_{2,\ell}(T,H)) \in \mathbb{Z}[J,H].$$

Now consider two positive integers ℓ_1 and $\ell_2 \in \{2,3,5,7,13\}$. The polynomial we will use to detect orientations by orders containing $\mathbb{Z}[\ell_1\sqrt{-p\ell_2}]$ is the resultant

$$R_{\ell_1,\ell_2}(X,H) := \mathrm{Res}_J(R_{\ell_2}(J,H), \Phi_{\ell_1}(X,J)) \in \mathbb{Z}[X,H]. \tag{2}$$

Proposition 4. *Let $R_{\ell_1,\ell_2}(X,Y) \in \mathbb{Z}[X,Y]$ be defined as in Equation (2). The degree of $R_{\ell_1,\ell_2}(X,Y)$ in X is $2(\ell_1+1)$, and the degree in Y is $(\ell_1+1)(\ell_2+1)$.*

[2] Theoretically, we only need $X_0(\ell)^+$ to have genus zero since our final polynomial only requires a single parameter input for points on $X_0(\ell)^+$. However, the particular method that we use to construct this polynomial requires $X_0(\ell)$ to have genus zero as well, so in practice we restrict to this case.

Proof. For two univariate polynomials $P(u)$ and $Q(u)$ over a field k of degrees m and n respectively, the resultant of P and Q in u is bihomogenous of bidegree (m, n) in the coefficients of Q and of P, meaning it is homogenous of degree m in the coefficients of Q and homogenous of degree n in the coefficients of P. The map $X_0(\ell_2) \to X(1)$ is $\ell_2 + 1$ to 1 so the degree of $r_{1,\ell_2}(T, J)$ is $\ell_2 + 1$ in T and 1 in J. Similarly, the map $X_0(\ell_2) \to X_0(\ell_2)^+$ is 2 to 1, so the degree of r_{2,ℓ_2} in T is 2 and its degree in H is 1. The degree of J in R_{ℓ_2} is therefore 2, and the degree of H in R_{ℓ_2} is $\ell_2 + 1$. The polynomial Φ_{ℓ_1} has degree $\ell_1 + 1$ in both X and J, so the degree of R_{ℓ_1,ℓ_2} is $2(\ell_1 + 1)$ and $(\ell_2 + 1)(\ell_1 + 1)$ in X and H, respectively. □

It is important to note that the polynomial $R_{\ell_1,\ell_2}(X, Y)$ has *integer* coefficients. We are therefore able to reduce these polynomials modulo p for any prime p. We denote the reduction by $R_{\ell_1,\ell_2,p}$. As noted previously, for $p \nmid \ell_1, \ell_2$, the moduli interpretations of the functions used to construct $R_{\ell_1,\ell_2}(X, Y)$ remain valid modulo p. We will use this fact in the proofs of Lemma 2 and Theorem 1.

Lemma 2. *Let j_0 be a supersingular j-invariant in $\overline{\mathbb{F}_p}$. Then $R_{\ell_1,\ell_2,p}(j_0, h)$ has an $\mathbb{F}_p$-factor if and only if there exist ℓ_1-isogenies ϕ_1, ϕ_2 with domain E_0 and an ℓ_2-isogeny ψ with domain equal to the codomain of ϕ_1, such that the codomain of ϕ_2 is the domain of either $\psi^{(p)}$ or $\hat{\psi}^{(p)}$.*

Proof. The parameter H on $X_0^+(\ell_2)$ determines an equivalence class under duals of isogenies between elliptic curves over $\mathbb{C}$, and parameter obtained by evaluating H and then reducing modulo p, which we denote by h, therefore determines an equivalence class of isogenies between elliptic curves over $\overline{\mathbb{F}_p}$ under duals. For any h corresponding to isogenies between supersingular elliptic curves, the Galois orbit is given by $\{h, h^p\}$. Thus $R_{\ell_1,\ell_2,p}(j_0, h)$ has an $\mathbb{F}_p$-factor if and only if there exists h_0 such that both h_0 and h_0^p are roots of $R_{\ell_1,\ell_2,p}(j_0, h)$.

We show that this happens only under the conditions described in the Lemma statement. First, suppose that we have such an h_0, and let $\{\psi, \hat{\psi}\}$ be the pair of isogenies corresponding to h_0. Then there exists a j-invariant j_1 such that $\Phi_{\ell_1}(j_0, j_1) = R_{\ell_2}(j_1, h_0) = 0$. It follows that there is an ℓ_1-isogeny $\phi_1 : E_0 \to E(j_1)$, and that j_1 is the j-invariant of the domain of either ψ or $\hat{\psi}$. Without loss of generality, suppose that j_1 is the j-invariant of the domain of ψ. Then by a similar argument using h_0^p, we see that there is an ℓ_1-isogeny ϕ_2 from E_0 to a curve E' whose j-invariant is either the domain of $\psi^{(p)}$ or the domain of $\hat{\psi}^{(p)}$, and so the result holds. For the converse, note that if we have such a set of isogenies ϕ_1, ϕ_2, ψ, then the parameters h_ψ and h_ψ^p associated to ψ and ψ^p both satisfy $R_{\ell_1,\ell_2}(j_0, h) = 0$, so $R_{\ell_1,\ell_2}(j_0, h)$ has an $\mathbb{F}_p$-factor. □

Theorem 1. *Let j_0 be a supersingular j-invariant in $\overline{\mathbb{F}_p}$, and ℓ_1, ℓ_2 be primes. Then $R_{\ell_1,\ell_2,p}(j_0, h)$ has a factor defined over $\mathbb{F}_p$ if and only if $E_0 \coloneqq E(j_0)$ has an endomorphism of degree $\ell_1^2 p$ or $\ell_1^2 \ell_2 p$.*

Proof. Using the notation of Lemma 2, we proceed in cases depending on whether the codomain of ϕ_2 is the domain of $\psi^{(p)}$ or the domain of $\hat{\psi}^{(p)}$. In the first case, the endomorphism $\hat{\phi_2}\pi\phi_1$ of E_0 has degree $\ell_1^2 p$, and in the second, the endomorphism $\hat{\phi_2}\pi\psi\phi_1$ has degree $\ell_1^2\ell_2 p$. Thus if there is an $\mathbb{F}_p$-factor of $R_{\ell_1,\ell_2}(j_0, h)$, then E_0 has an endomorphism of degree either $\ell_1^2 p$ or $\ell_1^2\ell_2 p$.

Conversely, if E_0 has an endomoprhism of degree $\ell_1^2 p$ or $\ell_1^2\ell_2 p$, then we can decompose such an endomorphism as either $\phi_2\pi\phi_1$, or $\phi_2\pi\psi\phi_1$, where ϕ_1, ϕ_2 have degree ℓ_1, and ψ has degree ℓ_2. In the first case, the isogenies $\phi_1, \hat{\phi}_2, \psi$ satisfy that the codomain of $\hat{\phi}_2$ is equal to the domain of $\psi^{(p)}$, where ψ is any ℓ_2-isogeny out of the codomain of ϕ_1. In the second case, $\phi_1, \hat{\phi}_2, \psi$ satisfy that the codomain of $\hat{\phi}_2$ is the domain of $\hat{\psi}^{(p)}$. □

Corollary 2. *Let* $D = \left\{\sqrt{-p}, \frac{1+\sqrt{-p}}{2}, \sqrt{-\ell_2 p}, \frac{1+\sqrt{-\ell_2 p}}{2}\right\}$, *and* $\ell_1 D$ *denote the set containing every element of* D *multiplied by* ℓ_1. *Assuming that* $p > 4\ell_1^2\ell_2$, *the polynomial* $R_{\ell_1,\ell_2,p}(j_0, h)$ *has an* $\mathbb{F}_p$*-rational factor if and only if* $E(j_0)$ *has a primitive orientation by* $\mathbb{Z}[\alpha]$, *where* $\alpha \in D \cup \ell_1 D$.

Proof. By Theorem 1 we have that $R_{\ell_1,\ell_2}(j_0, h)$ has an $\mathbb{F}_p$-factor if and only if $E(j_0)$ has an endomorphism of degree $\ell_1^2 p$ or $\ell_1^2\ell_2 p$. By the assumption that $p > 4\ell_1^2\ell_2$, such an endomorphism must have trace zero . Indeed, any inseparable endomorphism θ of E also has inseparable dual since E is supersingular, so $[t] = \theta + \widehat{\theta}$ is inseparable and hence $t \equiv 0 \pmod{p}$. On the other hand, if $\deg\theta = dp$ we also have the bound $t^2 \leq 4dp$ since the discriminant of θ is non-positive, so if t is nonzero then $p^2 \leq 4dp$ implying $p \leq 4d$. The discriminant of $\mathbb{Z}[\theta]$ is either $-4\ell_1^2 p$ or $-4\ell_1^2\ell_2 p$, and $\mathbb{Z}[\theta]$ is contained in either the field $\mathbb{Q}(\sqrt{-\ell_1^2 p}) = \mathbb{Q}(\sqrt{-p})$, of discriminant $-4p$ or $-p$, or $\mathbb{Q}(\sqrt{-\ell_1^2\ell_2 p}) = \mathbb{Q}(\sqrt{-\ell_2 p})$, of discriminant $-4\ell_2 p$ or $-\ell_2 p$. Thus $\mathbb{Z}[\theta]$ is maximal at every prime except possibly for ℓ_1 and 2. The corresponding primitive orientation therefore has discriminant $-p, -4p, -\ell_1^2 p, -4\ell_1^2 p, -\ell_2 p, -4\ell_2 p, -4\ell_1^2\ell_2 p$, or $-\ell_1^2\ell_2 p$, which correspond to the eight orders in the statement. For the converse, simply note that all of these orders contain an element of norm $\ell_1^2 p$ or of norm $\ell_1^2\ell_2 p$, and apply Theorem 1. □

The algorithm. We present OrientationFinder, which on input $j(E)$ and a valid pair (ℓ_1, ℓ_2) will output `true` if and only if E is oriented by $\mathbb{Z}[\alpha]$ where $\alpha \in D \cup \ell_1 D$ with D as above. Otherwise, it outputs `false`. It does this as follows: it fetches the resultant polynomial $R_{\ell_1,\ell_2,p}(X, Y)$, evaluates it at $X = j(E)$ to obtain a univariate polynomial $f(Y)$, and runs the RationalFactor routine from Section 2 on f. For completeness, we present the algorithm in full in **??**. Here, we are assuming that the corresponding resultant polynomial $R_{\ell_1,\ell_2,p} \in \mathbb{F}_p[X, Y]$ has been computed. Indeed, for each (ℓ_1, ℓ_2), R_{ℓ_1,ℓ_2} is stored, and its reduction modulo p is computed in precomputation (as it only depends on p and not the instance we are solving). The correctness of the algorithm follows from Corollary 2.

Lemma 3. *On input a* j*-invariant* $j \in \mathbb{F}_{p^2}$ *of a supersingular curve* E, *and a pair of integers* (ℓ_1, ℓ_2) *with* ℓ_1 *prime and* $\ell_2 \in \{2, 3, 5, 7, 13\}$, OrientationFinder *has cost at most*

$$6d_1 - 4 + 2N_{\ell_1,\ell_2,p} + (d_1 d_2 - 1)(2d_1 d_2 + 5)$$

$\mathbb{F}_p$*-multiplications with our cost metric, where* $N_{\ell_1,\ell_2,p}$ *is the number of monomials in* $R_{\ell_1,\ell_2,p}(X, Y)$, *and* $d_i = \ell_i + 1$ *for* $i = 1, 2$.

Proof. By Proposition 4, the degree of X in $R_{\ell_1,\ell_2,p}(X, Y)$ is $2d_1$, and there are $N_{\ell_1,\ell_2,p}$ monomials. To evaluate it at $X = j$, we first compute the powers $j^2, j^3, \ldots, j^{2d_1}$ with $2d_1 - 2$ multiplications in $\mathbb{F}_{p^2}$. Since the coefficients of $R_{\ell_1,\ell_2,p}$ lie in $\mathbb{F}_p$, obtaining f costs at most $6d_1 - 4 + 2N_{\ell_1,\ell_2,p}$ multiplications in $\mathbb{F}_p$. We then run RationalFactor on $f(Y)$, which is monic of degree $d = (\ell_1 + 1)(\ell_2 + 1) = d_1 d_2$, again by Proposition 4. By [CRSCS22, Prop. 2], this costs at most $(d - 1)(2d + 5)$ $\mathbb{F}_p$-multiplications. □

6 $\sqrt{-\star p}$-WayFinder, an instantiation of WayFinder

The goal of this section is to exhibit a concrete instantiation of $\mathcal{R}$-WayFinder. We refer to this instantiation as $\sqrt{-\star p}$-WayFinder since it involves searching for orientations by one or more number fields of the form $\mathbb{Q}(\sqrt{-dp})$. Concretely, we fix $\mathcal{R} = \mathcal{R}(\mathcal{L})$ to be the set of orders $R \supseteq \mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ for ℓ_1 an odd integer and $\ell_2 \in \{1, 2, 3, 5, 7, 13\}$. Following SuperSolver, we do not consider the pair $(2, \ell)$, because this intersects with the work done by walking in the 2-isogeny graph.

By combining our algorithms from Section 5 with NeighbourInFp, we are able to efficiently detect whether a curve is oriented by some $R \in \mathcal{R}$. We use this to solve the first step in $\mathcal{R}$-WayFinder. We then solve the second step of the $\mathcal{R}$-WayFinder algorithm by showing how to instantiate the algorithm in Section 4 for orders in $\mathcal{R}$. Together, these provide a concrete instance of $\mathcal{R}$-WayFinder. We analyse the cost of this instantiation, showing it gives a concrete improvement compared to both Delfs–Galbraith and SuperSolver. We explore the concrete complexity in detail in Section 7.

6.1 Instantiating the search and cost model for $\sqrt{-\star p}$-WayFinder

In SuperSolver the first step is to run a depth-first search in the 2-isogeny graph, and run NeighbourInFp(j, ℓ_1) for each j-invariant in the path for a well-chosen set of small, odd integers ℓ_1. $\sqrt{-\star p}$-WayFinder extends this approach by running ConjugateFinder(j, ℓ_2) and OrientationFinder(j, ℓ_1, ℓ_2) for a suitable set of ℓ_2 and pairs (ℓ_1, ℓ_2) at each step. For a concrete instance, we first precompute a list $\mathcal{L}$ of pairs of the form:

- $(\ell_1, 1)$ for odd $\ell_1 \in \mathbb{Z}_{>0}$, which corresponds to running NeighbourInFp(j, ℓ_1);
- $(1, \ell_2)$ with integer $\ell_2 \in \mathbb{Z}_{>0}$, which corresponds to running ConjugateFinder(j, ℓ_2);
- (ℓ_1, ℓ_2) with ℓ_1 as before and $\ell_2 \in \{2, 3, 5, 7, 13\}$; corresponds to OrientationFinder(j, ℓ_1, ℓ_2).

We give pseudocode in Algorithm 3.

Algorithm 3 $\sqrt{-\star p}$-WayFinder

Input: The j-invariant $j \in \mathbb{F}_{p^2}$ of a supersingular curve E, a search depth d, and a precomputed list $\mathcal{L}$ of pairs (ℓ_1, ℓ_2) with the corresponding modular and/or resultant polynomials reduced modulo p.

Output: An oriented supersingular elliptic curve E' and corresponding path $E = E_0, E_1, \ldots, E_d = E'$

if $j(E) \in \mathbb{F}_p$ **then return** E
$j_0, j_1, j_2 \leftarrow$ roots of $\Phi_2(j(E), Y))$
$r \leftarrow 1$
$S \leftarrow [(j(E), j_0, r), (j(E), j_1, r), (j(E), j_2, r)]$
`found` $\leftarrow$ `false`
while $S \neq \emptyset$ **do**
    $(j_{\text{prev}}, j, r) \leftarrow$ pop(`stack`)
    if $j \in \mathbb{F}_p$ **then**
        `found` $\leftarrow$ `true`
    for $(\ell_1, \ell_2) \in \mathcal{L}$ **do**
        if $\ell_1 = 1$ **then**
            `found` $\leftarrow$ ConjugateFinder(j, ℓ_2)
        else if $\ell_2 = 1$ **then**
            `found` $\leftarrow$ NeighbourInFp(j, ℓ_1)
        else
            `found` $\leftarrow$ OrientationFinder(j, ℓ_1, ℓ_2)
    if `found` **then**
        break
    if $r < d$ **then**
        $j_0, j_1 \leftarrow$ roots of $\Phi_2(j, Y)/(Y - j_{\text{prev}})$
        Append $(j, j_0, r+1)$ and $(j, j_1, r+1)$ to `stack`
Parse a path $E = E_0, E_1 \ldots, E'$ from `stack`
return $E_0, \ldots, E'$

The choice of $\mathcal{L}$ is crucial to the algorithm's performance. As in Section 3, we perform depth-first search for a curve in $V(\mathcal{R}(\mathcal{L}))$ in the 2-isogeny graph, and we test for $\mathbb{F}_p$-rationality of a node. Using the notation from Section 3, we estimate the total number of $\mathbb{F}_p$-multiplications by $M_{\mathcal{L}} := C_{\mathcal{L}} \cdot N_{\mathcal{L}}$. To minimise the total cost of the algorithm, we want to minimise $M_{\mathcal{L}}$. The optimal choice of $\mathcal{L}$ depends only on p and not on the input curve, so we can find $\mathcal{L}$ during precomputation.

6.1.1 Computing the cost per step

Let $\mathsf{cost}_{\mathsf{step}}$ be the cost of taking a step in the 2-isogeny graph, and let $\mathsf{cost}_{\mathcal{L}}$ denote the cost of running the detection for all $(\ell_1, \ell_2) \in \mathcal{L}$, using our cost metric of counting $\mathbb{F}_p$-multiplications. As in Section 3, we model the cost at each node as $\mathsf{cost}_{\mathsf{step}}/2 + \mathsf{cost}_{\mathcal{L}}$. Note that $\mathsf{cost}_{\mathsf{step}}$ depends only on p, while $\mathsf{cost}_{\mathcal{L}}$ depends only on (ℓ_1, ℓ_2) and is given by Lemma 1 if $\ell_1 = 1$, [CRSCS22, pg. 18] if $\ell_2 = 1$, and Lemma 3 otherwise.

Remark 4. In practice, at each step we precompute all the necessary powers of j and $j^{(p)}$ for some $j \in \mathbb{F}_{p^2}$, and then perform the detection for each pair in $\mathcal{L}$. Thus, $\mathsf{cost}_{\mathcal{L}}$ is given by adding this cost of powering to the cost of performing the detection once the required powers are computed.

6.1.2 Computing $N_{\mathcal{L}}$

From the discussion in Section 3, to determine the expected number of steps for a set of pairs $\mathcal{L}$, we need to determine the set of orders $\mathcal{R} = \mathcal{R}(\mathcal{L})$ and count the number of curves orientable by some order in $\mathcal{R}$.

We assume that if $(\ell_1, \ell_2) \in \mathcal{L}$ and $\ell_i > 1$, then $(\ell_1, 1), (1, \ell_2) \notin \mathcal{L}$. Otherwise, we would be detecting orientability for the same orientation twice. Furthermore, we assume that $(2, \ell) \notin \mathcal{L}$ for any ℓ, i.e., we assume ℓ_1 is odd. Indeed, detection performed by these pairs intersects with the work done by walking in the 2-isogeny graph.[3]

To deduce $N_{\mathcal{L}}$ for this restricted set $\mathcal{L}$, we must first count $\#V(\mathcal{L})$. It will be easier, however, to compute the cardinality of $\mathrm{SS}(\mathcal{R}(\mathcal{L})) := \bigcup_{R \in \mathcal{R}(\mathcal{L})} \mathrm{SS}(R)$, where $\mathrm{SS}(R)$ denotes the set of supersingular curves with an orientation by R. For a quadratic order R, recall $H(R) = \sum_{R' \supseteq R} h(R')$ is the Hurwitz class number of R, where the sum is over orders containing R. If D is a discriminant, let $H(D) = H(R_D)$ where R_D has discriminant D.

Proposition 5. *Let $\mathcal{L}$ be a set of pairs. Assume that if $(\ell_1, \ell_2) \in \mathcal{L}$ and $\ell_i > 1$ then $(\ell_1, 1), (1, \ell_2) \notin \mathcal{L}$. Assume also that for any ℓ, $(2, \ell) \notin \mathcal{L}$. Define*

$$w_{\mathcal{L}}(D) := 1 + \sum_{\substack{\ell_1 > 1 \\ (\ell_1, \ell_2) \in \mathcal{L}}} \left(\ell_1 - \left(\frac{D}{\ell_1} \right) \right)$$

The number of orientations detected is

$$\#\,\mathrm{SS}(\mathcal{R}(\mathcal{L})) = w_{\mathcal{L}}(-4p)H(-4p) + \sum_{\substack{\ell_2 > 1: \\ \exists \ell_1 : (\ell_1, \ell_2) \in \mathcal{L}}} w_{\mathcal{L}}(-4\ell_2 p)H(-4\ell_2 p).$$

Proof. We use $h(D)$ for the class number of the quadratic order of discriminant D. For convenience, we set $h(D) = 0$ if D is not a discriminant. Define three sets

$$\mathcal{L}_1 = \{\ell : (\ell, 1) \in \mathcal{L}\}, \quad \mathcal{L}_2 = \{\ell : (1, \ell) \in \mathcal{L}\}, \quad \mathcal{L}_{12} = \{(\ell_1, \ell_2) \in \mathcal{L} : \ell_i > 1\}.$$

[3] More generally, if we walk in the ℓ-isogeny graph for some ℓ, we do not consider pairs of the form $(\ell, -)$.

We count $\#\,\mathrm{SS}(\mathcal{R}(\mathcal{L}))$ by counting the contribution to $\mathcal{R}(\mathcal{L})$ for each $\ell \in \mathcal{L}_1 \cup \mathcal{L}_2$ and each pair $(\ell_1, \ell_2) \in \mathcal{L}_{12}$. If $\ell \in \mathcal{L}_1$ then we test for orientability by orders containing $\mathbb{Z}[\ell\sqrt{-p}]$. If $\ell \in \mathcal{L}_2$ we test for orientability of orders containing $\mathbb{Z}[\sqrt{-\ell p}]$. And if $(\ell_1, \ell_2) \in \mathcal{L}_{12}$ then we test for orientability by the orders containing $\mathbb{Z}[\ell_1\sqrt{-p}]$ or $\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$. Each orientation in $\mathrm{SS}(\mathcal{R}(\mathcal{L}))$ is primitive for some order.

The number of primitive orientations by a fixed order R in $\mathcal{R}(\mathcal{L})$ is the class number of R since p is ramified in each order of $\mathcal{R}(\mathcal{L})$ and prime to each conductor. If $\ell \in \mathcal{L}_1$ then ℓ contributes $H(\mathbb{Z}[\ell\sqrt{-p}])$ to $\#\,\mathrm{SS}(\mathcal{R}(\mathcal{L}))$. If there are multiple primes in $\mathcal{L}_1$ then each additional $\ell \in \mathcal{L}_1$ after the first contributes $H(\mathbb{Z}[\ell\sqrt{-p}]) - H(\mathbb{Z}[\sqrt{-p}])$. If $\ell \in \mathcal{L}_2$ then it contributes $H(\mathbb{Z}[\sqrt{-\ell p}])$ to $\#\,\mathrm{SS}(\mathcal{R}(\mathcal{L}))$. And if $(\ell_1, \ell_2) \in \mathcal{L}_{12}$ then it contributes $H(\mathbb{Z}[\ell_1\sqrt{-p}]) + H(\mathbb{Z}[\ell_1\sqrt{-\ell_2 p}])$ to $\#\,\mathrm{SS}(\mathcal{R}(\mathcal{L}))$. We assumed that if $(\ell_1, \ell_2) \in \mathcal{L}_{12}$ then $\ell_1 \notin \mathcal{L}_1$ and $\ell_2 \notin \mathcal{L}_2$. Thus we have overcounted by $H(\mathbb{Z}[\sqrt{-p}])$. Moreover, if $(\ell_1, \ell_2') \in \mathcal{L}_{12}$ then we overcount by an additional $H(\mathbb{Z}[\ell_1\sqrt{-p}]) - H(\mathbb{Z}[\sqrt{-p}])$, and if $(\ell_1', \ell_2) \in \mathcal{L}_{12}$ we overcount by $H(\mathbb{Z}[\sqrt{-\ell_2 p}])$.

Using the relative class number formula [Cox22, Corollary 7.28], we have $H(\ell_1\sqrt{-\ell_2 p}) - H(\sqrt{-p}) = \left(\ell_1 - \left(\frac{-\ell_2 p}{\ell_1}\right)\right)$ (for $\ell_2 = 1$ or a prime), and the result follows. □

Remark 5. Including pairs of the form $(2, \ell)$ complicates the clustering analysis as the subgraph of curves with an orientation by $\mathbb{Z}[2\sqrt{-\ell p}]$ will have connected components that are volcanoes of depth 2, a structure we did not consider in Section 3. We expect the techniques in Section 3 for analyzing the hitting time to apply just as well, but we omit this in favor of a consistent exposition. The formula for the contribution to $\mathrm{SS}(O(\mathcal{L}))$ from pairs of the form $(2, \ell)$ must be modified as well, as we now explain. We calculate that

$$H(\mathbb{Z}[2\sqrt{-\ell p}]) - H(\mathbb{Z}[\sqrt{-\ell p}]) = \begin{cases} 2H(\mathbb{Z}[\sqrt{-\ell p}]) & : -\ell p \not\equiv 1 \pmod 4 \\ H(\mathbb{Z}[\sqrt{-\ell p}]) & : -\ell p \equiv 1 \pmod 8 \\ \frac{3}{2}H(\mathbb{Z}[\sqrt{-\ell p}]) & : -\ell p \equiv 5 \pmod 8 \end{cases}.$$

Following the last line of the proof of Proposition 5, we use this to modify the count of $\mathrm{SS}(O(\mathcal{L}))$ to include the case where $(2, \ell) \in \mathcal{L}$ for some ℓ; it suffices to re-define

$$w_{\mathcal{L}}(D) := 1 + \epsilon_{\mathcal{L}}(p, \ell) + \sum_{\substack{\ell_1 > 1 \\ \exists \ell_2 : (\ell_1, \ell_2) \in \mathcal{L}}} \left(\ell_1 - \left(\frac{D}{\ell_1}\right)\right)$$

where

$$\epsilon_{\mathcal{L}}(p, \ell) = \begin{cases} 0 & : (2, \ell) \notin \mathcal{L} \\ 2 & : (2, \ell) \in \mathcal{L},\ -p\ell \not\equiv 1 \pmod 4 \\ 1 & : (2, \ell) \in \mathcal{L},\ -p\ell \equiv 1 \pmod 8 \\ \frac{3}{2} & : (2, \ell) \in \mathcal{L},\ -p\ell \equiv 5 \pmod 8 \end{cases}$$

Then the formula for $\#\,\mathrm{SS}(O(\mathcal{L}))$ is as in the statement of Proposition 5.

If $(2, \ell) \in \mathcal{L}$ for some ℓ then we should also subtract $H(-4p)$ from the number of orientations detected when trying to optimize the set $\mathcal{L}$. This is because we have assumed we perform depth-first search in the 2-isogeny graph, running the battery of tests at each node in the search tree. The $(2, \ell)$ test includes a test for $\mathbb{F}_p$-rationality and a test for being 2-isogenous to a $\mathbb{F}_p$-rational curve, meaning we never visit a $\mathbb{F}_p$-rational curve: we would have detected this one level above, and terminated. Essentially we are instead interested in the event that the node has an orientation by some order, given that it is not $\mathbb{F}_p$-rational. In our optimization, we never include the pair $(2, 1)$ for this reason, and we can modify the counting when considering the pair $(2, \ell)$ for $\ell > 1$. For simplicity, we consider sets with $(2, -) \notin \mathcal{L}$.

We estimate $\#V(\mathcal{L}) \approx \frac{1}{2}\#\operatorname{SS}(\mathcal{R}(\mathcal{L}))$. If E has an orientation by $R \in \mathcal{R}$ then conjugation gives a second, non-isomorphic, orientation. We implicitly assume that the number of curves with two or more non-conjugate orientations is negligible.

The following proposition shows that the number of such curves is bounded by a constant, depending only on $\mathcal{L}$ and not on p. Additionally, experimental evidence shows that in instances of WayFinder with $\ell_1, \ell_2 \in \{1, 2, 3, 5, 7, 13\}$, the number of these curves is a vanishingly small portion of $V(\mathcal{R})$ for primes of bitsize larger than 20.

Proposition 6. *Let $q_{max} = \max\{q : (q, r) \in \mathcal{L}\}$, let $r_{max} = \max\{r : (q, r) \in \mathcal{L}\}$, and let $L = \#\mathcal{L}$. Assume that $p > 16^8 q_{max} r_{max}^4$. Then*

$$\#V(\mathcal{L}) \geq \frac{1}{2}\#\operatorname{SS}(\mathcal{R}(\mathcal{L})) - 96L(18L(L-1)-1)(q_{max}r_{max})^{5/2}.$$

Proof. We split $V(\mathcal{L})$ into two sets, V_{good} and V_{bad}, according to the size of the fiber of the map $(E, \iota) \mapsto j(E)$. For $j \in V_{good}$ we have a fiber size of 2. For $j \in V_{bad}$ we have a larger fiber, of size at most C (to be determined, but a constant dependent on $\mathcal{L}$ but not on p). Then

$$\#\operatorname{SS}(\mathcal{R}(\mathcal{L})) \leq 2 \cdot \#V_{good} + C\#V_{bad} = 2\#V(\mathcal{L}) + (C-2)\#V_{bad}.$$

so

$$\#V(\mathcal{L}) \geq \frac{1}{2}\#\operatorname{SS}(\mathcal{R}(\mathcal{L})) - (C/2-1)\#V_{bad}.$$

Thus we must: 1) bound the number of j-invariants with "large" fiber, and 2) find C, a bound for the fiber size.

The number of isomorphism classes of primitive orientations $(E, \iota\colon R \hookrightarrow \operatorname{End}(E))$ over $\overline{\mathbb{F}_p}$ is given by the class number of R, since p ramifies in R but does not divide the conductor of R. Generically, we expect there to be either zero or two such isomorphism classes of primitive orientation of E for any Δ; if (E, ι) is an orientation by R then so is $(E, \bar{\iota})$, and these orientations are not isomorphic.

Suppose (E, ι) and (E', ι') are two primitive orientations by $R, R' \in \mathcal{R}(\mathcal{L})$ and that (E, ι') is not isomorphic to (E, ι) or $(E, \widehat{\iota})$. Then non-integer endomorphisms in $\iota(R)$ do not commute with non-integer endomorphisms in $\iota'(R')$. For some $r \geq 1$ square-free there is a minimal $q \geq 1$ such that $\mathbb{Z}[q\sqrt{-pr}] \subseteq R$ and $\iota(q\sqrt{-pr}) \in \operatorname{End}(E)$. There exists similar q', r' for (E, ι'). Then the endomorphisms $\theta = \iota(q\sqrt{-rp})$ and $\theta' = \iota(q'\sqrt{-r'p})$ are cyclic. Then $\theta\theta'$ has degree p^2N, where $p \nmid N$. By [Sil09, Corollary II.2.12 & Theorem V.3.1] it follows that $\theta\theta' = p\gamma$ for some separable endomorphism γ. Moreover $\gamma \notin \mathbb{Z}$ as otherwise (E, ι) would be isomorphic to either (E, ι) or $(E, \widehat{\iota})$. Indeed, if α, β are cyclic, non-integer endomorphisms and $\alpha\beta = n \in \mathbb{Z}$, then $(\deg\alpha)\beta = n\widehat{\alpha}$ as endomorphisms so considering the invariant factors of their kernels we must have $n = \pm\deg\alpha$ and $\beta = \pm\widehat{\alpha}$. This is because the largest N such that $E[N]$ is contained in $\ker n\widehat{\alpha}$ is $|n|$ since $\ker\widehat{\alpha}$ is cyclic; similarly the largest N such that $E[N]$ is contained in $\ker((\deg\alpha)\beta)$ is $(\deg\alpha)$.

This implies E has a nontrivial cyclic endomorphism of degree dividing $B = (qq')^2rr'$. For each integer N dividing B, and for each $0 \leq t < 2N$, let $D_t = t^2 - 4N$; this is a negative discriminant of a binary quadratic form that nontrivially represents N, and $|D_t| \leq 4B$. For each t, there are $h(D_t) = \#\operatorname{Cl}(D_t)$ many equivalence classes of such forms corresponding to (at most) $h(D_t) \leq 4|D_t|/3 \leq 16B/3$ many isomorphism classes of supersingular elliptic curves orientable by R_{D_t} (the first bound on $h(D_t)$ follows from the bounds on the coefficients of reduced forms, as in [Cox22, Theorem 2.8]). The number of divisors d of B is either $3 \cdot 3 \cdot 2 \cdot 2 = 36$, $5 \cdot 2 \cdot 2 = 20$, $3 \cdot 3 \cdot 3 = 27$, or $5 \cdot 3 = 15$ depending on the cases whether or not $q = q'$ or $r = r'$.

The largest norm possible is $q_{max}^4 r_{max}^2$, so the largest discriminant in absolute value is $-4q_{max}^4 r_{max}^2$. There are $L = \#\mathcal{L}$ pairs, so we count

$$\underbrace{L(L-1)/2}_{\#\text{ unordered pairs } (q,r),(q',r')} \cdot \underbrace{36}_{\text{possible norms}} \cdot \underbrace{2q_{max}^2 r_{max}}_{\text{discriminants per norm}} \cdot \underbrace{16q_{max}^4 r_{max}^2/3}_{\text{curves per discriminant}}$$

giving a total upper bound of $3 \cdot 2^6 L(L+1) q_{max}^6 r_{max}^3$ for the number of j-invariants with fiber size larger than 2.

But how large can this fiber be? Note that if E is primitively oriented by $R \in \mathcal{R}(\mathcal{L})$ then E has a $q^2 r$-isogeny to $E^{(p)}$. This $q^2 r$-isogeny ψ induces both the orientation (E, ι) and $(E, \bar{\iota})$ since $\pi\psi$ has trace 0, so $\widehat{\pi\psi} = -\pi\psi$. Now suppose E has two more orientations (E, ι_2) and (E, ι_3) and denote $\iota_1 = \iota$ and $\psi_1 = \psi$. Then we have isogenies $\psi_i \colon E \to E^{(p)}$ of degree $q_i^2 r_i$. The curve E has at most $q(q+1)(r+1)$ cyclic $q^2 r$-isogenies. We then have two endomorphisms $\rho_i = \widehat{\psi_i} \circ \psi_1$ for $i = 2, 3$. These endomorphisms must commute, since otherwise we have

$$4p \leq \operatorname{disc} \gamma_2 \cdot \operatorname{disc} \gamma_3 \leq 16 \deg \gamma_2 \deg \gamma_3 = q_1^4 q_2^2 q_3^2 r_1^2 r_2 r_3,$$

(following the argument of Kaneko in [Kan89]) which we assume to not be the case. By a similar argument if ι_4 is yet a fourth orientation corresponding to an isogeny $\psi_4 \colon E \to E^{(p)}$ of degree $q_4^2 r_4$, the endomorphism $\gamma_4 = \widehat{\psi_4}\psi_2$ must commute with γ_2. So we are left to bound the number of primitive elements of given norm in the field $Q(\gamma_2)$, where the norm is of the form $q^2 q'^2 r r'$ where we let $(q, r), (q', r') \in \mathcal{L}$ vary. An element α of norm N gives rise to an ideal (α) of norm N, so we will upper bound our count by a count of all ideals of given norm. If a given norm is $\prod \ell_i^{e_i}$ then in the worst case each ℓ_i splits in K and there are $\tau(N) = \prod_i (e_i + 1)$ many ideals of that norm. In the worst case, all the integers q, q', r, r' are distinct split primes, giving rise to $3 \cdot 3 \cdot 2 \cdot 2 = 36$ ideals, yielding $2 \cdot 36 = 72$ elements of norm $q^2 q'^2 r r'$. There are $L(L-1)/2$ many ways to choose two elements of $\mathcal{L}$ with replacement, so the fiber size has at most $36L(L-1)$.

□

Remark 6. When $\mathcal{L}$ is a subset of $S = \{(\ell_1, \ell_2) : \ell_1, \ell_2 \in \{1, 2, 3, 5, 7, 13\}\}$, given a prime p we can explicitly compute an upper bound on the difference between $\frac{1}{2}\,\mathrm{SS}(\#O(\mathcal{L}))$ and $\#V(\mathcal{L})$. We begin by taking $\mathcal{L} = S$. Using Sage, we compute all possible D_t and determine the roots of the associated Hilbert class polynomials. For each unique root, we check whether the associated elliptic curve isomorphism class has an orientation by the order corresponding to (ℓ_1, ℓ_2) for each such pair in S. Running this code for 10 instances of primes $2^{20} \leq p \leq 2^{25}$, the number of curves with orientations by two or more orders was only a few hundred, and no curve had orientations by more than four orders.

By applying Proposition 5, we obtain the expected number of nodes visited. After including the clustering weights $c_2(\ell p) := c_2(\mathbb{Q}(\sqrt{-\ell p}))$, we have

$$N_{\mathcal{L}} = 2 \cdot \#V \cdot \left(c_2(p) w_{\mathcal{L}}(-4p) H(-4p) + \sum_{\substack{\ell_2 \\ (\ell_1, \ell_2) \in \mathcal{L}}} c_2(\ell_2 p) w_{\mathcal{L}}(-4\ell_2 p) H(-4\ell_2 p) \right)^{-1}.$$

6.1.3 Computing versus estimating class numbers

When running our cost model for precomputing the set $\mathcal{L}$, we are required to compute the class number $h(D)$ of a quadratic order of discriminant D. In particular, we have $D = -\ell_2 p$ or $-4\ell_2 p$. Although this is feasible for small primes, it quickly becomes infeasible. Indeed, computing $h(D)$ has subexponential classical cost [HM89] (under GRH), or polynomial quantum cost [Hal05]. As a result, to compute our cost model for cryptographically sized primes we are required to estimate $h(D)$.

Remark 7. We remark that computing the class numbers needed to run our precomputation using our cost model is exponentially cheaper than solving the isogeny problem. Therefore, an attacker looking to attack the isogeny problem for a particular prime p would compute the relevant class numbers to obtain the most accurate predictions in precomputation.

The naive estimate $h(D) \approx \sqrt{|D|}$ is not sufficiently accurate for our purposes. To improve this estimate, we begin with the formula of Dirichlet [Dav00, p.50, Equation 15]: for a fundamental discriminant $D < 0$, we have

$$h(D) = \frac{\#R_D^\times \sqrt{|D|}}{2\pi} L(1, \chi_D)$$

where $L(1, \chi_D)$ denotes the L-function $L(s, \chi_D) = \prod_q (1 - \chi_D(q)q^{-s})^{-1}$ of the character $\chi_D = \left(\frac{D}{\cdot}\right)$ of $\mathbb{Q}(\sqrt{D})$ for $\mathrm{Re}(s) > 1$, and its analytic continuation otherwise [Dav00]. Thus to estimate the expectation of $h(D)$ we estimate $\mathbb{E}[L(1, \chi_D)]$. Granville and Soundararajan [GS03, pg. 2-3] model $L(1, \chi_D)$ as a random Euler product $L = \prod_q (1 - X(q)q^{-1})^{-1}$, where $X(q)$ is the random variable that is ± 1 with probability $q/(2(q+1))$ respectively and 0 with probability $(q+1)^{-1}$. They compare moments of $L(1, \chi_D)$ with moments of $L(1, X)$ for D varying over the family of all fundamental discriminants.

We, however, are considering only discriminants $-p$ $-4p$, $-8p$, $-\ell p$ or $-4\ell p$ for a small odd prime ℓ. Note that for a prime $q \neq 2, \ell, p$, the density of fundamental discriminants D such that $\chi_D(q) = 0$ will be positive, while if we restrict to discriminants of the above form, this density will be 0. Let us first consider the case where D varies over fundamental discriminants of the form $D = -p$ or $D = -4p$ where p is an odd prime. The other cases are similar. We let $X(q)$ be the random variable taking on values ± 1 with probability $1/2$ respectively. Then the expectation of the "random local Euler factor" $(1 - X(q)q^{-1})^{-1}$ is

$$\left(\frac{q-1}{q}\right)^{-1} \frac{1}{2} + \left(\frac{q+1}{q}\right)^{-1} \frac{1}{2} = \frac{1}{2}\left(\frac{2q^2}{q^2-1}\right) = \left(1 - q^{-2}\right)^{-1},$$

and we approximate the expectation of $L(1, X) = \prod_q \left(1 - X(q)q^{-1}\right)^{-1}$ by $\prod_q \left(1 - q^{-2}\right)^{-1} = \zeta(2) = \pi^2/6$.

We can condition on the congruence class of the discriminant modulo 8 to obtain $\mathbb{E}[L(1, \chi_D) \mid D \bmod 8]$: we simply divide out by the expectation of $X(2)$ which is $(1 - 2^{-2})^{-1} = 4/3$ and multiply by $(1 - \left(\frac{D}{2}\right))^{-1}$. This gives

$$\mathbb{E}[L(1, \chi_D) \mid D \bmod 8] \approx \frac{\pi^2}{8}\left(1 - \left(\frac{D}{2}\right)\frac{1}{2}\right)^{-1}.$$

We repeat the derivation of $\mathbb{E}[L(1, \chi_D)]$, now for discriminants of the form $-\ell p$ or $-4\ell p$ for a prime $\ell \neq p$. We replace the quadratic character $\chi_D(q)$ with a random variable $X(q)$ that is ± 1 with probability $1/2$ for $q \neq \ell$ and is 0 for $q = \ell$, since $\chi_D(\ell) = 0$. We then model

$$\mathbb{E}[L(1, \chi_D)] \approx \prod_{q \neq \ell} (1 - q^{-2})^{-1} = \zeta(2)(1 - \ell^{-2}).$$

If $\ell \neq 2$ and we wish to condition our estimate for the expectation of $L(1, \chi_D)$ on $D \pmod 8$ then we instead model

$$\mathbb{E}[L(1, \chi_D) \mid D \pmod 8] \approx \left(\prod_{q \neq 2, \ell} (1 - q^{-2})^{-1}\right)\left(1 - \left(\frac{D}{2}\right)\frac{1}{2}\right)^{-1} = \frac{3(1 - \ell^{-2})\zeta(2)}{4}\left(1 - \left(\frac{D}{2}\right)\frac{1}{2}\right)^{-1},$$

which simplifies to $\frac{\pi^2}{8}(1 - \ell^{-2})\left(1 - \left(\frac{D}{2}\right)\frac{1}{2}\right)^{-1}$. When $\ell = 2$, then 2 always ramifies in $\mathbb{Q}(\sqrt{-2p})$, and we estimate

$$\mathbb{E}[L(1, \chi_D)] \approx \prod_{q \neq 2} (1 - q^{-2})^{-1} = \frac{\pi^2}{8}.$$

To summarize, we estimate

$$\mathbb{E}[L(1,\chi_D) \mid D \bmod 8] \approx \begin{cases} \frac{\pi^2}{8\left(1-\left(\frac{D}{2}\right)\frac{1}{2}\right)} & : D = \operatorname{disc}\mathbb{Q}(\sqrt{-p}) \\ \frac{\pi^2}{8} & : D = \operatorname{disc}\mathbb{Q}(\sqrt{-2p}) \\ \frac{\pi^2\left(1-\ell^{-2}\right)}{8\left(1-\left(\frac{D}{2}\right)\frac{1}{2}\right)} & : D = \operatorname{disc}\mathbb{Q}(\sqrt{-\ell p}), \ell \text{ odd} \end{cases}.$$

To validate our estimates for $\mathbb{E}[L(1,\chi_D)]$, we did the following: for each $\ell \in \{1,2,3,5\}$ and for bit lengths $10 \leq b < 40$, we generated 30 random primes of bit length b. For each such prime we compute $h(\mathbb{Q}(\sqrt{-\ell p}))$ and sort these into classes according to whether the fundamental discriminant $\operatorname{disc}(\mathbb{Q}(\sqrt{-\ell p})$ is even or 1 or 5 modulo 8, i.e., whether 2 is ramified, split, or inert. We compute the sample mean of $h(\mathbb{Q}(\sqrt{-\ell p}))$ in each class. Plots of the data, as well as estimates of the predicted and observed means in each class, are displayed in Figures 8 to 11.

Remark 8. While we can understand the expectation of $L(1,\chi_D)$, Littlewood proved that under GRH [Lit28], we have $\sqrt{|D|}(\log\log|D|)^{-1} \ll h(D) \ll \sqrt{|D|}\log\log|D|$. The value $h(D)$ varies within this range, and these small changes impact the predicted optimal sets.

Remark 9. As shown by Equation (1), to compute the relative cost of $\sqrt{-\star p}$-WayFinder versus Solver running only SuperSolver-like pairs, we do not need to compute class numbers. Therefore, for this restricted set of pairs, we can accurately predict the relative improvements of $\sqrt{-\star p}$-WayFinder. For instance, for the SQIsign Level-I prime, we predict that the optimal set is $\{3,5,7,11,13\}$, with predicted ratio 0.12.

6.2 The second step

6.2.1 The class group action for $\mathbb{Z}[\sqrt{-dp}]$-orientations

We discuss how to efficiently detect, compute, and represent orientations by the order $\mathbb{Z}[\sqrt{-dp}]$ for small squarefree integers d prime to p. Indeed, we can assume we have computed isogenies $E_i \to E_i'$ where E_i' are oriented by orders of the form $\mathbb{Z}[\ell_{1i}\sqrt{-\ell_{2i}p}]$; possibly after using ascending ℓ_{1i}-isogenies we can assume E_i' are oriented by $\mathbb{Z}[\sqrt{-\ell_{2i}p}]$, and in fact we can reduce to the case that both are oriented by the same order $\mathbb{Z}[\sqrt{-dp}]$. We pick models for our elliptic curves defined over $\mathbb{F}_{p^2}$ in one of the isogeny classes corresponding to $\pm 2p$ so that all endomorphisms and isogenies of our elliptic curves are defined over $\mathbb{F}_{p^2}$.

An orientation $\iota\colon \mathbb{Z}[\sqrt{-dp}] \hookrightarrow \operatorname{End}(E)$ is determined by $\alpha = \iota(\sqrt{-dp})$. Then α is inseparable and factors as $\alpha = \psi \circ \pi$ where $\psi\colon E^{(p)} \to E$ is separable and cyclic of degree d. We can detect orientability by $\mathbb{Z}[\sqrt{-dp}]$ by testing if $\Phi_d(j, j^p) = 0$, and if so, we can compute the orientation by computing $\psi\colon E^{(p)} \to E$ of degree d. We generally expect j^p to be a root of multiplicity one of $\Phi_d(j,Y)$ so the two possible orientations $\mathbb{Z}[\sqrt{-dp}] \hookrightarrow \operatorname{End}(E)$ correspond to ψ and $-\psi$: we have $\bar{\iota} = -\iota = -\psi \circ \pi$.

We present an efficient SmoothAction procedure. We only detail the computation of actions by ideals of small prime norms q, as the computation of smooth actions can be directly derived from it, see [HW25, Proof of Corollary 3] for instance. Suppose q splits in $\mathbb{Z}[\sqrt{-dp}]$, so $q = \mathfrak{q}\bar{\mathfrak{q}} = (q, \sqrt{-dp} - r)(q, \sqrt{-dp} + r)$; here $r \in \{1, 2, \ldots, q-1\}$ is a quadratic residue modulo q, $r^2 \equiv -dp \pmod{q}$. We wish to compute the action of $\mathfrak{q} = (q, \sqrt{-dp} - r)$. The endomorphism α induces an $\mathbb{F}_q$-linear operator on $E[q]$ that is diagonalizable since its characteristic polynomial $x^2 + dp$ splits modulo q. There are two distinct eigenspaces corresponding to the two distinct kernels of oriented isogenies of (E,α). The eigenspace C corresponding to the eigenvalue r is the kernel of the isogeny corresponding to $\mathfrak{q}$, i.e. $E[\mathfrak{q}] = C$.

Thus we may compute $C = E[\mathfrak{q}]$ by doing the following. Suppose E is given by $y^2 = f(x)$ with $f(x) = x^3 + Ax + B \in \mathbb{F}_{p^2}[x]$. Assume q is odd, since the problem is easier

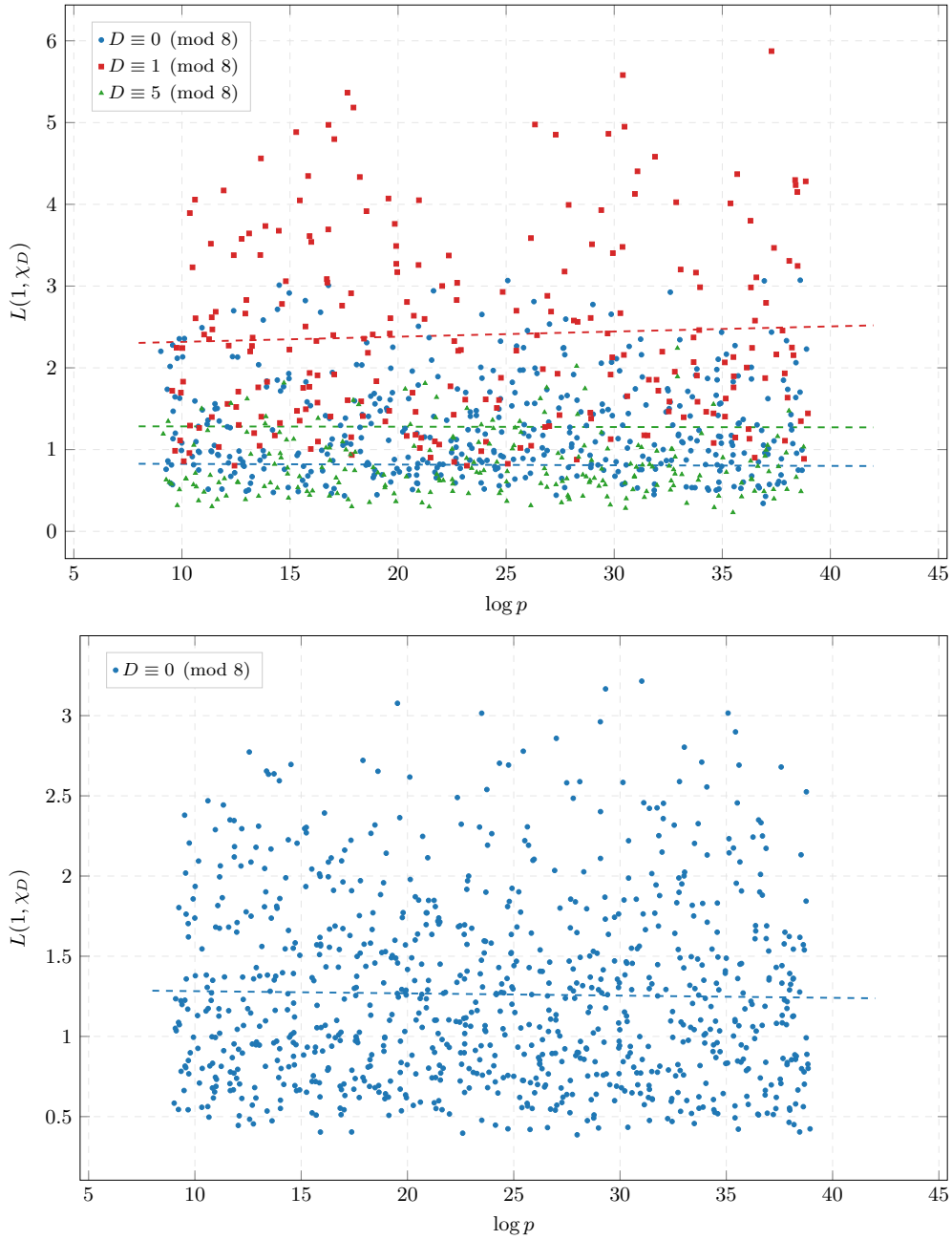


Figure 3: Plots of $L(1, \chi_D)$ for fundamental discriminants $D = -8p$. We plot the best-fit line. The predicted mean is 1.23370. The observed mean is 1.26276.

for $q = 2$. Write $\psi = (x_\psi(X), s^{-1}(x_\psi(X))'Y)$ with $x_\psi \in \mathbb{F}_{p^2}(X)$ and $s \in \mathbb{F}_{p^2}^\times$. Then

$$\begin{aligned}\alpha(X, Y) &= (\psi \circ \pi)(X, Y) \\ &= (x_\psi(X^p), s^{-1}(x_\psi(X^p))'Y^p) \\ &= (x_\psi(X^p), s^{-1}(x_\psi(X^p))' f(X)^{(p-1)/2} Y).\end{aligned}$$

We can compute the (generically) two q-neighbours of E that are d-isogenous to their Galois conjugate as in the work of Chenu and Smith [CS21, pg. 7-8], by computing the

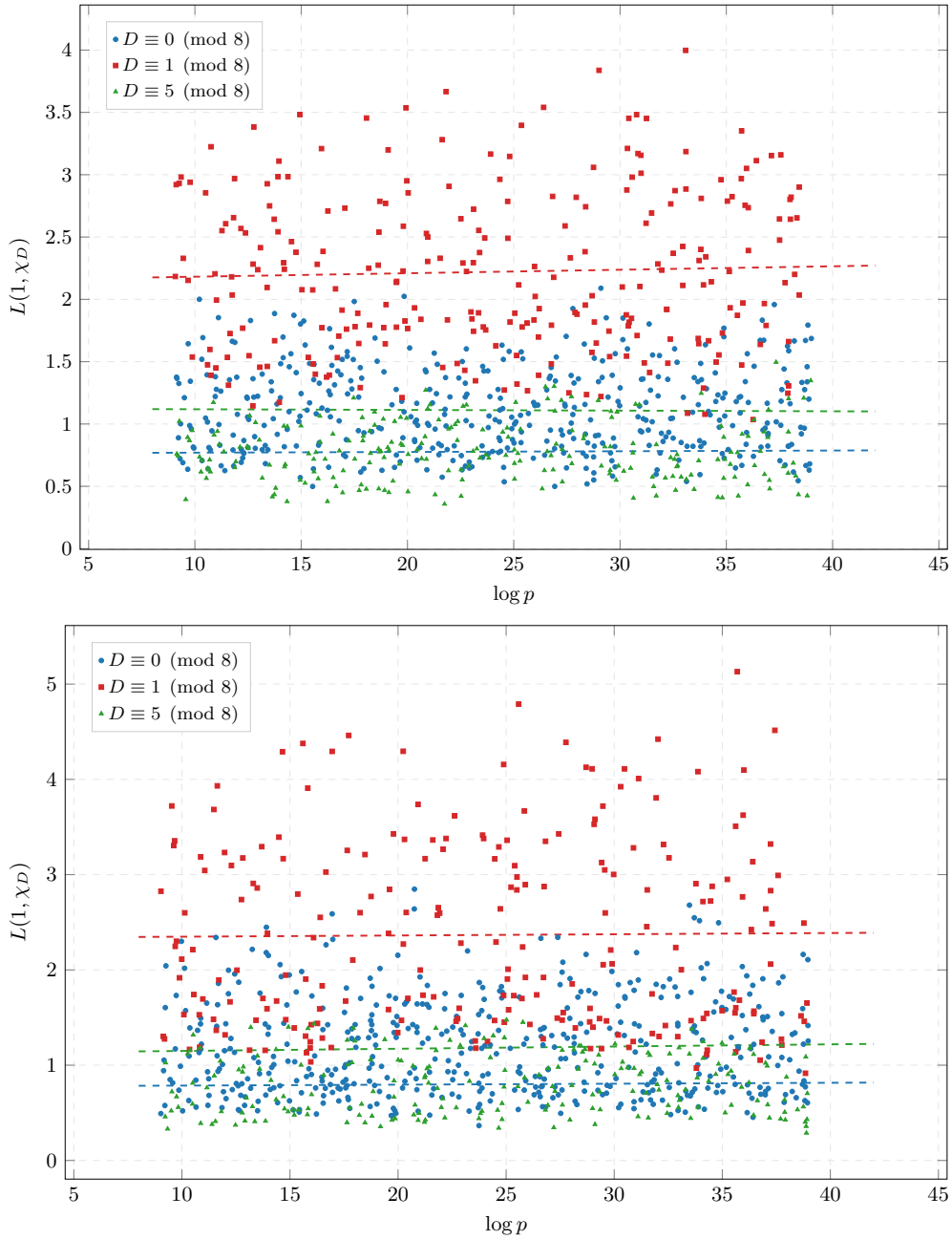


Figure 5: Plots of $L(1, \chi_D)$ for fundamental discriminants $D = -5p$ or $D = -20p$, sorted by congruency class of D mod 8. We plot the best-fit line for each congruency class of D even or $D \equiv 1, 5 \pmod 8$. The predicted means are 1.18435, 2.36870, and 0.78957. The observed means are 1.18177, 2.36657, and 0.80003.

roots of $\gcd(\Phi_q(j, x), \Phi_d(x, x^p))$; this gcd can be efficiently computed by first reducing x^p modulo $\Phi_q(j, x)$. Then we can compute all cyclic subgroups C of E of order d such that E/C is d-isogenous to its Frobenius conjugate. Then, for each cyclic subgroup C of order q in $E[\ell]$ such that E/C is d-isogenous to $E^{(p)}/C^{(p)}$:

1. Compute the kernel polynomial $h(x)$ for C.

2. Compute $a, b \in \mathbb{F}_{p^2}[X]/(h(X))$, the images of $x_\psi(X^p)$ and $s^{-1}(x_\psi(X^p))' f(X)^{(p-1)/2}$.
3. Compute a', b' in $\mathbb{F}_{p^2}[X]/(h(x))$, the images of the x and y coordinates of $[r]\colon E \to E$.
4. if $a = a'$ and $b = b'$ (as elements of $R = \mathbb{F}_{p^2}[X]/(h(X))$), return C and h.

Each iteration of the above algorithm requires about $O(\log r)$ inversions in R and $O(q+\log r)$ multiplications so it is efficient. It is clearly correct since $\alpha \equiv r \pmod{\mathfrak{q}}$ and the kernel polynomial h of $E[\mathfrak{q}]$ cuts out the subgroup scheme $E[\mathfrak{q}]$ of E.

Now we know how to match up the ideal $\mathfrak{q}$ with the subgroup $E[\mathfrak{q}]$; let $E_\mathfrak{q} = E/E[\mathfrak{q}]$ and let $\varphi_\mathfrak{q}$ be the corresponding isogeny, which we assume is normalized. We need to compute the induced orientation $(\varphi_\mathfrak{q})_*\iota$. Chenu and Smith describe how to do this using a generator of the kernel or with isogeny evaluations, but in general one can compute the pushforward of an isogeny ψ by an isogeny ψ' using only the coordinate functions of ψ and ψ': the kernel polynomial of $\psi'_*\psi$ is $\operatorname{Res}_X(H(X), TV(X) - U(X))/\operatorname{Res}_X(H(X), V(X))$, where U/V is the x-coordinate of ψ' and H is the kernel polynomial of ψ.

Another idea is to observe that the curve $E_\mathfrak{q}$ is d-isogenous to $E_\mathfrak{q}^{(p)}$. Typically there is only going to be one subgroup K of order d such that $E_\mathfrak{q}^{(p)}/K = E_\mathfrak{q}$. Let us ignore when this fails as well as the case that $j(E_\mathfrak{q}) = 0, 1728$ – our implementation explicitly avoids this case.

We must decide which of the two isogenies $E_\mathfrak{q}^{(p)} \to E_\mathfrak{q}$ correspond to $(\varphi_\mathfrak{q})_*\iota$. Recall $\psi(X, Y) = (x_\psi(X), s^{-1}(x_\psi(X))'Y)$. Let $\sigma\colon E_\mathfrak{q}^{(p)} \to E_\mathfrak{q}$ be any cyclic d-isogeny. We know that

$$\psi^*\omega_E = s\omega_{E^{(p)}}$$

and

$$(\varphi_\mathfrak{q})_*\alpha = \frac{1}{q}\varphi_\mathfrak{q} \circ \alpha \circ \widehat{\varphi}_\mathfrak{q} = \frac{1}{q}\left(\pi \circ \varphi_\mathfrak{q}^{(p)} \circ \psi^{(p)} \circ \widehat{\varphi}_\mathfrak{q}\right).$$

By our assumption that $\varphi_\mathfrak{q}$ is normalized, we also have that $\varphi_\mathfrak{q}^{(p)}$ is, and so we have $(\varphi_\mathfrak{q}^{(p)})^*\omega_{E_\mathfrak{q}^{(p)}} = \omega_{E_\mathfrak{q}}$ and $(\widehat{\varphi}_\mathfrak{q})^*\omega_E = q\omega_{E_\mathfrak{q}}$. Therefore:

$$\left(\frac{1}{q}\varphi^{(p)} \circ \psi^{(p)} \circ \widehat{\varphi}\right)^* \omega_{E_\mathfrak{q}^{(p)}} = \left(\frac{1}{q} \cdot 1 \cdot s \cdot q\right)\omega_{E_\mathfrak{q}} = s\omega_{E_\mathfrak{q}}.$$

Thus if $\sigma^*\omega_{E_\mathfrak{q}} = s\omega_{E_\mathfrak{q}^{(p)}}$ then $(\varphi)_*\iota(\sqrt{-dp}) = \sigma \circ \pi$; otherwise it is $-\sigma \circ \pi$.

Let us return to the setting of $\sqrt{-\star p}$-WayFinder. We wish to compute an isogeny between two given supersingular elliptic curves. We assume we have computed isogenies $\varphi_i\colon E_i \to E_i'$, where E_i' are oriented by orders R_i, as well as R_i-oriented isogenies $\psi_i\colon E_i' \to E_i^{\mathrm{spe}}$ and the endomorphism rings of $\mathrm{End}(E_i^{\mathrm{spe}})$. We then use the Deuring correspondence to compute an isogeny $\gamma\colon E_1^{(\mathrm{spe})} \to E_2^{(\mathrm{spe})}$; for this we use the implementation of Cai, Chen, and Petit in [CCP25]. Concatenating the isogenies solves the instance of the isogeny problem.

Chenu–Smith orientations. We conclude with some comparisons of our methods with those of Chenu and Smith. Supersingular elliptic curves oriented by quadratic orders of the form $\mathbb{Z}[\sqrt{-dp}]$, for d a square-free integer, were studied in [CS21]. In this article, the authors study the corresponding isogeny graphs and suggest some cryptographic applications. To conduct their analysis, they introduce the notion of a (d, ε)-structure. We say that a pair (E, φ) is a (d, ε)-structure if φ is a d-isogeny from E to its Galois-conjugate curve $E^{(p)}$ such that $\hat{\varphi} = \varepsilon \circ \varphi^{(p)}$ with $\varepsilon \in \{-1, 1\}$, and where $\varphi^{(p)}$ is the isogeny obtained by p-th powering all the coefficients in a rational map defining φ. Among the applications given by Chenu and Smith, one can find a discussion on a possible generalisation of the Delfs–Galbraith algorithm to these more general orientations.

Let us briefly compare our approach with Chenu and Smith's approach to computing an isogeny between two (d, ε)-structures. Using the language of orientations rather than (d, ϵ)-structures, Chenu and Smith insist on finding a *crossroads*, a curve E orientable by both $\mathbb{Z}[\sqrt{-d_1 p}]$ and $\mathbb{Z}[\sqrt{-d_2 p}]$, and then computing an isogeny between (E_1, ι_1) and (E_2, ι_2) where $\iota_i \colon \mathbb{Z}[\sqrt{-d_i p}] \hookrightarrow \mathrm{End}(E_i)$ are orientations, by connecting (E_i, ι_i) to the orientation of E by $\mathbb{Z}[\sqrt{-d_i p}]$ for $i = 1, 2$. Our use of the Deuring correspondence circumvents the need to find (d_1, d_2)-crossroads, which may not exist for a given p.

As already discussed in the previous subsection, another notable difference concerns the SmoothAction procedure. In [CS21, Section 4.4], the authors provide an extensive analysis of the computation of ideal actions with prime norm using Vélu's formulas. This approach can be very costly depending on where the generators of the isogeny kernel live. Using kernel polynomials allows one to avoid this restriction and to works entirely over the base field $\mathbb{F}_{p^2}$: we can compute pushforward orientations without kernel generators.

While Chenu and Smith largely dismiss the application of (d, ε)-structures in solving the general isogeny problem, our cost model lets us accurately model the tradeoff between the cost of detecting orientability and the increased size of the set of orientable curves. We found in Section 7 that the use of more orientations, particularly by orders of the form $\mathbb{Z}[\sqrt{-dp}]$, results in faster algorithms. Finally, our use of the Deuring correspondence gives us an efficient algorithm for obtaining isogenies from integral ideals giving a collision found by the adaptation of Pollard's ρ. Again this circumvents the need for a concrete Clapoti-style algorithm that would, in practice, impose restrictions on the characteristic p.

Remark 10. Experiments tend to show that the existence of crossroads between orientations by $\mathbb{Z}[\sqrt{-\ell p}]$ and $\mathbb{Z}[\sqrt{-\ell' p}]$, for ℓ and ℓ' in $\{1, 2, 3, 5, 7, 13\}$, are not rare. Nevertheless, the issue of getting a low-storage algorithm remains prior to our work.

Computing the endomorphism ring. We observe that WayFinder can be used to directly compute the endomorphism ring of a given curve without resolving an isogeny path problem.

Our detection algorithm returns an isogeny $\varphi \colon E \to E_1$ of degree 2^e for some e where E_1 has an orientation by an order R_1 where R_1 contains a square root of $-dp$ for some $d \geq 1$. Then we have a nontrivial endomorphism $\alpha = \widehat{\varphi} \circ \iota(\sqrt{-dp}) \circ \varphi$ of degree $2^{2e} dp$. Suppose we continue exploring the DFS tree until finding several more isogenies from E to oriented curves $(E_2, \iota_2), (E_3, \iota_3), \ldots, (E_t, \iota_t)$, where $\iota_i \colon R_i \hookrightarrow \mathrm{End}(E_i)$ is an embedding and R_i has a square root of $-d_i p$, resulting in more inseparable endomorphisms $\alpha_i = \widehat{\varphi_i} \circ \iota_i(\sqrt{-d_i p}) \circ \varphi_i$. A modification of the proof of [FIK+25, Proposition 3.13] shows that two such endomorphisms generates a Gorenstein suborder of $\mathrm{End}(E)$. Every endomorphism α_i is in $\mathbb{Z} + P$ where $P = \mathrm{Hom}(E^{(p)}, E)\pi$ is the ideal of inseparable endomorphisms of E. Heuristically (see [FIK+25, Section 6.2]), the expected number t of inseparable endomorphisms required to find a generating set for $\mathbb{Z} + P$ is a small constant, independent of p. The only maximal order containing $\mathbb{Z} + P$ is $\mathrm{End}(E)$ and can be recovered efficiently from a basis of $\mathbb{Z} + P$ [FIK+25, Proposition 3.1, Remark 3.3].

7 Experiments

We conclude by discussing experiments with our SageMath/Python implementation on the first stage of $\sqrt{-\star p}$-WayFinder. In Section 7.1, we show the accuracy of our cost model in predicting the total cost of the algorithm. Through the introduction of our refined estimates for the class number of imaginary quadratic orders in Section 6.1.3, we can exhibit WayFinder's performance for cryptographic-sized primes. This allows our cost model and its implementation to serve as a "hardness estimator" of the isogeny problem for parameters currently used in isogeny-based cryptosystems.

In Appendix D we give extended details on our results. All experiments can be reproduced with our attached implementation.

7.1 Results for small primes

We implemented $\sqrt{-\star p}$-WayFinder in SageMath/Python, building upon the SuperSolver implementation provided in [CRSCS22]. In particular, we follow their depth-first search implementation, and extend the detection of subfield curves by additionally allowing the detection of orientations as described in Section 6. Furthermore, we implemented the precomputation procedure that on input a prime p outputs the best candidates for the optimal set $\mathcal{L}$ as well as the predicted cost and relative improvement over Solver. By restricting to an empty set for $\mathcal{L}$ or to pairs $(\ell_1, 1)$, our implementation is able to run Solver or SuperSolver, respectively, instead of $\sqrt{-\star p}$-WayFinder. All reported results for Solver and SuperSolver have been obtained in this way.

In Table 2 in Appendix D, we report on experiments for the largest primes of bitlengths between 20 and 44. For each of these primes, we generate a number of pseudo-random j-invariants of supersingular curves defined over $\mathbb{F}_{p^2}$, obtained by performing random walks in the 3-isogeny graph starting from E_0. For all of these instances, we run Solver, SuperSolver and $\sqrt{-\star p}$-WayFinder with the five best sets predicted for p by our precomputation. Table 2 shows the predicted and measured average cost in $\mathbb{F}_p$-multiplications of Solver, and for SuperSolver and $\sqrt{-\star p}$-WayFinder the set that performed best among these with its expected cost relative to the expected Solver cost, and the measured average cost relative to the measured average cost of Solver, respectively. We ran 1000 instances per prime of bitlengths 20-29, 500 instances for bitlengths 30-34, 100 instances for bitlengths 35-39, and 50 instances for bitlengths 40-44. For each bitlength we used the same prime as that reported in [CRSCS22, Table 5] to allow for comparison.

The results show that both our predicted cost and the predicted speed-up over Solver are very accurate. The measured speed-up of $\sqrt{-\star p}$-WayFinder over SuperSolver is presented in the last column of Table 2 and heavily depends on the prime, as expected: in some cases we see speed-ups up to 63%, while in some cases SuperSolver sets perform best. However, only for one prime $\sqrt{-\star p}$-WayFinder preferred a set $\mathcal{L}$ that performed slightly worse than the SuperSolver set, which might be due to the low number of instances ran at this bitlength.

Remark 11. As discussed in both Section 3 and Section 6.1, we use a different cost model than [CRSCS22] to precompute our optimal set. It is natural to wonder how our predictions compare when running SuperSolver (i.e., when restricting to pairs of the form $(\ell, 1)$). For $p = 2^{20} - 3$, the model in [CRSCS22] predicts a ratio of 0.17 versus Solver, whereas we predict 0.49. This is far closer to the actual ratio of 0.52. This trend continues for all primes, e.g., for $2^{30} - 35$ they predict a ratio of 0.15, and for $2^{40} - 87$ a ratio of 0.16. In contrast, the actual ratios are 0.61 and 0.39, respectively. We conclude that our cost model more accurately predicts the improvement over Solver.

Pairs (ℓ_1, ℓ_2) with both $\ell_i \neq 1$ do not appear in the optimal set at these small bitsizes. Indeed, the detection cost for these pairs is still relatively high. We remark that there is potential to optimise this greatly through multipoint evaluation, which we leave for future work. Despite this, such pairs will appear for cryptographic-sized primes as the cost of taking a step is larger.

Running similar experiments as those reported in Table 2 with $\mathcal{L}$ including these types of pairs (which use the resultant polynomials for detection), we observe that our predictions remain very accurate. We exhibit this explicitly for pairs with $1 < \ell_i \leq 5$ in Table 3 in Appendix D. With this, we believe our cost metric will accurately predict the cost of WayFinder for cryptographic sized primes (where these pairs are included in $\mathcal{L}$).

7.2 Estimated costs for cryptographic primes

With an eye towards the cryptanalysis of isogeny-based schemes, we now estimate the expected complexity of $\sqrt{-\star p}$-WayFinder for cryptographic primes.

As the choice of optimal sets and thus expected total cost of running $\sqrt{-\star p}$-WayFinder is sensitive to both the size and shape of the prime p, we are not able to "average over all primes" and extrapolate an estimated improvement for cryptographic-sized primes. Instead, we run our cost model on a particular prime p to predict the improvement we expect over Solver. In Section 7.1 we demonstrated that our cost model accurately predicts the cost of $\sqrt{-\star p}$-WayFinder. Thus, combining this with our class number estimates from Section 6.1.3, we should obtain accurate (relative) predictions. Indeed, we observe that using our class number estimate always recovers the set reported in Table 2 in the top five optimal sets predicted.

To this end, in Table 1 we predict the estimated cost of our algorithm for the primes used in the NIST submission of SQIsign [AAA+25], (also used in, for example, PRISM [BBC+25]), and POKÉ [BM25], as these are the main parameter sets used in isogeny-based cryptography. Note that, for a fair comparison, we give the predicted cost of SuperSolver using our cost model. As discussed in Remark 11, this differs from the predictions in [CRSCS22].

From the data in Table 1, we can immediately estimate the concrete complexity of WayFinder for cryptographic primes p. For instance, solving the isogeny problem for the SQIsign Level-I prime requires (on average) $2^{132.36} \cdot 0.1 = 2^{129.04}$ $\mathbb{F}_p$-multiplications, using the set $\{(1,2),(3,1),(1,3),(5,1),(7,1),(1,7)\}$. Furthermore, we see that WayFinder consistently outperforms SuperSolver for parameters used in cryptographic schemes.

Table 1: The estimated cost in $\mathbb{F}_p$-multiplications of running Delfs–Galbraith, SuperSolver and $\sqrt{-\star p}$-WayFinder with the predicted optimal set on primes p used in isogeny-based cryptographic schemes SQIsign and POKÉ.

		Solver	SuperSolver		$\sqrt{-\star p}$-WayFinder		
	Prime p	expected cost	opt. set	expected cost vs. Solver	opt. set	expected cost vs. Solver	expected cost vs. SuperSolver
	$5 \cdot 2^{248} - 1$	$2^{132.36}$	$\{3,5,7,11,13\}$	0.12	$\{(1,2),(3,1),(1,3),(5,1),(7,1),(1,7)\}$	0.10	0.83
[AAA+25]	$65 \cdot 2^{376} - 1$	$2^{198.80}$	$\{3,5,7,11\}$	0.10	$\{(5,2),(3,1),(1,3),(7,1),(1,7),(11,1),(1,11)\}$	0.07	0.70
	$27 \cdot 2^{500} - 1$	$2^{258.92}$	$\{3,5,7,11,13\}$	0.07	$\{(3,2),(1,3),(5,1),(7,1),(1,7),(11,1),(1,11)\}$	0.06	0.86
	$2^{129} \cdot 3^{164} \cdot 5^{18} - 1$	$2^{221.39}$	$\{3,5,7,11,13\}$	0.10	$\{(3,2),(1,3),(5,1),(7,1),(1,7)\}$	0.07	0.70
[BM25]	$2^{192} \cdot 3^{243} \cdot 5^{28} \cdot 7^2 - 1$	$2^{330.46}$	$\{3,5,7,9,11,13\}$	0.08	$\{(3,2),(5,2),(1,3),(7,1),(1,7),(11,1),(1,11)\}$	0.06	0.75
	$2^{256} \cdot 3^{324} \cdot 5^{36} \cdot 547 - 1$	$2^{438.07}$	$\{3,5,7,9,11,13\}$	0.06	$\{(3,2),(5,2),(1,3),(7,1),(1,7),(11,1),(1,11)\}$	0.05	0.83

References

[AAA+25] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2025. URL: https://sqisign.org.

[ACL+24] Sarah Arpin, Mingjie Chen, Kristin E. Lauter, Renate Scheidler, Katherine E. Stange, and Ha T. N. Tran. *Orientations and Cycles in Supersingular Isogeny Graphs*, pages 25–86. Springer International Publishing, Cham, 2024. doi:10.1007/978-3-031-51677-1_2.

[ACNL+23] Sarah Arpin, Catalina Camacho-Navarro, Kristin Lauter, Joelle Lim, Kristina Nelson, Travis Scholl, and Jana Sotáková. Adventures in supersingularland. *Exp. Math.*, 32(2):241–268, 2023. URL: https://doi-org.ezproxy.lib.vt.edu/10.1080/10586458.2021.1926009, doi:10.1080/10586458.2021.1926009.

[BBC+25] Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. Prism: Simple and compact identification and signatures from large prime degree isogenies. In Tibor Jager and Jiaxin Pan, editors, *Public-Key Cryptography – PKC 2025*, pages 300–332, Cham, 2025. Springer Nature Switzerland.

[BCC+23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part II*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30617-4_14.

[BLS12] Reinier Bröker, Kristin Lauter, and Andrew V. Sutherland. Modular polynomials via isogeny volcanoes. *Math. Comp.*, 81(278):1201–1231, 2012. URL: https://doi-org.ezproxy.lib.vt.edu/10.1090/S0025-5718-2011-02508-1, doi:10.1090/S0025-5718-2011-02508-1.

[BM25] Andrea Basso and Luciano Maino. POKÉ: A compact and efficient PKE from higher-dimensional isogenies. In *Advances in Cryptology – EUROCRYPT 2025, Part II*, Lecture Notes in Computer Science, pages 94–123. Springer, Cham, Switzerland, June 2025. doi:10.1007/978-3-031-91124-8_4.

[BS12] Gaetan Bisson and Andrew V. Sutherland. A low-memory algorithm for finding short product representations in finite groups. *Des. Codes Cryptogr.*, 63(1):1–13, 2012. URL: https://doi-org.ezproxy.lib.vt.edu/10.1007/s10623-011-9527-8, doi:10.1007/s10623-011-9527-8.

[CCP25] Shiping Cai, Mingjie Chen, and Christophe Petit. Faster algorithms for isogeny computations over extensions of finite fields. In Andrzej Dąbrowski, Josef Pieprzyk, and Jacek Pomykała, editors, *Number-Theoretic Methods in Cryptology*, pages 136–155, Cham, 2025. Springer Nature Switzerland.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_15.

[CD25] Jesús-Javier Chi-Domínguez. Improved subfield curve search for specific field characteristics. Cryptology ePrint Archive, Paper 2025/226, 2025. URL: https://eprint.iacr.org/2025/226.

[CEMR24] Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders. AprèsSQI: Extra fast verification for SQIsign using extension-field signing. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 63–93, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58716-0_3.

[CK20] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *J. Math. Cryptol.*, 14(1):414–437, 2020. URL: https://doi-org.ezproxy.lib.vt.edu/10.1515/jmc-2019-0034, doi:10.1515/jmc-2019-0034.

[Cox22] David A. Cox. *Primes of the form $x^2 + ny^2$—Fermat, class field theory, and complex multiplication*. AMS Chelsea Publishing, Providence, RI, third edition, 2022. With contributions by Roger Lipsett.

[CRSCS22] Maria Corte-Real Santos, Craig Costello, and Jia Shi. Accelerating the Delfs-Galbraith algorithm with fast subfield root detection. In *Advances in cryptology—CRYPTO 2022. Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 285–314. Springer, Cham, 2022. URL: https://doi.org/10.1007/978-3-031-15982-4_10, doi:10.1007/978-3-031-15982-4_10.

[CS21] Mathilde Chenu and Benjamin Smith. Higher-degree supersingular group actions. *Mathematical Cryptology*, 1(2):85–101, 2021.

[Dar25] Pierrick Dartois. Fast computation of 2-isogenies in dimension 4 and cryptographic applications. *Journal of Algebra*, 683:449–514, 2025. URL: https://www.sciencedirect.com/science/article/pii/S0021869325003771, doi:10.1016/j.jalgebra.2025.06.033.

[Dav00] Harold Davenport. *Multiplicative number theory*, volume 74 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 2000. Revised and with a preface by Hugh L. Montgomery.

[DG16] Christina Delfs and Steven D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016. doi:10.1007/s10623-014-0010-1.

[DR73] P. Deligne and M. Rapoport. Les schémas de modules de courbes elliptiques. In *Modular functions of one variable, II (Proc. Internat. Summer School, Univ. Antwerp, Antwerp, 1972)*, volume Vol. 349 of *Lecture Notes in Math.*, pages 143–316. Springer, Berlin-New York, 1973.

[DS05] Fred Diamond and Jerry Shurman. *A first course in modular forms*, volume 228 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005.

[Elk98] Noam D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational perspectives on number theory (Chicago, IL, 1995)*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 21–76. Amer. Math. Soc., Providence, RI, 1998. URL: https://doi-org.ezproxy.lib.vt.edu/10.1090/amsip/007/03, doi:10.1090/amsip/007/03.

[FIK+25] Jenny Fuselier, Annamaria Iezzi, Mark Kozek, Travis Morrison, and Changningphaabi Namoijam. Computing supersingular endomorphism rings using inseparable endomorphisms. *J. Algebra*, 668:145–189, 2025. URL: https://doi-org.ezproxy.lib.vt.edu/10.1016/j.jalgebra.2025.01.012, doi:10.1016/j.jalgebra.2025.01.012.

[FM02] Mireille Fouquet and François Morain. Isogeny volcanoes and the SEA algorithm. In *Proceedings of the 5th International Symposium on Algorithmic Number Theory*, ANTS-V, page 276–291, Berlin, Heidelberg, 2002. Springer-Verlag.

[Fou01] Mireille Fouquet. *Anneau d'endomorphismes et cardinalité des courbes elliptiques : aspects algorithmiques*. PhD thesis, École polytechnique, 2001.

[GHS02] Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the GHS Weil descent attack. In *Advances in cryptology—EUROCRYPT 2002 (Amsterdam)*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 29–44. Springer, Berlin, 2002. URL: https://doi-org.ezproxy.lib.vt.edu/10.1007/3-540-46035-7_3, doi:10.1007/3-540-46035-7_3.

[GS03] A. Granville and K. Soundararajan. The distribution of values of $L(1, \chi_d)$. *Geom. Funct. Anal.*, 13(5):992–1028, 2003. URL: https://doi-org.ezproxy.lib.vt.edu/10.1007/s00039-003-0438-3, doi:10.1007/s00039-003-0438-3.

[GS13] Steven Galbraith and Anton Stolbunov. Improved algorithm for the isogeny problem for ordinary elliptic curves. *Appl. Algebra Engrg. Comm. Comput.*, 24(2):107–131, 2013. URL: https://doi-org.ezproxy.lib.vt.edu/10.1007/s00200-013-0185-0, doi:10.1007/s00200-013-0185-0.

[Hal05] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474. ACM, New York, 2005. URL: https://doi-org.ezproxy.lib.vt.edu/10.1145/1060590.1060660, doi:10.1145/1060590.1060660.

[HAS26] Taha Hedayat, Sarah Arpin, and Renate Scheidler. The spine of a supersingular ℓ-isogeny graph. In *LuCaNT: databases, algorithms, and computational number theory*, volume 840 of *Contemp. Math.*, pages 75–101. Amer. Math. Soc., [Providence], RI, [2026] ©2026. URL: https://doi-org.ezproxy.lib.vt.edu/10.1090/conm/840/16808, doi:10.1090/conm/840/16808.

[HM89] James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2(4):837–850, 1989. URL: https://doi-org.ezproxy.lib.vt.edu/10.2307/1990896, doi:10.2307/1990896.

[HW25] Arthur Herlédan Le Merdy and Benjamin Wesolowski. The supersingular endomorphism ring problem given one endomorphism. *IACR Communications in Cryptology*, 2(1), 2025. doi:10.62056/akgyivrzn.

[Kan89] Masanobu Kaneko. Supersingular j-invariants as singular moduli mod p. *Osaka J. Math.*, 26(4):849–855, 1989. URL: http://projecteuclid.org.ezproxy.lib.vt.edu/euclid.ojm/1200781857.

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

[KM85] Nicholas M. Katz and Barry Mazur. *Arithmetic moduli of elliptic curves*, volume 108 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 1985. URL: https://doi-org.ezproxy.lib.vt.edu/10.1515/9781400881710, doi:10.1515/9781400881710.

[Koh96] David R. Kohel. *Endomorphism rings of elliptic curves over finite fields*. ProQuest LLC, Ann Arbor, MI, 1996. Thesis (Ph.D.)–University of California, Berkeley.

[Lit28] J. E. Littlewood. On the Class-Number of the Corpus $P(\sqrt{-k})$. *Proc. London Math. Soc. (2)*, 27(5):358–372, 1928. URL: https://doi-org.ezproxy.lib.vt.edu/10.1112/plms/s2-27.1.358, doi:10.1112/plms/s2-27.1.358.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_16.

[MW25] Arthur Herlédan Le Merdy and Benjamin Wesolowski. Unconditional foundations for supersingular isogeny-based cryptography. In *TCC 2025: 23rd Theory of Cryptography Conference, Part III*, Lecture Notes in Computer Science, pages 266–297. Springer, Cham, Switzerland, November 2025. doi:10.1007/978-3-032-12296-4_9.

[Onu21] Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields and Their Applications*, 69:101777, 2021. URL: https://www.sciencedirect.com/science/article/pii/S1071579720301465, doi:10.1016/j.ffa.2020.101777.

[PR23a] Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Paper 2023/1766, 2023. URL: https://eprint.iacr.org/2023/1766.

[PR23b] Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766, 2023. URL: https://eprint.iacr.org/2023/1766.

[PRSU26] Lorenz Panny, Ryan Rueger, Alessandro Sferlazza, and Aleksei Udovenko. HyperSolver: Asymptotically and concretely accelerating the delfs–galbraith attack using isogeny ladders. Cryptology ePrint Archive, Paper 2026/1823, 2026. URL: https://eprint.iacr.org/2026/1823.

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_17.

[S+23] W. A. Stein et al. *Sage Mathematics Software (Version 10.8)*. The Sage Development Team, 2023. http://www.sagemath.org.

[Sil09] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Graduate texts in mathematics. Springer, Dordrecht, 2009. doi:10.1007/978-0-387-09494-6.

[Sut13] Andrew Sutherland. Isogeny volcanoes. *The Open Book Series*, 1(1):507–530, November 2013. URL: http://dx.doi.org/10.2140/obs.2013.1.507, doi:10.2140/obs.2013.1.507.

[Wes22a] Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes*

in Computer Science, pages 345–371, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07082-2_13.

[Wes22b] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd Annual Symposium on Foundations of Computer Science*, pages 1100–1111, Denver, CO, USA, February 7–10, 2022. IEEE Computer Society Press. doi:10.1109/FOCS52979.2021.00109.

[Wes26] Benjamin Wesolowski. The supersingular isogeny problem in time and memory $p^{1/3+o(1)}$. Cryptology ePrint Archive, Paper 2026/1486, 2026. URL: https://eprint.iacr.org/2026/1486.

A Validating the clustering impact on DFS complexity

To demonstrate the impact of clustering on the complexity of depth-first search (DFS) for subfield curves, and verify the clustering constants above, we ran the following experiment. For each bit length $10 \leq b \leq 39$, we collected 20 random primes. For each prime p, we selected 20 psuedorandom supersingular j-invariants j and ran 20 iterations of depth-first search from j for a subfield curve, keeping track of the total number of nodes visited. For each prime p we also computed $\#V(\mathbb{F}_p) = H(-4p)/2$, the number of $\mathbb{F}_p$-rational supersingular j-invariants. In Figures 6 and 7, we plot the ratio of the average number of nodes visited across the DFS trials to $\#V/\#V(\mathbb{F}_p)$, the expected number of random walks before terminating in $V(\mathbb{F}_p)$. We display these ratios as well as the normalized ratios $\#V/(c_2(\mathbb{Z}[\sqrt{-p}])\#V(\mathbb{F}_p))$.

B Validating estimates of $\mathbb{E}[L(1, \chi_D)]$

To validate our estimates for $\mathbb{E}[L(1, \chi_D)]$, we did the following: for each $\ell \in \{1, 2, 3, 5\}$ and for bit lengths $10 \leq b < 40$, we generated 30 random primes of bit length b. For each such prime we compute $h(\mathbb{Q}(\sqrt{-\ell p}))$ and sort these into classes according to whether the fundamental discriminant $\mathrm{disc}(\mathbb{Q}(\sqrt{-\ell p})$ is even or 1 or 5 modulo 8, i.e., whether 2 is ramified, split, or inert. We compute the sample mean of $h(\mathbb{Q}(\sqrt{-\ell p}))$ in each class. The plots of the data, as well as estimates of the predicted and observed means in each class, are shown in Figures 8 to 11.

C Algorithms

In Algorithm 4 we give the full description of OrientationFinder.

Algorithm 4 OrientationFinder

Input: The j-invariant $j \in \mathbb{F}_{p^2}$ of a supersingular curve E, and a pair of integers (ℓ_1, ℓ_2) with ℓ_1 an odd integer and $\ell_2 \in \{2, 3, 5, 7, 13\}$.

Output: A boolean `bool` indicating whether E is oriented by $\mathbb{Z}[\sqrt{\alpha}]$ where $\alpha \in D \cup \ell_1 D$ with $D = \{\sqrt{-p}, \frac{1+\sqrt{-p}}{2}, \sqrt{-\ell_2 p}, \frac{1+\sqrt{-\ell_2 p}}{2}\}$.

1: Fetch the resultant polynomial $R_{\ell_1,\ell_2,p}(X, Y)$
2: $f \leftarrow R_{\ell_1,\ell_2,p}(j, Y)$
3: **return** RationalFactor(f)

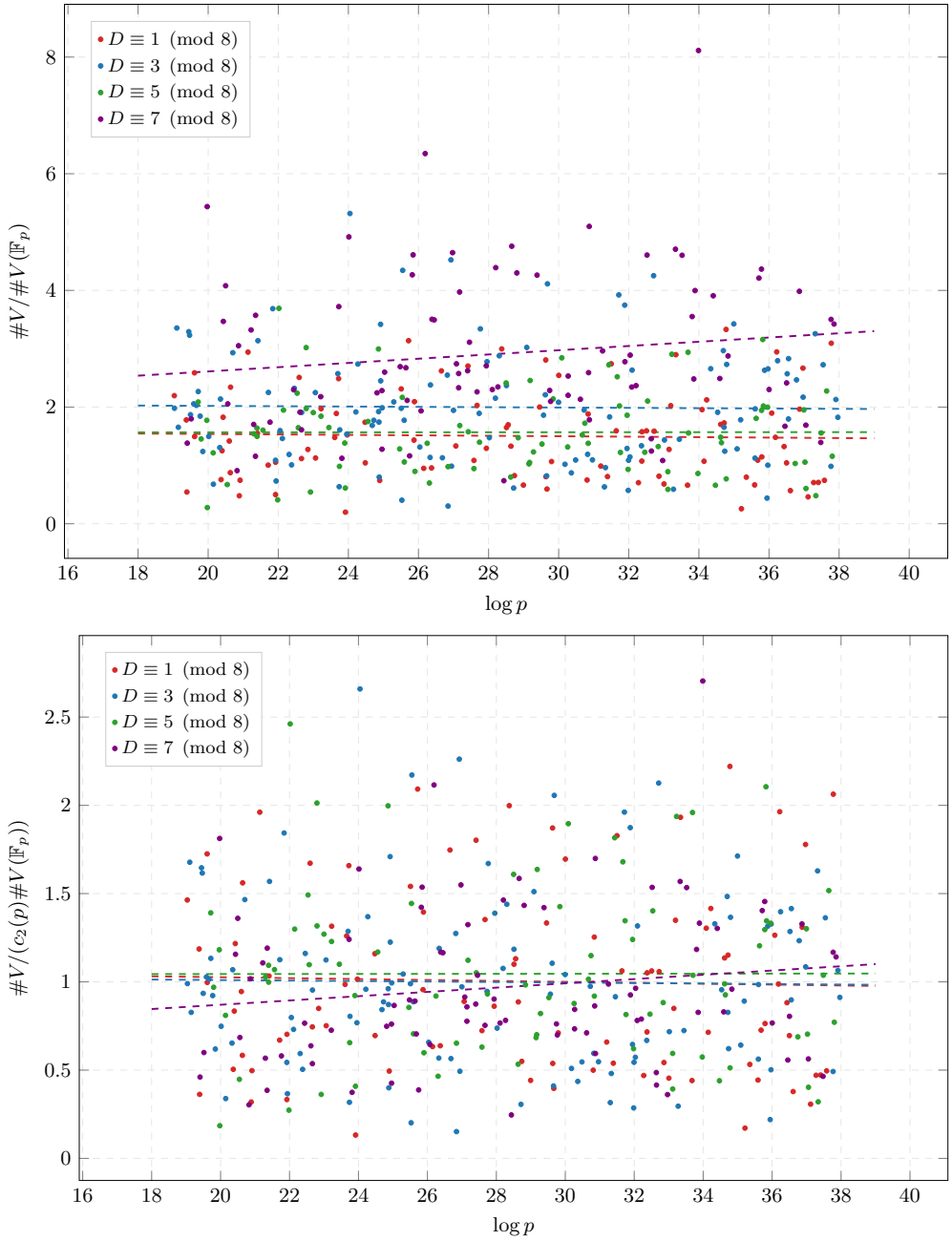


Figure 7: Ratios of average number of DFS trials by $\#V/(c_2(p)\#V(\mathbb{F}_p))$, sorted by congruency class mod 8. We plot the best-fit line for each congruency class. The slopes are $-0.002603, -0.001407, 0.000151$, and 0.012118. The means are $1.0034, 0.9987, 1.0456$, and 0.9722.

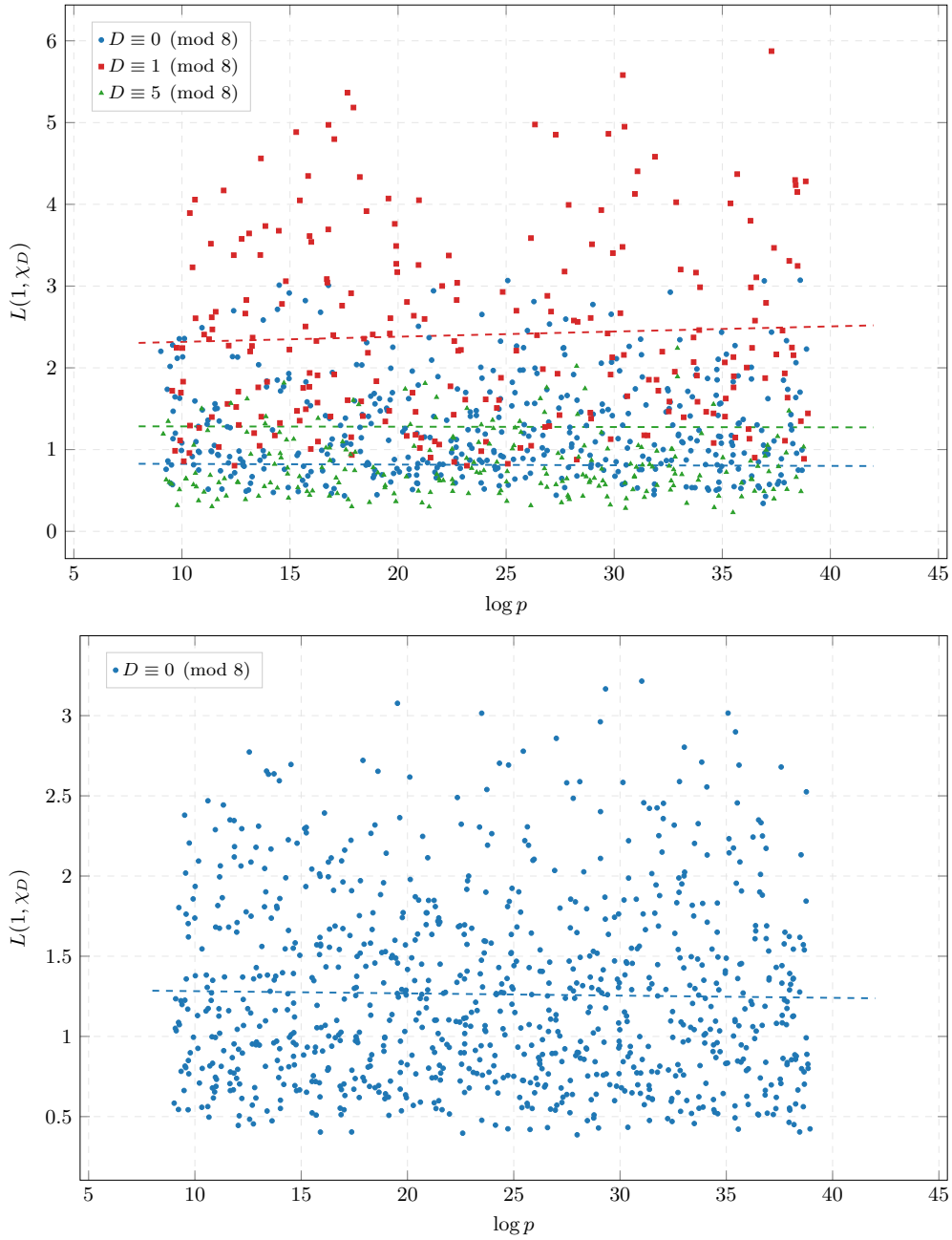


Figure 9: Plots of $L(1, \chi_D)$ for fundamental discriminants $D = -8p$. We plot the best-fit line. The predicted mean is 1.23370. The observed mean is 1.26276.

D Experiment Results

In Tables 2 and 3, we give detailed results of the experiments described in Section 7.

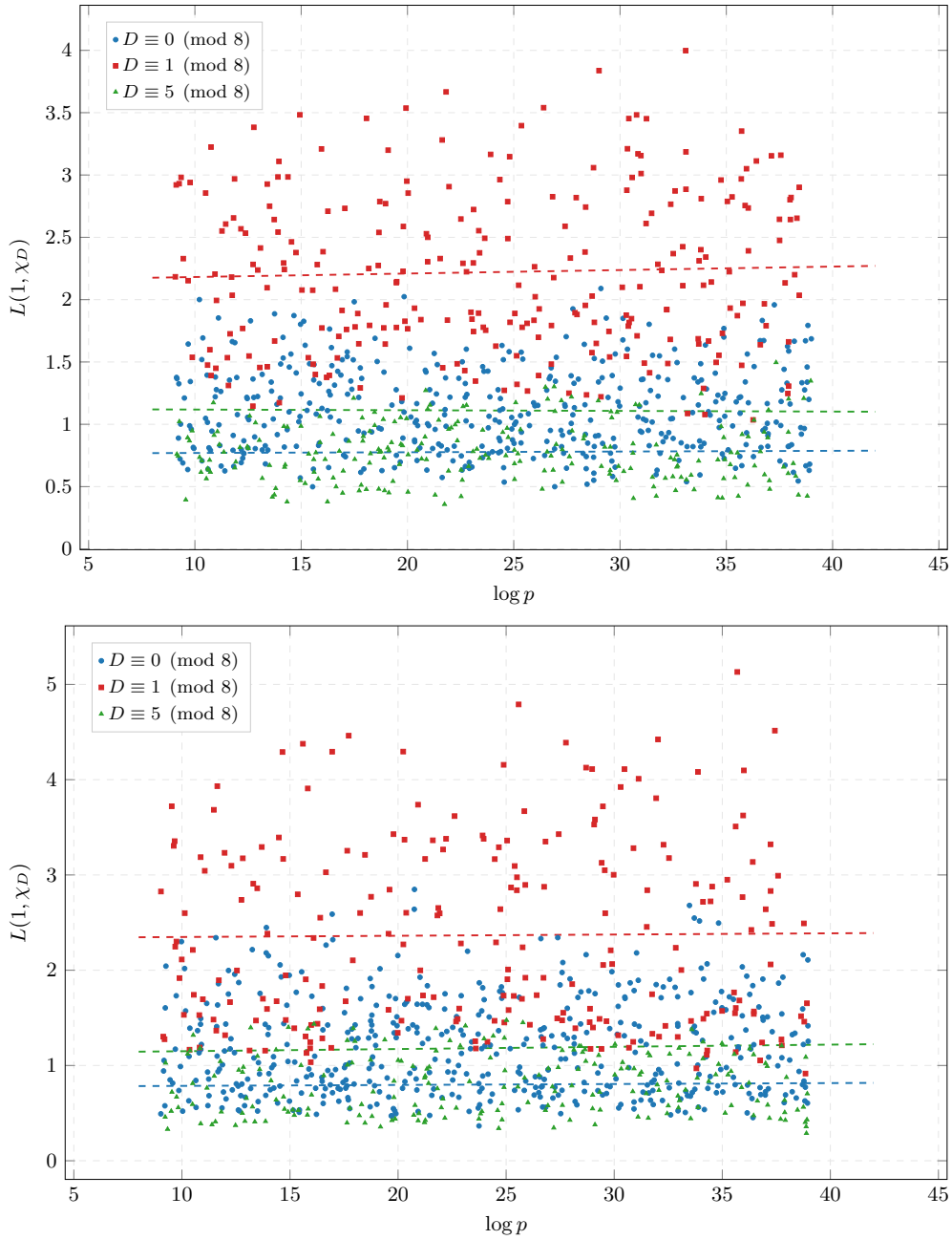


Figure 11: Plots of $L(1, \chi_D)$ for fundamental discriminants $D = -5p$ or $D = -20p$, sorted by congruency class of D mod 8. We plot the best-fit line for each congruency class of D even or $D \equiv 1, 5 \pmod 8$. The predicted means are 1.18435, 2.36870, and 0.78957. The observed means are 1.18177, 2.36657, and 0.80003.

Table 2: Predicted and average cost in $\mathbb{F}_p$-multiplications of running Solver, SuperSolver, and $\sqrt{-\star p}$-WayFinder for several small primes. Further explanation in text.

	Solver		SuperSolver			$\sqrt{-\star p}$-WayFinder			
Prime p	expected cost	average cost	opt. set	expected cost vs. Solver	average cost vs. Solver	optimal set	expected cost vs. Solver	average cost vs. Solver	average cost vs. SuperSolver
$2^{20}-3$	40910	41350	$\{3,5\}$	0.49	0.52	$\{(1,2),(1,3),(1,5)\}$	0.22	0.25	0.48
$2^{21}-9$	22683	22175	$\{3,5\}$	0.63	0.61	$\{(1,2),(3,1),(5,1)\}$	0.60	0.58	0.95
$2^{22}-3$	49015	49484	$\{3\}$	0.47	0.53	$\{(1,2),(3,1),(1,5)\}$	0.39	0.41	0.77
$2^{23}-15$	56361	56488	$\{3,5\}$	0.50	0.53	$\{(1,2),(1,3),(5,1),(1,7)\}$	0.44	0.45	0.85
$2^{24}-3$	112040	105689	$\{3,5\}$	0.44	0.49	$\{(1,2),(3,1),(1,3))\}$	0.28	0.33	0.67
$2^{25}-39$	115380	118128	$\{3,5\}$	0.51	0.57	$\{(1,2),(5,1),(1,7)\}$	0.44	0.46	0.81
$2^{26}-5$	82650	84283	$\{3,5,7\}$	0.75	0.70	$\{(3,1),(5,1),(7,1)\}$	0.75	0.70	1.00
$2^{27}-39$	227147	220768	$\{3,5,7\}$	0.58	0.61	$\{(1,2),(3,1),(5,1),(7,1)\}$	0.57	0.60	0.98
$2^{28}-57$	538116	532990	$\{3\}$	0.45	0.52	$\{(1,2),(3,1),(1,3),(1,5)\}$	0.30	0.33	0.63
$2^{29}-3$	287869	286619	$\{3,7\}$	0.57	0.56	$\{(3,1),(7,1)\}$	0.57	0.56	1.00
$2^{30}-35$	404396	387930	$\{3,5,7\}$	0.55	0.61	$\{(3,1),(1,3),(5,1),(7,1)\}$	0.55	0.59	0.97
$2^{31}-1$	1824253	1834066	$\{3,5\}$	0.4	0.47	$\{(1,2),(3,1),(1,3),(1,5)\}$	0.26	0.31	0.66
$2^{32}-5$	694130	709819	$\{3,5,7\}$	0.64	0.60	$\{(3,1),(5,1),(7,1)\}$	0.64	0.60	1.00
$2^{33}-9$	1393836	1431637	$\{3,5\}$	0.47	0.50	$\{(1,2),(3,1),(5,1)\}$	0.47	0.49	0.98
$2^{34}-41$	2210271	2427901	$\{3,5,7\}$	0.54	0.47	$\{(1,2),(3,1),(1,3),(5,1)\}$	0.46	0.46	0.98
$2^{35}-31$	15695659	14693746	$\{3,5,7\}$	0.34	0.40	$\{(1,2),(1,3),(1,5)\}$	0.12	0.15	0.38
$2^{36}-5$	4472064	4574455	$\{3,5\}$	0.56	0.60	$\{(3,1),(1,3),(5,1)\}$	0.56	0.60	1.00
$2^{37}-25$	16456723	15377330	$\{3,5\}$	0.36	0.38	$\{(1,2),(3,1),(1,3),(1,5)\}$	0.24	0.29	0.76
$2^{38}-45$	15240647	14900375	$\{3,7\}$	0.43	0.41	$\{(1,2),(3,1),(1,3))\}$	0.33	0.32	0.78
$2^{39}-7$	23469050	21952533	$\{3,7\}$	0.37	0.38	$\{(1,2),(3,1),(1,3)\}$	0.30	0.35	0.92
$2^{40}-87$	35799311	36787801	$\{3,5\}$	0.39	0.39	$\{(1,2),(3,1),(1,3),(1,5),(5,1)\}$	0.33	0.28	0.72
$2^{41}-21$	33862347	36026564	$\{5,7\}$	0.48	0.40	$\{(1,2),(3,1),(5,1),(7,1)\}$	0.46	0.41	1.02
$2^{42}-11$	74783295	68551770	$\{3,5\}$	0.39	0.39	$\{(1,2),(3,1),(1,3),(5,1),(1,7)\}$	0.31	0.30	0.77
$2^{43}-57$	51284346	51162604	$\{5,7\}$	0.46	0.40	$\{(5,1),(7,1)\}$	0.46	0.40	1.00
$2^{44}-17$	82201280	73917189	$\{3,5,7\}$	0.44	0.44	$\{(1,2),(3,1),(5,1),(7,1)\}$	0.44	0.41	0.93

Table 3: For primes p with $\log(p) \in \{20, 25, 30, 35, 40\}$, this table reports the ratio between the predicted cost of running Solver and $\sqrt{-\star p}$-WayFinder with $\mathcal{L} = \{(\ell_1, \ell_2)\}$ for $1 < \ell_i \leq 5$ versus the actual average cost. For each bitlength, we average the cost over 20 randomly generated primes and n instances per prime, where n is as for Table 2. Namely, $n = 1000$ for $\log(p) = 20, 25$, $n = 500$ for $\log(p) = 30$, $n = 100$ for $\log(p) = 35$ and $n = 50$ for $\log(p) = 40$. The closer to 1 the entry is, the more accurate our prediction.

		Expected vs. Average Cost for $\log(p)$				
	Set $\mathcal{L}$	**20**	**25**	**30**	**35**	**40**
Solver	$\{\,\}$	0.99	0.99	1.01	0.91	0.93
$\sqrt{-\star p}$-WayFinder	$\{(3,2)\}$	0.90	1.00	0.99	1.03	0.89
	$\{(3,3)\}$	0.88	0.92	1.00	0.85	0.85
	$\{(3,5)\}$	0.88	0.95	1.01	0.92	1.05
	$\{(5,2)\}$	0.96	0.96	0.95	0.95	0.91
	$\{(5,3)\}$	0.87	0.97	0.97	0.97	0.94
	$\{(5,5)\}$	0.95	1.00	0.98	1.00	0.83