Complex-Multiplication Terminals for Supersingular Isogeny Path-Finding

Zheng Tao[1], Zhi Hu[2⋆], Yijing Zhang[2], and Changan Zhao[3,4]

[1] School of Computer Science and Technology, Changsha University of Science and Technology, Changsha 410076, China
zhengt@csust.edu.cn

[2] School of Mathematics and Statistics, Central South University, Changsha 410083, China
huzhi_math@csu.edu.cn
holmeows2024@outlook.com

[3] School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China
zhaochan3@mail.sysu.edu.cn

[4] Guangdong Key Laboratory of Information Security, Guangzhou 510006, China

Abstract. We propose a complementary stopping strategy based on complex multiplication (CM) for the subfield-search stage of supersingular isogeny path-finding, a bottleneck in the Delfs–Galbraith/SuperSolver algorithm. The original search performs a non-backtracking walk in the supersingular 2-isogeny graph until it reaches the subfield terminal set S_p. Our idea is to enlarge the set of recognizable terminals, by adding a precomputed set of CM supersingular vertices. For a discriminant bound M, we construct $S_{\mathrm{CM}}(M)$ from roots of Hilbert class polynomials $H_D(X)$ over $\mathbb{F}_{p^2}$, where D ranges over inert negative fundamental discriminants with $|D| < M$. The walk then stops upon reaching $S_p \cup S_{\mathrm{CM}}(M)$. The only additional per-visit cost is an expected $O(1)$ hash-table membership query in the precomputed CM terminal table. We estimate the size and preprocessing cost of $S_{\mathrm{CM}}(M)$, obtaining the heuristic growth $|S_{\mathrm{CM}}(M)| = \Theta(M^{3/2})$, and show that the overlap $S_{\mathrm{CM}}(M) \cap S_p$ is lower order. We also give terminal-connection procedures showing how searches that stop at CM vertices can be converted into full isogeny paths under the standard quaternionic and KLPT heuristics. The method does not change the asymptotic exponent of the underlying Delfs–Galbraith search; instead, it provides a complementary technique for existing subfield-search methods by adding efficiently recognizable CM terminals. Experiments on small parameters show that enlarging the terminal union reduces both visited vertices and field multiplications, while lookup benchmarks at SQIsign parameters confirm that the additional table membership test has stable and moderate per-visit overhead.

Keywords: Isogeny-based cryptography · supersingular isogeny problem · complex multiplication

⋆ Corresponding author.

1 Introduction

Given two supersingular j-invariants over $\mathbb{F}_{p^2}$, the supersingular isogeny path problem asks for an explicit isogeny path connecting the corresponding supersingular elliptic curves. This is a central algorithmic task in isogeny-based cryptography, underlying constructions and assumptions such as CGL-type hash functions and SQIsign-type signatures [7,14]. For a small prime ℓ, the problem can be viewed as finding paths in the supersingular ℓ-isogeny graph G_ℓ, whose vertices are supersingular j-invariants over $\mathbb{F}_{p^2}$ and whose edges correspond to cyclic ℓ-isogenies [24,27].

The Delfs–Galbraith algorithm and SuperSolver reduce the general path-finding problem to a subfield search [15,11]. Starting from a supersingular vertex in S_{p^2}, where S_{p^2} denotes the set of supersingular j-invariants over $\mathbb{F}_{p^2}$, the search walks in the 2-isogeny graph until it reaches the subfield subset $S_p = S_{p^2} \cap \mathbb{F}_p$. Since $|S_{p^2}| = \Theta(p)$ whereas $|S_p| = \Theta(p^{1/2})$, the subfield target is sparse [29,15]. Under the standard rapid-mixing heuristic, the expected number of visited vertices before hitting S_p is therefore of order $\frac{|S_{p^2}|}{|S_p|} = \Theta(p^{1/2})$, making this subfield search the main bottleneck in the Delfs–Galbraith framework.

SuperSolver improves the concrete cost of this bottleneck by optimized terminal detection: besides testing whether the current vertex lies in S_p, it also detects whether small-degree neighbours lie in S_p [11]. This searches a larger local neighbourhood per unit of work, but the recognizable terminal structure remains tied to the subfield.

This paper develops a complementary approach. Rather than modifying the underlying 2-isogeny walk, we enlarge the set of efficiently recognizable terminal vertices. The additional terminals come from complex multiplication (CM). For a parameter M, our analyzed core set uses negative fundamental discriminants D with $|D| < M$, and lets $H_D(X)$ be the corresponding Hilbert class polynomial. The roots of H_D parametrize complex elliptic curves with CM by $\mathcal{O}_D$, and $\deg H_D = h(D)$ is the corresponding class number [13]. By Deuring reduction, when p is inert in $\mathbb{Q}(\sqrt{D})$, the roots of $H_D(X)$ over $\mathbb{F}_{p^2}$ give supersingular j-invariants [16,21]. We precompute these vertices as an additional terminal set, denoted by $S_{\mathrm{CM}}(M)$ and store them in a precomputed CM terminal table. This choice is motivated by the work of Love and Boneh on supersingular curves with small noninteger endomorphisms [23]. Their results show that CM vertices associated with small discriminants carry useful structural information: same-discriminant vertices admit controlled connections, while vertices from distinct discriminants satisfy nontrivial separation bounds. We use this structure as a source of additional recognizable terminals in the subfield-search stage. The search phase then uses the enlarged terminal set $S_p \cup S_{\mathrm{CM}}(M)$. At each visited vertex, the algorithm keeps the SuperSolver subfield tests and adds one membership query in the precomputed CM terminal table. Thus the walk in G_2 is unchanged; only the stopping condition is enlarged.

To turn this enlarged stopping rule into a complete path-finding method, several issues must be addressed. One must estimate the size of $S_{\mathrm{CM}}(M)$ and

its overlap with S_p, account for the preprocessing and lookup costs of the CM table, and connect a full isogeny path after the search stops at a CM vertex. Our contributions address these points.

Our Contributions. The main contributions of this paper are summarized as follows:

1. We introduce a CM terminal set $S_{\mathrm{CM}}(M)$ obtained from roots of Hilbert class polynomials $H_D(X) \bmod p$, where D ranges over inert negative fundamental discriminants with $|D| < M$. The search phase uses the enlarged terminal set $S_p \cup S_{\mathrm{CM}}(M)$, while the underlying non-backtracking walk in G_2 is unchanged. This gives a complementary stopping mechanism to the subfield terminals used in Delfs–Galbraith and SuperSolver.
2. We estimate the supply of CM terminal vertices using class-number sums. Under the inertness-frequency heuristic (H1), stated and motivated in Section 3.2, the raw CM supply grows as $R_{\mathrm{CM}}(M) = \Theta(M^{3/2})$. We also bound the subfield overlap $S_{\mathrm{CM}}(M) \cap S_p$, showing that its contribution is lower order, namely $\widetilde{O}(M)$. We further account for the preprocessing cost of constructing $S_{\mathrm{CM}}(M)$ and the lookup cost of the CM terminal table, and model each membership query as an expected $O(1)$ operation.
3. We give post-search procedures for connecting terminal vertices. The Delfs–Galbraith subfield connection applies when both lie in S_p. For CM terminals, we distinguish same-discriminant CM–CM, different-discriminant CM–CM, and mixed CM–subfield cases. Same-discriminant terminals are connected by a Galbraith-style bidirectional collision search, in which small split-prime-degree neighbours are generated directly using modular polynomials. The different-discriminant and mixed cases are routed through reference quaternionic data and the KLPT algorithm.
4. We implement the CM-enhanced search in SageMath and Python. On small search parameters, enlarging the terminal union reduces both the average number of visited vertices and the $\mathbb{F}_p$-multiplication counts. Lookup benchmarks at SQIsign-size parameters show that the CM table membership test has stable overhead, while the main practical cost lies in preprocessing and storage.

Organization. Section 2 reviews the necessary background on supersingular isogeny graphs, the Delfs–Galbraith/SuperSolver algorithm, complex multiplication, Hilbert class polynomials, class group actions, and quaternionic methods. Section 3 constructs the CM terminal set $S_{\mathrm{CM}}(M)$ and analyzes the preprocessing stage. It gives bounds for the CM root supply, the overlap with the subfield subset S_p, and the cost of constructing and storing the precomputed CM terminal table. Section 4 presents the CM-enhanced search algorithm and the terminal-connection procedures. It also gives the corresponding complexity analysis. Section 5 gives worked examples illustrating the CM-enhanced search and

CM terminal connection procedures. Section 6 reports implementation measurements, including lookup overhead, visited-vertex counts, and operation counts. Section 7 concludes and discusses limitations and future directions.

2 Preliminaries

We begin by fixing the notation used throughout the paper and recalling the basic objects in supersingular isogeny graphs and complex multiplication. The main notation is summarized in Table 1.

Table 1. Basic notation.

Notation	Meaning
S_{p^2}	The set of supersingular j-invariants in $\mathbb{F}_{p^2}$
S_p	The prime-field supersingular subset $S_{p^2} \cap \mathbb{F}_p$
G_ℓ	The supersingular ℓ-isogeny graph on S_{p^2}
D	A negative fundamental discriminant in the analyzed core set
$\mathcal{D}_{\mathrm{inert}}(M)$	Negative fundamental discriminants D with $\lvert D\rvert < M$ and $\left(\frac{D}{p}\right) = -1$
$R_{\mathrm{CM}}(M)$	The raw CM supply $\sum_{D \in \mathcal{D}_{\mathrm{inert}}(M)} h(D)$
$N_p(M)$	The number of subfield CM roots contributed by the discriminant window
$S_{\mathrm{CM}}(M)$	The CM terminal set from roots of $H_D(X) \bmod p$, $D \in \mathcal{D}_{\mathrm{inert}}(M)$

Throughout the paper, p denotes an odd prime, $\mathbb{F}_p$ denotes the finite field with p elements, and $\mathbb{F}_{p^2}$ denotes its quadratic extension. We write S_{p^2} for the set of supersingular j-invariants over $\mathbb{F}_{p^2}$, and $S_p := S_{p^2} \cap \mathbb{F}_p$ for the prime field supersingular subset.

It is classical that $|S_{p^2}| = \left\lfloor \frac{p}{12} \right\rfloor + \epsilon_p, \epsilon_p \in \{0, 1, 2\}$, where ϵ_p depends only on $p \bmod 12$ [29, Thm. V.4.1]. Hence $|S_{p^2}| = \Theta(p)$, while $|S_p| = \Theta(p^{1/2})$ [15]. Thus a uniformly distributed supersingular vertex in S_{p^2} lies in S_p with probability $\Theta(p^{-1/2})$.

For $j \in S_{p^2}$, we have $j \in S_p$ if and only if $j^p = j$. Otherwise, j and j^p form a Frobenius-conjugate pair.

Supersingular isogeny graphs Let $\ell \geq 2$ be an integer with $\gcd(\ell, p) = 1$. The supersingular ℓ-isogeny graph G_ℓ has vertex set S_{p^2}, and its edges correspond to cyclic ℓ-isogenies between the associated supersingular elliptic curves, up to isomorphism. When ℓ is prime, G_ℓ is connected and, away from the usual exceptional vertices with extra automorphisms, is $(\ell + 1)$-regular. These graphs are Ramanujan and form expander families [24,27].

In this work the search phase is performed primarily in the 2-isogeny graph G_2. The Delfs–Galbraith algorithm searches from a given supersingular vertex until it reaches S_p. Under the standard rapid-mixing heuristic for non-backtracking walks in G_2, the expected number of visited vertices before hitting

S_p is approximately $\frac{|S_{p^2}|}{|S_p|} = \Theta(p^{1/2})$ [15]. This cost is the main bottleneck in the Delfs–Galbraith algorithm. SuperSolver reduces the concrete cost of this search by testing whether the current vertex has small-degree neighbours in S_p [11].

Delfs–Galbraith algorithm and SuperSolver The Delfs–Galbraith algorithm reduces the problem of connecting two supersingular vertices in S_{p^2} to the problem of reaching S_p. In practice, this search is implemented as a depth-first search (DFS) of the non-backtracking 2-isogeny tree, often with randomized branch choices. For a full path-finding instance with two input vertices j_A, j_B, the procedure is applied to both sides, producing $j_A \rightsquigarrow t_A, j_B \rightsquigarrow t_B; t_A, t_B \in S_p$. The second phase of the Delfs–Galbraith algorithm then connects t_A and t_B inside the $\mathbb{F}_p$-supersingular graph.

SuperSolver improves the terminal-detection step by avoiding full root factorization of the specialized modular polynomial $\Phi_\ell(X, j)$, where $\Phi_\ell(X, Y) \in \mathbb{Z}[X, Y]$ is the classical modular polynomial: for two j-invariants j_1, j_2, the equation $\Phi_\ell(j_1, j_2) = 0$ means that the corresponding elliptic curves are connected by a cyclic ℓ-isogeny [4,11]. For each selected ℓ, it specializes the modular polynomial at the current vertex, $f(X) = \Phi_\ell(X, j) \in \mathbb{F}_{p^2}[X]$. Writing $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$, it decomposes f into its real and imaginary parts

$$f(X) = g_1(X) + \beta g_2(X), g_1, g_2 \in \mathbb{F}_p[X].$$

Then f has a root in $\mathbb{F}_p$ if and only if g_1 and g_2 have a common root over $\mathbb{F}_p$. If g_1 and g_2 have a nonconstant greatest common divisor, then the current vertex has an ℓ-isogenous neighbour in S_p. The search can therefore terminate by following this ℓ-isogeny to the corresponding terminal.

Complex multiplication, Hilbert class polynomials, and class group actions Let $\mathbb{C}$ denote the field of complex numbers. The Hilbert class polynomial of discriminant D is

$$H_D(X) = \prod_{\substack{[E]/\cong \\ E/\mathbb{C},\ \mathrm{End}(E)\cong\mathcal{O}_D}} \big(X - j(E)\big) \in \mathbb{Z}[X],$$

where the product runs over isomorphism classes of complex elliptic curves with complex multiplication by $\mathcal{O}_D$. Since D is fundamental, the order $\mathcal{O}_D$ is the maximal order of the imaginary quadratic field $K = \mathbb{Q}(\sqrt{D})$, that is, $\mathcal{O}_D = \mathcal{O}_K$, where $\mathcal{O}_K$ is the ring of integers of K [13, Ch. 7]. The Hilbert class polynomial $H_D(X)$ has degree $\deg H_D = h(D)$, the class number of $\mathcal{O}_D$.

For later estimates, we will use Dirichlet's class number formula. For $D < -4$, $h(D) = \frac{\sqrt{|D|}}{\pi} L(1, \chi_D)$, where χ_D is the quadratic Dirichlet character defined by $\chi_D(n) = \left(\frac{D}{n}\right)$, with $\left(\frac{D}{n}\right)$ denoting the Kronecker symbol [18, Ch. 10].

We will later reduce $H_D(X)$ modulo p. If $\left(\frac{D}{p}\right) = -1$, then p is inert in $K_D = \mathbb{Q}(\sqrt{D})$. Deuring reduction of a CM elliptic curve with endomorphism

ring $\mathcal{O}_D$ gives a supersingular curve $E/\mathbb{F}_{p^2}$, together with an embedding

$$\iota_D : \mathcal{O}_D \hookrightarrow \mathrm{End}(E).$$

An $\mathcal{O}_D$-orientation is called optimal (or primitive) when

$$\iota_D(K_D) \cap \mathrm{End}(E) = \iota_D(\mathcal{O}_D).$$

Because D is fundamental, $\mathcal{O}_D = \mathcal{O}_{K_D}$ is already the maximal order of K_D, so the induced orientation is automatically optimal. Thus a root of $H_D(X)$ mod p is the j-invariant of a supersingular curve *admitting* an optimal $\mathcal{O}_D$-orientation [26, Prop. 3.2]. The polynomial records only the unoriented j-invariant: it specifies neither a curve equation nor a choice of optimal embedding. This distinction is important in the terminal-connection step of Section 4.2.

Deuring correspondence and quaternionic methods Let $B_{p,\infty}$ denote the quaternion algebra over $\mathbb{Q}$ ramified exactly at p and at infinity. The Deuring correspondence relates supersingular elliptic curves in characteristic p to maximal orders in $B_{p,\infty}$. More precisely, if $E/\mathbb{F}_{p^2}$ is supersingular, then $\mathrm{End}(E)$ is isomorphic to a maximal order in $B_{p,\infty}$. Conversely, maximal orders in $B_{p,\infty}$, up to the appropriate equivalence, arise as endomorphism rings of supersingular elliptic curves [16,21]. In particular, when $p \equiv 3 \pmod 4$, the supersingular vertex $j = 1728$ has CM by the order of discriminant -4, and its corresponding maximal order in $B_{p,\infty}$ is explicitly known.

Under this correspondence, separable isogenies are described by quaternionic ideals. If two supersingular curves E_1 and E_2 have endomorphism rings corresponding to maximal orders $\mathcal{R}_1 \simeq \mathrm{End}(E_1), \mathcal{R}_2 \simeq \mathrm{End}(E_2)$, then an isogeny from E_1 to E_2 corresponds to an ideal linking $\mathcal{R}_1$ and $\mathcal{R}_2$, the degree of which is given by the reduced norm of the corresponding ideal. KLPT algorithm converts such quaternionic ideals into isogeny paths of small-prime or powersmooth degree. Under their standard heuristic assumptions on the distribution of suitable primes, these algorithms run in probabilistic polynomial time [20].

3 Construction of CM Terminal Set

In this section, we construct a family of CM vertices forming an additional stopping set for the supersingular isogeny search. We then estimate its size and the corresponding preprocessing cost.

3.1 Definition and Construction of CM Terminal Set

CM reduction to supersingular vertices A classical way to generate supersingular elliptic curves over $\mathbb{F}_{p^2}$ is to reduce CM elliptic curves modulo p [3]. Let $D < 0$ be a negative fundamental discriminant. If p is inert in $\mathbb{Q}(\sqrt{D})$, equivalently $\left(\frac{D}{p}\right) = -1$, then Deuring's theorem implies that the reductions modulo

p of elliptic curves with CM by $\mathcal{O}_D$ are supersingular [16]. Hence the roots of $H_D(X) \bmod p$ in $\mathbb{F}_{p^2}$ are supersingular j-invariants.

This principle underlies CM-based supersingular-curve construction algorithms, such as Bröker's algorithm and its variants [3,25]. Here we use it to build a terminal set: for many inert fundamental discriminants D, we extract the roots of $H_D(X) \bmod p$ and store the resulting supersingular j-invariants as CM terminal vertices.

Definition of the CM Terminal Set Motivated by this construction, for a parameter $M > 0$, we define the CM terminal set

$$S_{\mathrm{CM}}(M) = \left\{ j \in \mathbb{F}_{p^2} : \begin{array}{l} D < 0 \text{ is a fundamental discriminant with } |D| < M, \\ \left(\frac{D}{p}\right) = -1, \\ j \text{ is a root of } H_D(X) \bmod p \end{array} \right\}.$$

For fundamental D, the order $\mathcal{O}_D$ is maximal and the Hilbert class polynomial $H_D(X)$ has degree exactly $h(D)$ [13, Chs. 7 and 13]. In addition to the j-invariant, the table stores its discriminant label D and a Frobenius-conjugation bit; the latter is needed because the implementation uses one canonical key for $\{j, j^p\}$.

Structural spread of CM terminals Love and Boneh show that CM vertices associated with different discriminants cannot be connected by very small-degree isogenies [23, Thm. 1.3(a)]. In the range $|D| < M$, vertices associated with distinct fundamental discriminants $D_1 \neq D_2$ are separated by an isogeny-degree lower bound of order $\Omega\left(\frac{\sqrt{p}}{M}\right)$. Although this does not prove that $S_{\mathrm{CM}}(M)$ is uniformly distributed, it shows that CM vertices arising from different discriminants are structurally separated.

For vertices with the same discriminant D, Algorithm 2 searches directly through small split-prime-degree isogenies generated by modular polynomials. These auxiliary isogenies may have degrees $\ell \neq 2$, and therefore need not correspond to short paths in the 2-isogeny graph G_2. More generally, the detour bounds of Love and Boneh illustrate that replacing an auxiliary ℓ-isogeny by isogenies of degree coprime to ℓ can be expensive [23, Prop. 6.1].

Thus, the CM terminal set should not be viewed as a single tightly clustered region of G_2. The separation results above provide structural support for treating $S_{\mathrm{CM}}(M)$ as an additional source of recognizable terminal vertices.

3.2 Size Estimates for the CM Terminal Set

Main estimates and heuristic assumptions. We first estimate the number of CM roots contributed by the inert fundamental discriminants used in the construction of $S_{\mathrm{CM}}(M)$. We denote by $\mathcal{D}_{\mathrm{inert}}(M)$ the set of fundamental discriminants D satisfying $|D| < M$ and $\left(\frac{D}{p}\right) = -1$, namely the discriminants

used in the definition of $S_{\mathrm{CM}}(M)$. Since $H_D(X)$ has degree $h(D)$, a discriminant $D \in \mathcal{D}_{\mathrm{inert}}(M)$ contributes $h(D)$ CM roots before reduction to distinct vertices. We work under the standing parameter assumption $M < \sqrt{p}$. Thus, for any two distinct discriminants $D_1, D_2 \in \mathcal{D}_{\mathrm{inert}}(M)$, we have $|D_1 D_2| < M^2 < p < 4p$. Kaneko proved that if two Hilbert class polynomials attached to distinct discriminants D_1 and D_2 have a common supersingular root modulo p, then $|D_1 D_2| \geq 4p$ [19]. Hence distinct discriminants in our range cannot contribute common supersingular roots modulo p.

We therefore introduce the class-number sum

$$R_{\mathrm{CM}}(M) := \sum_{D \in \mathcal{D}_{\mathrm{inert}}(M)} h(D),$$

which will serve as our primary estimate for the size of the CM terminal set. We also consider the subfield contribution

$$N_p(M) := \sum_{\substack{|D|<M \\ D \text{ fundamental}}} \#\{x \in \mathbb{F}_p : \ H_D(x) \equiv 0 \pmod{p}\}.$$

This quantity controls the possible overlap between CM terminal set and S_p. The main estimates used in the sequel are

$$R_{\mathrm{CM}}(M) \approx \frac{c_0}{\pi^3} M^{3/2}, \qquad N_p(M) = \widetilde{O}(M),$$

where c_0 denotes the mean of the random Euler product modeling $L(1, \chi_D)$ [17]. Thus the subfield contribution is lower order compared with the total CM root count: $N_p(M) \ll R_{\mathrm{CM}}(M)$. Consequently, most CM roots are expected to lie outside $\mathbb{F}_p$, and hence to provide new terminal vertices beyond S_p.

We make explicit the heuristic assumptions used in the estimates below.

(H1) *Inertness frequency.* For negative fundamental discriminants D with $|D| < M$, the condition $\left(\frac{D}{p}\right) = -1$ has frequency approximately $1/2$.

(H2) *Average behavior of $L(1, \chi_D)$.* The values $L(1, \chi_D)$ may be modeled on average by a random Euler product $L(1, X)$, with mean $c_0 := \mathbb{E}[L(1, X)]$; see Remark 1 for its Euler-product expression and numerical value.

Assumptions (H1)–(H2) enter our estimate of $R_{\mathrm{CM}}(M)$.

Granville and Soundararajan developed the relevant random Euler-product model [17]. It describes the distribution of $L(1, \chi_d)$ over fundamental discriminants. In that model, let $\mathbb{P}$ denote probability. For a fixed prime q, the random variable $\chi_d(q) = \left(\frac{d}{q}\right)$ has local distribution

$$\mathbb{P}(\chi_d(q) = 0) = \frac{1}{q+1}, \qquad \mathbb{P}(\chi_d(q) = 1) = \mathbb{P}(\chi_d(q) = -1) = \frac{q}{2(q+1)}.$$

Here $\chi_d(q) = 0$ corresponds to the local event $q \mid d$. Hence, after excluding this local event, the two nonzero signs occur with equal probability. Applying this

with $q = p$ gives the inertness-frequency model $\left(\frac{D}{p}\right) = -1$ with probability $1/2$, which is (H1).

The same random Euler-product model is used in (H2) to control the average size of $L(1, \chi_D)$. Combined with Dirichlet's class number formula $h(D) = \frac{\sqrt{|D|}}{\pi} L(1, \chi_D)$, it gives the heuristic order of magnitude for the class-number weighted CM supply in the discriminant window $|D| < M$.

The total CM root count We first estimate the raw number of CM roots contributed by the inert fundamental discriminants satisfying $|D| < M$.

Proposition 1 (Heuristic total CM root count). *Assume $M < \sqrt{p}$. Under the inertness-frequency model (H1) and the Granville–Soundararajan random Euler-product model (H2), the raw CM root count satisfies*

$$R_{\mathrm{CM}}(M) \sim \frac{c_0}{\pi^3} M^{3/2},$$

where $c_0 = \mathbb{E}[L(1, X)]$ is the mean of the random Euler product used to model $L(1, \chi_D)$. In particular, $R_{\mathrm{CM}}(M) = \Theta(M^{3/2})$.

Proof. For a negative fundamental discriminant $D < -4$, Dirichlet's class number formula gives $h(D) = \frac{\sqrt{|D|}}{\pi} L(1, \chi_D)$, where $\chi_D(n) = \left(\frac{D}{n}\right)$ is the associated quadratic Dirichlet character [13,18]. The exceptional discriminants $D = -3, -4$ contribute only $O(1)$ to the estimates below.

By definition, $R_{\mathrm{CM}}(M) = \sum_{D \in \mathcal{D}_{\mathrm{inert}}(M)} h(D)$. Using the class number formula, we get

$$R_{\mathrm{CM}}(M) = \frac{1}{\pi} \sum_{D \in \mathcal{D}_{\mathrm{inert}}(M)} \sqrt{|D|}\, L(1, \chi_D) + O(1).$$

By (H1), the inertness condition $\left(\frac{D}{p}\right) = -1$ selects one half of the admissible fundamental discriminants, up to the local event $p \mid D$. By (H2), the average value of $L(1, \chi_D)$ is modeled by $c_0 = \mathbb{E}[L(1, X)]$. Therefore,

$$R_{\mathrm{CM}}(M) \sim \frac{c_0}{2\pi} \sum_{\substack{D<0 \\ D \text{ fundamental} \\ |D|<M}} \sqrt{|D|}.$$

The standard density estimate for negative fundamental discriminants [5] is

$$\#\{D < 0 : \ D \text{ fundamental}, \ |D| \le M\} \sim \frac{3}{\pi^2} M.$$

This also follows from the classification of fundamental discriminants together with the density of squarefree integers in arithmetic progressions [9,18]. Hence,

by partial summation,

$$\sum_{\substack{D<0\\ D \text{ fundamental}\\ |D|<M}} \sqrt{|D|} \sim \frac{3}{\pi^2}\int_0^M t^{1/2}\,dt = \frac{2}{\pi^2}M^{3/2}.$$

Substituting this into the previous estimate gives

$$R_{\mathrm{CM}}(M) \sim \frac{c_0}{2\pi}\cdot\frac{2}{\pi^2}M^{3/2} = \frac{c_0}{\pi^3}M^{3/2}.$$

This proves the claimed heuristic asymptotic. □

Remark 1 (The constant c_0). The constant c_0 is the mean value in the Granville–Soundararajan random Euler-product model for $L(1,\chi_D)$ [17]. If $X(q)\in\{0,\pm1\}$ denotes the local random variable at a prime q, then

$$c_0 = \mathbb{E}[L(1,X)] = \prod_q \mathbb{E}\left[\left(1-\frac{X(q)}{q}\right)^{-1}\right].$$

Numerically, this constant is approximately 1.45. Its precise value affects only the leading constant; the growth rate $R_{\mathrm{CM}}(M)=\Theta(M^{3/2})$ is the quantity used in the later complexity analysis.

CM roots in $\mathbb{F}_p$ We next bound the subfield contribution $N_p(M)$, which measures the part of the CM roots that may overlap with S_p. Since we work under the standing assumption $M<\sqrt{p}$, we have $p>|D|$ for every discriminant in the window $|D|<M$. Hence the root-count theorem of Chen and Xue [8, Thm. 1.1] applies to the $\mathbb{F}_p$-roots of $H_D(X)$.

Proposition 2. *Assume $M<\sqrt{p}$. Then the prime-field contribution satisfies $N_p(M)=O(M\log M)$. In particular, $N_p(M)=\widetilde{O}(M)$, which is lower order than the heuristic raw CM supply $R_{\mathrm{CM}}(M)=\Theta(M^{3/2})$.*

Proof. Let $D<0$ be a fundamental discriminant with $|D|<M$, and let $r(D)$ denote the number of distinct odd prime divisors of D. Chen and Xue prove that the set of $\mathbb{F}_p$-roots of $H_D(X)$ mod p is either empty or admits a free and transitive action of $\mathrm{Pic}(\mathcal{O}_D)[2]$ [8, Thm. 1.1]. Its cardinality is therefore either zero or $|\mathrm{Pic}(\mathcal{O}_D)[2]|$. By genus theory, this 2-torsion subgroup has size $O(2^{r(D)})$ [13, Prop. 3.11].

Therefore,

$$N_p(M) \ll \sum_{\substack{|D|<M\\ D \text{ fundamental}}} 2^{r(D)}.$$

Since the odd part of D is squarefree and has $r(D)$ distinct prime factors, it has $2^{r(D)}$ divisors, all of which divide $|D|$. Hence $2^{r(D)}\le\tau(|D|)$, where $\tau(n)$ denotes

the number of positive divisors of n.. By the classical average order of the divisor function, $\sum_{n\le M}\tau(n) = O(M\log M)$ [1, Thm. 3.3]. It follows that

$$\sum_{\substack{|D|<M \\ D \text{ fundamental}}} 2^{r(D)} \le \sum_{n\le M}\tau(n) = O(M\log M).$$

Thus $N_p(M) = O(M\log M)$. □

We get $N_p(M) = \widetilde{O}(M), R_{\mathrm{CM}}(M) = \Theta(M^{3/2})$. Thus the CM roots lying in $\mathbb{F}_p$ form a lower-order contribution to the CM terminal set.

3.3 Preprocessing Cost

We now estimate the preprocessing cost required to construct $S_{\mathrm{CM}}(M)$ and store it in a CM terminal table. The goal of the preprocessing phase is to produce a table that supports fast membership tests during the search phase. For each inert fundamental discriminant D with $|D| < M$, this requires three operations:

$$H_D(X) \bmod p \quad \longrightarrow \quad \text{roots in } \mathbb{F}_{p^2} \quad \longrightarrow \quad \text{table insertion.}$$

The resulting preprocessing cost has two main arithmetic contributions: the computation of the Hilbert class polynomials and the extraction of their roots over $\mathbb{F}_{p^2}$.

The computation of Hilbert class polynomials. We first consider the Hilbert class polynomials. Since only $H_D(X) \bmod p$ is needed, Sutherland's Chinese remainder theorem (CRT) method for computing Hilbert class polynomials modulo a prescribed integer provides an appropriate reference complexity [31]. Under the Generalized Riemann Hypothesis (GRH), its expected bounds are

$$\begin{aligned}\text{time} &= O\big(|D|\log^5|D|(\log\log|D|)^4\big)\,,\\ \text{space} &= O\Big(|D|^{1/2}(\log|D| + \log p)\log\log|D|\Big)\,.\end{aligned}$$

In the heuristic notation used here, we summarize the time cost per discriminant as $\widetilde{O}(|D|)$. Since $\sum_{D\in\mathcal{D}_{\mathrm{inert}}(M)}|D| = O(M^2)$, the aggregate class polynomial computation cost is $\widetilde{O}(M^2)$.

Root extraction. We next consider root extraction. Once $H_D(X) \bmod p$ has been computed, we extract its roots over $\mathbb{F}_{p^2}$. Since $\deg H_D = h(D)$, the total degree of all polynomials processed in this step is $\sum_{D\in\mathcal{D}_{\mathrm{inert}}(M)} h(D) = R_{\mathrm{CM}}(M)$. Using the finite field root extraction estimate employed in extended Bröker algorithms for generating supersingular CM j-invariants [25], the aggregate root-extraction cost is $\widetilde{O}\big(R_{\mathrm{CM}}(M)(\log p)^2\big)$. By Proposition 1, this contribution becomes $\widetilde{O}\big(M^{3/2}(\log p)^2\big)$.

Table insertion and storage. Finally, each extracted root is inserted into the terminal table once. Thus the number of insertions is $O(R_{\mathrm{CM}}(M))$. An element of $\mathbb{F}_{p^2}$ can be represented by two elements of $\mathbb{F}_p$. Therefore the storage required for the CM j-invariants is $O(R_{\mathrm{CM}}(M)\log p) = O(M^{3/2}\log p)$ bits. In the implementation, Frobenius-conjugate roots are stored only once: for $j = a + b\beta$, we store the canonical key $(a, \min\{b, -b \bmod p\})$, which represents the pair $\{j, j^p\}$. This Frobenius compression changes only the constant factor in the table size. Hence the storage remains $O(M^{3/2}\log p)$ bits.

Combining the class polynomial computation and root extraction costs, the total preprocessing time for constructing $S_{\mathrm{CM}}(M)$ is heuristically bounded by $\widetilde{O}(M^2)$. The persistent storage for the stored CM j-invariants is $O(M^{3/2}\log p)$ bits, excluding temporary working memory for class polynomial computation and data structure overhead.

Modular-polynomial data. Algorithm 2 repeatedly uses modular polynomials to generate small-degree isogeny steps. We therefore precompute $\Phi_\ell \bmod p$ once and reuse it. Under GRH, the class group of $\mathcal{O}_D$ is generated by prime ideals of norm $O(\log^2|D|)$ [2]. Since $|D| < M$, it suffices to use the global degree bound $L(M) = O(\log^2 M)$. Accordingly, we precompute the reduced modular polynomials for the prime degrees $\ell \le L(M)$. A smaller bound may be used in practice when the required connectivity is verified directly.

For each prime ℓ, $\Phi_\ell \bmod p$ can be computed in expected time $\widetilde{O}(\ell^3)$, and its representation requires $O(\ell^2\log p)$ bits [4,22]. Summing over the prime degrees $\ell \le L(M)$, the total preprocessing costs are $\widetilde{O}(L(M)^4)$ time and $\widetilde{O}(L(M)^3\log p)$ bits. With $L(M) = O(\log^2 M)$, these become $\widetilde{O}(\log^8 M)$ and $\widetilde{O}(\log^6 M\log p)$, respectively. They are therefore lower order than the $\widetilde{O}(M^2)$ construction of the CM terminal table. The reduced modular polynomials are reusable for all discriminants and path-finding instances with the same characteristic p.

4 CM-Enhanced Search Algorithm and Complexity

This section describes the search phase of our CM-enhanced path-finding algorithm. Starting from the Delfs–Galbraith/SuperSolver algorithm, we modify the terminal tests by adding membership queries against $S_{\mathrm{CM}}(M)$, implemented through a precomputed CM terminal table. We present the resulting bidirectional search procedure, discuss the terminal cases and the corresponding path-connection methods, and then analyze the costs introduced by the CM enhancement: the lookup overhead, the effect of the enlarged terminal set on the expected hitting time, and the post-processing cost for path connection.

4.1 CM-Enhanced Search Procedure

We describe the search procedure obtained by incorporating the precomputed CM terminal table into the Delfs–Galbraith/SuperSolver algorithm. The modification is made at the terminal test: the algorithm first checks whether any

candidate belongs to $S_{\mathrm{CM}}(M)$, and then applies the original subfield-based terminal tests.

Given two input vertices $j_1, j_2 \in \mathbb{F}_{p^2}$, the algorithm runs two searches, one from each endpoint, until both walks reach terminal vertices. The terminal vertices may be subfield terminals or CM terminals. Once the terminal vertices have been obtained, a terminal-to-terminal connection is computed using the appropriate method, and the final path is obtained by concatenating the two recorded walk segments with this terminal connection.

The resulting procedure is summarized in Algorithm 1. The algorithm treats the terminal-to-terminal connection as a separate step, since the appropriate method depends on the types of the two terminals obtained by the searches. We discuss these terminal cases in the next subsection.

Algorithm 1: CM-enhanced isogeny path search

Input: Prime p; vertices $j_1, j_2 \in \mathbb{F}_{p^2}$; CM bound M; search parameters.
Output: An isogeny path from j_1 to j_2, or failure.

Preprocessing. Construct $S_{\mathrm{CM}}(M)$ and store it, together with the associated CM data, in the CM terminal table;

Search. Run the Delfs–Galbraith/SuperSolver search from j_1 and j_2. Whenever candidate next vertices are obtained from a modular-polynomial equation, first test membership in $S_{\mathrm{CM}}(M)$, and then apply the original subfield-based terminal tests. Stop when both walks reach terminal vertices t_1, t_2, and record the corresponding walk segments and terminal data;

Terminal connection. Connect t_1 and t_2 using the method determined by their terminal types;

Output. Concatenate $j_1 \rightsquigarrow t_1, t_1 \rightsquigarrow t_2, t_2 \rightsquigarrow j_2$ to obtain a path from j_1 to j_2.;

4.2 Terminal Cases and Path Connection

After the bidirectional search stage, we obtain two walk segments $j_1 \rightsquigarrow t_1, j_2 \rightsquigarrow t_2$, where t_1 and t_2 are terminal vertices detected by the stopping condition. It remains to connect the two terminal vertices t_1 and t_2.

The connection procedure depends on the types of the two terminals. A terminal may lie in S_p, or it may be a CM terminal in $S_{\mathrm{CM}}(M)$. We distinguish three terminal cases:

$$(S_p \rightsquigarrow S_p), \qquad (S_{\mathrm{CM}}(M) \rightsquigarrow S_{\mathrm{CM}}(M)), \qquad (S_{\mathrm{CM}}(M) \rightsquigarrow S_p).$$

The first case is handled by the Delfs–Galbraith subfield connection. The CM–CM case splits into two subcases, depending on whether the two CM terminals come from the same discriminant D or from distinct discriminants $D_1 \neq D_2$. The mixed CM–subfield case is treated separately. We describe each connection procedure separately.

Same-discriminant CM connection Our same-discriminant connection is adapted from the bidirectional isogeny search of Delfs and Galbraith [15]. Let $E_0, E_1/\mathbb{F}_p$ be supersingular elliptic curves. Their $\mathbb{F}_p$-rational endomorphism rings are quadratic orders in $\mathbb{Q}(\sqrt{-p})$. If a prime $\ell \neq p$ satisfies $\left(\frac{-p}{\ell}\right) = 1$, then ℓ splits in the corresponding quadratic order, and the prime ideals above ℓ give horizontal $\mathbb{F}_p$-rational ℓ-isogenies. Delfs and Galbraith therefore choose a degree set

$$\mathcal{L}(B) = \left\{ \ell < B : \ell \text{ prime and } \left(\frac{-p}{\ell}\right) = 1 \right\}$$

and perform bidirectional random walks from E_0 and E_1 using isogenies of degrees in $\mathcal{L}(B)$. When the two sets of visited vertices intersect, the recorded walk segments yield an isogeny path between the endpoints. The choice of B is justified by the class-group interpretation. Under GRH, prime ideals of norm at most $6\log^2|d|$ generate the ideal class group of a quadratic order of discriminant d [2,15]. Hence one may take $B = O(\log^2|d|)$ as a theoretical bound. In practice, Delfs and Galbraith use a smaller degree bound, since a much smaller collection of split primes often already generates the relevant graph.

Algorithm 2: Same-discriminant CM connection

Input: CM terminals $t_0, t_1 \in \mathbb{F}_{p^2}$ with the same discriminant label D, and a set L_D.
Output: A path S from t_0 to t_1.
$S \leftarrow [\,]$;
$S_0 \leftarrow [t_0]$, $S_1 \leftarrow [t_1]$;
disjoint $\leftarrow$ true;
$i \leftarrow 0$;
while disjoint **do**
  $\ell \leftarrow_R \mathcal{L}_D$;(Here $\leftarrow_R$ denotes uniform random sampling from the set L_D.)
  $\Psi \leftarrow \Phi_\ell \bmod p$;
  $j \leftarrow_R \left\{ j' \in \mathrm{Roots}_Y\big(\Psi(\mathrm{Last}(S_i), Y)\big) : j' \text{ has CM discriminant label } D \right\}$;
  Append(S_i, j);
  if $j \in S_{1-i}$ **then**
    disjoint $\leftarrow$ false;
    $S \leftarrow \mathrm{Cat}(S_0[1, \ldots, \mathrm{Index}(S_0, j)], S_1[\mathrm{Index}(S_1, j) - 1, \ldots, 1])$;
  $i \leftarrow 1 - i$;
return S;

As shown in Algorithm 2, we adapt this strategy to two CM terminals t_1, t_2 having the same fundamental discriminant label D. We define

$$\mathcal{L}_D(B_D) = \left\{ \ell < B_D : \ell \text{ prime}, \ \ell \nmid pD, \text{ and } \left(\frac{D}{\ell}\right) = 1 \right\}.$$

Under GRH, the theoretical choice $B_D = O(\log^2 |D|)$ is sufficient at the level of class-group generators. Since all discriminants used in the preprocessing satisfy $|D| < M$, a single global bound $L(M) = O(\log^2 M)$ may be used to precompute the required modular polynomials. In the implementation, a smaller initial value of B_D may be used and increased if the resulting degree set does not connect the required component.

Starting from t_1 and t_2, we perform bidirectional random walks using the degree set $\mathcal{L}_D(B_D)$. Each visited vertex is stored together with its predecessor and the degree of the corresponding isogeny. Once the two sets of visited vertices intersect, the two recorded walk segments are concatenated to obtain an isogeny path from t_1 to t_2.

Different-discriminant CM connection *Reference quaternionic data.* We first recall how the reference quaternionic data attached to a fundamental discriminant D_i are obtained, following the construction of Love and Boneh [23, Sec. 7]. Let $K_i = \mathbb{Q}(\sqrt{D_i})$. Since D_i is fundamental, the corresponding quadratic order is the full ring of integers $\mathcal{O}_{D_i} = \mathcal{O}_{K_i}$. Equivalently, $\mathcal{O}_{D_i}$ is generated by an element α_i satisfying $4\alpha_i^2 = D_i$ if $D_i \equiv 0 \pmod 4$ and $4\alpha_i^2 - 4\alpha_i + 1 = D_i$ if $D_i \equiv 1 \pmod 4$, thus $\mathbb{Z}[\alpha_i] \simeq \mathcal{O}_{D_i}$.

The next step is to realize this quadratic order inside the quaternion algebra $B_{p,\infty}$. By the embedding criterion for quadratic fields into quaternion algebras, there exists an embedding $K_i \hookrightarrow B_{p,\infty}$ under the inertness condition used in the CM construction [35, Prop. 14.6.7]. Equivalently, after fixing a presentation of $B_{p,\infty}$, the construction reduces to solving the corresponding norm equation. Solving this norm equation gives an element of $B_{p,\infty}$ satisfying the same quadratic equation as the generator of $\mathcal{O}_{D_i}$. Hence we obtain an explicit embedding $\mathcal{O}_{D_i} \hookrightarrow B_{p,\infty}$.

After this embedding is fixed, choose a maximal order $\mathcal{R}_{D_i} \subset B_{p,\infty}$ containing the image of $\mathcal{O}_{D_i}$. Since $\mathcal{O}_{D_i}$ is the maximal order of K_i, this embedding is optimal: $\mathcal{R}_{D_i} \cap K_i = \mathcal{O}_{D_i}$. In the form of the Deuring correspondence used here, the conjugacy class of $\mathcal{R}_{D_i}$ determines a supersingular j-invariant only up to the Galois action $a_{D_i} \longleftrightarrow a_{D_i}^p$; it does not determine a Weierstrass equation. We therefore choose a representative a_{D_i} of this Galois orbit, a curve model $A_{D_i}/\mathbb{F}_{p^2}$ with $j(A_{D_i}) = a_{D_i}$, and an identification $\mathcal{R}_{D_i} \simeq \mathrm{End}(A_{D_i})$ transporting the chosen optimal embedding. With these choices, A_{D_i} is optimally $\mathcal{O}_{D_i}$-oriented and $H_{D_i}(a_{D_i}) \equiv 0 \pmod p$. Thus the reference data consist of a chosen Galois representative, a curve model, and its quaternionic identification; none of these choices is determined by the abstract maximal order alone.

Different-discriminant CM connection. We next consider the case where both terminals are CM, but arise from different fundamental discriminants. Let $t_1, t_2 \in S_{\mathrm{CM}}(M), D_1 \neq D_2$, where t_i is a root of $H_{D_i}(X) \bmod p$.

The connection is therefore routed through the reference CM vertices constructed above. For each $i = 1, 2$, the terminal t_i and the reference vertex a_{D_i} lie in the same H_{D_i}-root set. Since the two vertices have the same discriminant label, they are connected using Algorithm 2.

Algorithm 3: Different-discriminant CM connection

Input: CM terminals t_1, t_2 with discriminant labels $D_1 \neq D_2$, and reference data $(a_{D_i}, \mathcal{R}_{D_i})$ for $i = 1, 2$.
Output: An isogeny path from t_1 to t_2, if found
for $i = 1, 2$ **do**
  Retrieve the reference CM vertex a_{D_i} and the corresponding maximal order $\mathcal{R}_{D_i} \subset B_{p,\infty}$;
  Run Algorithm 2 to connect $t_i \rightsquigarrow a_{D_i}$ using discriminant label D_i;
Compute a connecting ideal between the known maximal orders $\mathcal{R}_{D_1}$ and $\mathcal{R}_{D_2}$. Apply a KLPT algorithm to convert this connecting ideal into an explicit isogeny path $a_{D_1} \rightsquigarrow a_{D_2}$.
Return the concatenated path $t_1 \rightsquigarrow a_{D_1} \rightsquigarrow a_{D_2} \rightsquigarrow t_2$.

After these two local connections have been found, it remains to connect the two reference vertices $a_{D_1} \rightsquigarrow a_{D_2}$. This middle connection is computed quaternionically: one computes a connecting ideal between the known maximal orders $\mathcal{R}_{D_1}, \mathcal{R}_{D_2}$, and then applies a KLPT algorithm to convert this ideal into an explicit isogeny path [20]. The complete terminal path is the concatenation $t_1 \rightsquigarrow a_{D_1}, a_{D_1} \rightsquigarrow a_{D_2}, a_{D_2} \rightsquigarrow t_2$.

Mixed CM–subfield connection It remains to consider the mixed case, where one is CM terminal and the other lies in S_p. Let $t_{\mathrm{CM}} \in S_{\mathrm{CM}}(M), t_{\mathbb{F}_p} \in S_p$, and suppose that t_{CM} is a root of $H_D(X)$ mod p for a fundamental discriminant D.

There are two possible connection routes. If $H_D(X)$ mod p has a root $b_D \in \mathbb{F}_p$, then b_D is a natural bridge between the CM terminal and the subfield terminal. Since t_{CM} and b_D lie in the same H_D-root set, they are connected using Algorithm 2. The remaining segment $b_D \rightsquigarrow t_{\mathbb{F}_p}$ is then computed inside the $\mathbb{F}_p$-rational supersingular graph using the Delfs–Galbraith subfield connection.

If no such prime-field root b_D is available, we route the connection through fixed reference data. Let a_D be the reference CM vertex associated with D, as in the different-discriminant connection, and let $b_0 \in S_p$ be a fixed subfield reference vertex with known quaternionic data. For example, when $p \equiv 3 \pmod 4$, one may take b_0 to be the vertex with $j = 1728$, which has CM by the order of discriminant -4. Alternatively, when $p \equiv 2 \pmod 3$, one may use the vertex with $j = 0$, corresponding to CM discriminant -3. In this route, we first connect $t_{\mathrm{CM}} \rightsquigarrow a_D$ using the same-discriminant CM connection. We then connect the reference vertices $a_D \rightsquigarrow b_0$ using a quaternionic connecting ideal and a KLPT algorithm. Finally, we connect $b_0 \rightsquigarrow t_{\mathbb{F}_p}$ inside the $\mathbb{F}_p$-rational supersingular graph using the Delfs–Galbraith subfield procedure. Thus the full mixed path is $t_{\mathrm{CM}} \rightsquigarrow a_D \rightsquigarrow b_0 \rightsquigarrow t_{\mathbb{F}_p}$.

4.3 Complexity Analysis

This subsection analyzes the asymptotic and heuristic costs of the CM-enhanced search procedure. The goal is to isolate the contributions of the $S_{\mathrm{CM}}(M)$ lookup

Algorithm 4: Mixed CM–subfield connection

Input: A CM terminal t_{CM} with discriminant label D; a subfield terminal $t_{\mathbb{F}_p} \in S_p$; the CM terminal table; and reference data $(a_D, \mathcal{R}_D)$.
Output: An isogeny path from t_{CM} to $t_{\mathbb{F}_p}$, if found.
if *the CM terminal table contains a vertex* $b_D \in \mathbb{F}_p$ *with discriminant label* D **then**
  Run Algorithm 2 to connect $t_{\mathrm{CM}} \rightsquigarrow b_D$ using discriminant label D;
  Connect $b_D \rightsquigarrow t_{\mathbb{F}_p}$ inside the $\mathbb{F}_p$-rational supersingular graph using the Delfs–Galbraith subfield connection;
  return *the concatenated path* $t_{\mathrm{CM}} \rightsquigarrow b_D \rightsquigarrow t_{\mathbb{F}_p}$;
else
  Choose a fixed subfield reference vertex $b_0 \in S_p$ with known quaternionic data;
  Run Algorithm 2 to connect $t_{\mathrm{CM}} \rightsquigarrow a_D$ using discriminant label D;
  Compute a connecting ideal between the maximal orders attached to a_D and b_0;
  Apply a KLPT algorithm to obtain an explicit path $a_D \rightsquigarrow b_0$;
  Connect $b_0 \rightsquigarrow t_{\mathbb{F}_p}$ inside the $\mathbb{F}_p$-rational supersingular graph using the Delfs–Galbraith subfield connection;
  return *the concatenated path* $t_{\mathrm{CM}} \rightsquigarrow a_D \rightsquigarrow b_0 \rightsquigarrow t_{\mathbb{F}_p}$.

and the subsequent terminal-connection step. Implementation details, parameter choices, and empirical timings are deferred to Section 6.

Lookup overhead. The CM enhancement does not change the underlying random walk. Its additional cost is the membership test in the precomputed CM terminal table storing $S_{\mathrm{CM}}(M)$. After a candidate neighbour j' is computed, the algorithm checks whether $j' \in S_{\mathrm{CM}}(M)$. If the CM terminal table is implemented as a hash table indexed by the representation of $j' \in \mathbb{F}_{p^2}$, then membership queries have expected $O(1)$ cost under standard assumptions on the hash function and load factor [10, Ch. 11]. Thus the per-step overhead introduced by the CM terminal set is constant.

Expected number of visited vertices We measure the effect of the enlarged terminal set on the expected search length in the 2-isogeny graph G_2. Under the rapid-mixing heuristic used in the Delfs–Galbraith analysis, set

$$T(M) := S_p \cup S_{\mathrm{CM}}(M).$$

After mixing, a visited vertex is modeled as uniform in S_{p^2}, so the probability of hitting the terminal set at a given visit is approximately $|T(M)|/|S_{p^2}|$. If τ_T counts inspected vertices up to and including the first terminal hit, then

$$\mathbb{E}[\tau_T] \approx \frac{|S_{p^2}|}{|S_p \cup S_{\mathrm{CM}}(M)|} = \frac{|S_{p^2}|}{|S_p| + |S_{\mathrm{CM}}(M)| - |S_p \cap S_{\mathrm{CM}}(M)|}.$$

The corresponding running-time contribution is obtained by multiplying the expected search length by the per-step cost. Let C_{step} denote the average cost, measured in $\mathbb{F}_p$-multiplications, of one step of the SuperSolver walk. This includes computing the non-backtracking 2-isogeny neighbours and performing the subfield-neighbour tests used by SuperSolver [11]. The CM enhancement adds one membership query to the precomputed CM terminal table storing $S_{\mathrm{CM}}(M)$ for each inspected candidate. With a hash-table implementation, this lookup has expected $O(1)$ operation count under standard assumptions on hashing. Hence the expected search cost is heuristically $C_{\mathrm{search}} \approx \mathbb{E}[\tau_T]\big(C_{\mathrm{step}} + O(1)\big)$.

Path connection cost After the search phase has reached terminal vertices, the remaining cost is the terminal-to-terminal connection cost. This cost depends on the types of the two terminals. In this subsection, we count only the per-instance connection cost.

Subfield–subfield connection. If both terminals lie in S_p, then the connection is exactly the Delfs–Galbraith subfield connection [15,11], and the cost is heuristically $\widetilde{O}(p^{1/4})$.

Same-discriminant CM–CM connection. Now suppose that both terminals are CM and have the same fundamental discriminant label D. Assuming that the selected degrees connect the relevant same-discriminant graph and that the walks mix rapidly, its number of vertices is at most $h(D)$. By the birthday-collision heuristic, the two visited sets are expected to intersect after $\widetilde{O}(\sqrt{h(D)})$ vertex expansions, with comparable storage. Since $h(D) = \widetilde{O}(|D|^{1/2})$, we obtain $\widetilde{O}(\sqrt{h(D)}) = \widetilde{O}(|D|^{1/4})$ modular-polynomial graph steps. Uniformly for $|D| < M$, this becomes $\widetilde{O}(M^{1/4})$ graph steps. Each step specializes a precomputed modular polynomial, computes its roots over $\mathbb{F}_{p^2}$, and retains only neighbours with discriminant label D. Constructing the explicit isogeny chain is output-sensitive.

Different-discriminant CM–CM connection. Suppose that the two CM terminals have distinct fundamental discriminants $D_1 \neq D_2$. Algorithm 3 first connects each terminal to a reference vertex with the same discriminant, and then connects the two reference vertices using a quaternionic connecting ideal and a KLPT-type algorithm.

By the preceding analysis, the two same-discriminant searches require

$$\widetilde{O}\Big(|D_1|^{1/4} + |D_2|^{1/4}\Big)$$

modular-polynomial graph steps. Constructing the reference quaternionic data [23], when not already available, and performing the subsequent KLPT connection require probabilistic polynomial time under the usual smoothness heuristics. Hence, uniformly for $|D_1|, |D_2| < M$, the total cost is $\widetilde{O}(M^{1/4})$ graph steps, together with $\mathrm{poly}(M, \log p)$ quaternionic work. The dependence on M in this

bound mainly arises from the relative norm-equation step. For $p \equiv 3 \pmod 4$, this equation is solved over $K(i)/K, K = \mathbb{Q}(\sqrt{D})$. Simon's algorithm reduces the problem to relative class group computation [30]. The relative class group can be obtained from the absolute class groups of $K(i)$ and K. Since these number fields have fixed degree and $|\operatorname{disc}(K(i))| = O(|D|^2)$, Buchmann's probabilistic subexponential class group algorithm gives the more precise estimate

$$\exp\Big(O\Big(\sqrt{\log|D|\ \log\log|D|}\Big)\Big)$$

for computation of the relative norm equation [6]. Consequently, uniformly for $|D| < M$, this step costs

$$\exp\Big(O\Big(\sqrt{\log M\ \log\log M}\Big)\Big)\,.$$

Thus, its dependence on M is smaller than M^{ε} for every fixed $\varepsilon > 0$, asymptotically.

Mixed CM–subfield connection. Finally, suppose that one terminal is a CM terminal of fundamental discriminant D, while the other lies in S_p.

If $H_D(X) \bmod p$ has a root in $\mathbb{F}_p$, this root is used as a bridge. The connection then consists of one same-discriminant search and one Delfs–Galbraith subfield connection, with cost $\widetilde{O}(|D|^{1/4}) + \widetilde{O}(p^{1/4})$ graph steps.

Otherwise, the CM terminal is first connected to a reference vertex of discriminant D. A quaternionic connecting ideal and a KLPT-type algorithm then connect this reference vertex to a fixed subfield reference vertex, after which the Delfs–Galbraith procedure completes the path in S_p. Under the usual KLPT smoothness heuristics, the total cost is $\widetilde{O}(|D|^{1/4} + p^{1/4})$ graph steps, together with $\operatorname{poly}(M, \log p)$ quaternionic work.

Uniformly for $|D| < M$, both cases are bounded by $\widetilde{O}(M^{1/4} + p^{1/4})$ graph steps, with an additional $\operatorname{poly}(M, \log p)$ term only when the quaternionic reference route is required. The quaternionic reference route incurs an additional $\operatorname{poly}(M, \log p)$ cost; as analyzed above, the explicitly quantified dependence on M in its relative norm-equation step is subpolynomial.

5 Worked Examples

We now give two worked examples to illustrate the CM-enhanced search and connection procedure. The purpose of the examples is twofold. First, we show how a CM terminal is detected during the search. Second, we demonstrate how the stored discriminant information is used to connect the terminal-to-terminal connection.

We work over $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$, where $\beta^2 = c$ for a fixed nonsquare $c \in \mathbb{F}_p$. Each element of $\mathbb{F}_{p^2}$ is written as $a + b\beta$, with $a, b \in \mathbb{F}_p$.

5.1 A Complete Small-parameter Example

For the toy example, we take $p = 2^{19} - 1$ and $M = 512$. The preprocessing phase computes the CM terminal set $S_{\mathrm{CM}}(M)$. For each stored root, the table records the corresponding discriminant D.

To generate two pseudo-random supersingular starting vertices, we follow the sampling method used in SuperSolver. Starting from a known supersingular vertex $j \in S_p$, we take sufficiently long random walks in a small-degree isogeny graph, for example in G_3, using independent random choices. This gives two pseudo-random supersingular j-invariants in $S_{p^2} \setminus S_p$ as

$$j_1 = 103962 + 468792\beta, \qquad j_2 = 36009 + 267892\beta.$$

Starting from j_1 and j_2, the CM-enhanced search runs two independent non-backtracking walks in the 2-isogeny graph G_2. At each visited vertex j, the algorithm checks whether $j \in S_p$, and also checks whether $j \in S_{\mathrm{CM}}(M)$. In this example, the first walk reaches the CM terminal $t_1 = 318697 + 441925\beta$ after visiting 119 vertices, and the lookup table returns the discriminant $D_1 = -323$. The second walk reaches the subfield terminal $t_2 = 423460 \in S_p$ after visiting 77 vertices. The recorded 2-isogeny paths from j_1 to t_1 and from j_2 to t_2 are given in Appendix A.

This yields a mixed CM–subfield terminal pair: $t_1 \in S_{\mathrm{CM}}(M), t_2 \in S_p$. Since $H_{-323}(X) \bmod p$ has no root in $\mathbb{F}_p$, the connection cannot use a same-discriminant subfield bridge. We therefore use the reference-vertex route of Algorithm 4.

Following the Love–Boneh reference construction, the discriminant $D = -323$ gives a maximal order whose corresponding CM vertex is $a_{-323} = 443990 + 439877\beta$. The terminal connection decomposes as

$$t_1 \rightsquigarrow a_{-323} \rightsquigarrow (j = 1728) \rightsquigarrow t_2.$$

The first segment connects t_1 to a_{-323} inside the same $D = -323$ CM root set. The middle segment connects a_{-323} to the fixed prime-field anchor $j = 1728$ via the KLPT quaternionic connection; in this example, the corresponding isogeny has total degree 2^{76}. The last segment connects the vertex $j = 1728$ to the subfield terminal t_2. The full connected terminal-to-terminal path is also displayed in Appendix A.

Combining the two search paths and the terminal connection gives a complete isogeny path

$$j_1 \rightsquigarrow t_1 \rightsquigarrow a_{-323} \rightsquigarrow (j = 1728) \rightsquigarrow t_2 \rightsquigarrow j_2,$$

where the final segment is obtained by reversing the recorded search path from j_2 to t_2. This example illustrates the mixed-terminal case: after the search stops at a CM terminal, the discriminant label stored in the CM terminal table determines the appropriate connection route.

5.2 A Large-parameter Connection Example

We now give a larger-parameter example focusing on the connection stage. Unlike the complete example above, this example does not include the initial random-walk search from arbitrary supersingular starting vertices. Instead, it starts from a CM terminal that has already been identified by the lookup table, and illustrates how the stored discriminant information is used to construct a connection to a fixed subfield reference vertex.

We take $p = 5 \cdot 2^{248} - 1$. This prime satisfies $p \equiv 3 \pmod 4$, so the vertex with $j = 1728$ is supersingular over $\mathbb{F}_p$. Moreover, the vertex $j = 1728$ has CM by the order of discriminant -4, and hence serves as a fixed prime-field reference vertex with known quaternionic data.

Let

$$j = 321755368724400922312175200230206817474673236470975409057566552617626952015 + 758885926305201454094005635472041956657807889239017712313831904835035336809\beta \in \mathbb{F}_{p^2},$$

where $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$. This vertex is a root of the reduced Hilbert class polynomial $H_D(X) \bmod p$ for $D = -503$.

From the same discriminant $D = -503$, we construct the reference quaternionic data as described in Section 4.2. Namely, we construct an embedding $\mathcal{O}_D \hookrightarrow B_{p,\infty}$ and extend its image to a maximal order $\mathcal{R}_D \subset B_{p,\infty}$ containing an optimal embedding of $\mathcal{O}_D$. The corresponding reference CM vertex is

$$j_D = 189642809373769484737479351137843655825507757613178943084598358395863420739 6 \in \mathbb{F}_p.$$

By construction, $H_D(j_D) \equiv 0 \pmod p$. Thus j and j_D are two CM vertices associated with the same fundamental discriminant $D = -503$, although they are not equal. In this example, the connected same-discriminant connection from j to j_D has total degree 7.

After this same-discriminant segment has been connected, the remaining task is to connect the reference CM vertex j_D to the fixed prime-field reference vertex with $j = 1728$. This becomes a connecting-ideal problem between two known maximal orders. In this example, the KLPT algorithm returns an isogeny of total degree 2^{992}. Thus the connection in this example has the form

$$j \overset{\deg 7}{\rightsquigarrow} j_D \overset{\text{KLPT},\deg 2^{992}}{\rightsquigarrow} (j = 1728).$$

6 Implementation Results

This section reports implementation results for the CM-enhanced search framework. The experiments are designed to evaluate two aspects of the construction. First, we measure the overhead of membership queries in the precomputed CM terminal table. Second, we compare the search phase of the original SuperSolver framework with the CM-enhanced variant, in order to evaluate the effect of enlarging the terminal set from S_p to $S_p \cup S_{\mathrm{CM}}(M)$. We focus on quantities that directly correspond to the complexity discussion in Section 4.3. These include the number of visited vertices, the number of field operations, and the cost of membership queries in the CM table. Wall-clock time is reported only when it is useful for comparison; the main implementation-independent measurements are visited vertices and field-operation counts.

6.1 Experimental Setup

Our implementation builds on two open-source codebases. SuperSolver was written by Corte-Real Santos, Costello, and Shi [12]; we also use the ApresSQI implementation [33]. We extend their routines for supersingular isogeny walks, quaternionic computations, and KLPT algorithm by adding the CM terminal table construction and membership tests.

All experiments were implemented in SageMath [32] and Python [28]. The experiments were run in a Conda environment under Ubuntu 24.04 LTS in a virtual machine, on a host machine equipped with an Intel Core i7-11800H CPU at 2.3 GHz and 16 GB of RAM. The SageMath and Python versions were 10.7 and 3.12, respectively. Moreover, we call SageMath's built-in routines for Hilbert class polynomials and modular polynomials directly. The finite field $\mathbb{F}_{p^2}$ is represented as $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$, where $\beta^2 = c$ for a fixed nonsquare $c \in \mathbb{F}_p$. Elements of $\mathbb{F}_{p^2}$ are represented as $a + b\beta, a, b \in \mathbb{F}_p$.

For the search-stage experiments, we use five small experimental parameters, chosen in the form $p = f \cdot 2^n - c$ with $p \equiv 3 \pmod 4$. Their approximate bitlengths and the corresponding numbers of independent instances are

bitlength	30	35	40	45	50
instances	2000	1000	500	100	50

respectively. Here an instance means one independently generated search trial, not a repeated timing of a fixed input: a fresh pseudo-random supersingular starting vertex and fresh random branch choices are sampled using the same method as in SuperSolver. The five-bit progression reaches the largest parameter size for which we can run many trials, and the decreasing number of instances keeps the total running time manageable. For consistency with the Delfs–Galbraith and SuperSolver settings, the maximum search depth is kept proportional to $\log p$ throughout.

We measure the effect of the CM enhancement inside the optimized SuperSolver implementation. In this experiment, we keep the optimized odd-degree choices used in the SuperSolver and add the CM-table membership test as an additional stopping condition. For each instance, we record the number of visited vertices, the lookup cost, and the number of $\mathbb{F}_p$-multiplications. The multiplication count is included specifically to allow a direct comparison with the SuperSolver, whose arithmetic costs are reported in $\mathbb{F}_p$-multiplications.

6.2 Lookup Overhead

The first experiment measures the overhead of membership queries in the precomputed CM terminal table. The table stores normalized representations of CM vertices in $S_{\mathrm{CM}}(M)$. For each tested parameter set, we compare the average cost of a membership query with the cost of one $\mathbb{F}_{p^2}$-multiplication.

This experiment is intended to justify treating the CM table lookup as a stable per-visit overhead in the search-phase complexity estimate. We also record the number of stored CM vertices and the size of the serialized table.

For the lookup-time measurements, Turbo Boost and Hyper-Threading were disabled, and the benchmarks were restricted to a single core. This reduces timing variability and makes the lookup-cost measurements more stable.

In this benchmark, all CM terminal tables are generated with $M = 2^{18}$. We measure the lookup overhead for three parameter sizes corresponding to the NIST-I, NIST-III, and NIST-V parameter levels [34]. For each parameter, we perform 10^6 membership queries on randomly sampled elements of $\mathbb{F}_{p^2}$ and report the average query time.

Table 2. Lookup overhead for CM terminal tables.

Parameter	Table size	Lookup cost (μs)	$\mathbb{F}_p$-mul. cost (μs)
NIST I, $p = 5 \cdot 2^{248} - 1$	485.20 MB	18.48	1.25
NIST III, $p = 65 \cdot 2^{376} - 1$	714.57 MB	19.65	1.35
NIST V, $p = 27 \cdot 2^{500} - 1$	936.44 MB	20.05	1.46

Table 2 reports the average membership-query time for the precomputed CM terminal tables. Across the three parameter sizes, the lookup cost remains between 18.48 μs and 20.05 μs, while the table size increases from 485.20 MB to 936.44 MB (very close to the theoretic bound $\frac{c_0}{\pi^3} M^{3/2} \cdot 2 \log p$). This indicates that the hash-table based membership test has stable performance at the scale of our experiments.

The corresponding lookup-to-multiplication ratios are approximately 14.78, 14.56, and 13.73, respectively. Thus one CM table lookup costs about 14–15 $\mathbb{F}_p$-multiplications in these measurements. Although this overhead is not negligible compared with a single field multiplication, it remains modest compared with a full SuperSolver search step, which includes neighbour generation, finite-field arithmetic, and subfield-related tests.

These measurements support the complexity model used above, in which each CM-table membership test is treated as an expected $O(1)$ in-memory hash-table lookup under the standard hashing assumption [10, Ch. 11]. They also show that the dominant cost of the CM enhancement lies in preprocessing and storing the CM terminal table, rather than in membership queries.

6.3 Search Experiments

We report two search-stage experiments. The first isolates the contribution of the CM terminal set by adding $S_{\mathrm{CM}}(M)$ to a Delfs–Galbraith 2-isogeny walk. The second integrates the corresponding CM terminal table into the optimized SuperSolver implementation, where the original odd-degree probing strategy is retained. The former focuses on CM hit rates and visited vertices, while the latter measures the implementation-level effect through visited vertices and $\mathbb{F}_p$-multiplication counts.

Delfs–Galbraith algorithm with CM terminals We first isolate the effect of enlarging the terminal set in the Delfs–Galbraith search. In this experiment, the search runs a Delfs–Galbraith non-backtracking walk in the 2-isogeny graph and stops when it reaches either S_p or $S_{\mathrm{CM}}(M)$. The odd-degree neighbour tests used in SuperSolver are disabled in this experiment.

For each prime p, we run the experiment with two effective CM terminal set sizes. In the implementation, the table stores one canonical representative for each Frobenius-conjugate pair. Thus a table containing about $\frac{1}{4}|S_p|$ and $\frac{1}{2}|S_p|$ canonical representatives corresponds to an effective CM terminal coverage of about $|S_{\mathrm{CM}}(M)| \approx \frac{1}{2}|S_p|$ and $|S_{\mathrm{CM}}(M)| \approx |S_p|$, respectively. For each run, we record the number of searches ending at a CM terminal and the number of searches ending at a subfield terminal. The CM hit rate is defined as the number of CM hits divided by the total number of search instances. It measures how often the additional CM terminal set, rather than the subfield set S_p, is responsible for stopping the search.

Table 3. CM hit rates for Delfs–Galbraith search with CM terminals.

p	$\lvert S_p\rvert$	$\lvert S_{\mathrm{CM}}(M)\rvert \approx \frac{1}{2}\lvert S_p\rvert$	$\lvert S_{\mathrm{CM}}(M)\rvert \approx \lvert S_p\rvert$
$2^{31}-1$	19865	41.8%	63.2%
$5\cdot 2^{32}-1$	96145	45.4%	63.4%
$17\cdot 2^{36}-1$	625541	44.2%	62.4%
$19\cdot 2^{41}-1$	3409293	37%	54%
$5\cdot 2^{48}-1$	21509945	64%	80%

Table 3 reports the CM hit rates for the Delfs–Galbraith search augmented with CM terminals. The results show a clear increase in CM stopping events as the effective size of the CM terminal set is increased. When $|S_{\mathrm{CM}}(M)| \approx \frac{1}{2}|S_p|$, the CM hit rate is between 37.0% and 45.4% for the first four parameter sets. When the CM terminal set is increased to $|S_{\mathrm{CM}}(M)| \approx |S_p|$, the hit rate rises to between 54.0% and 63.4% for the same four parameter sets.

These values are consistent with the intended effect of the CM terminal set: it provides an additional stopping source whose contribution is comparable to that of the subfield terminal set when the two sets have comparable effective size. The 50-bit experiment shows higher hit rates, namely 64% and 80%, but this parameter set uses only 50 independent instances, so the sampling fluctuation is larger. We therefore interpret this row as supportive but less stable than the smaller-parameter rows.

Overall, the data show that CM terminals account for a substantial fraction of the stopping events even before the SuperSolver odd-degree probing tests are enabled. This confirms that the CM terminal set acts as a meaningful additional stopping source in the Delfs–Galbraith search.

CM-enhanced SuperSolver search In this experiment, we evaluate the effect of adding the precomputed CM terminal table to the SuperSolver search. The

CM table is used as an additional stopping condition, while the original odd-degree isogeny tests of SuperSolver are kept unchanged. For each parameter set, the CM terminal set is chosen so that $|S_{\mathrm{CM}}(M)| \approx |S_p|$. Thus this experiment measures the effect of the CM enhancement inside the optimized SuperSolver implementation, rather than the isolated contribution of the CM terminal set alone.

For each parameter set, the starting vertices are generated using the same sampling routine as in the SuperSolver implementation. The SuperSolver and the CM-enhanced variant are then run on the same type of input instances.

The metrics reported in Table 4 are as follows. The column "CM-Hit" gives the fraction of searches that terminate at a CM terminal in the CM-enhanced SuperSolver run. The column "Fast ℓ's" records the auxiliary odd primes selected by the SuperSolver implementation for fast subfield-neighbour detection. The columns "SS nodes" and "SS mults" report, respectively, the average number of visited vertices and the average number of $\mathbb{F}_p$-multiplications for the SuperSolver search. The columns "CM-SS nodes" and "CM-SS mults" report the same quantities after adding the CM terminal set as an extra stopping condition.

Following the cost accounting used in SuperSolver, we measure arithmetic cost in $\mathbb{F}_p$-multiplications. In this count, one squaring in $\mathbb{F}_p$ is counted as one multiplication, while field additions are ignored. The multiplication counts therefore provide a direct implementation-level comparison between the SuperSolver and CM-enhanced searches.

The number of independent search instances is decreased as the prime size grows to keep the total runtime manageable. Specifically, for bitlengths approximately 30, 35, 40, 45, 50, we run 2000, 1000, 500, 100, 50 instances, respectively. The results are summarized in Table 4.

Table 4. Search-stage comparison between SuperSolver and CM-enhanced SuperSolver.

p	CM-Hit	Fast ℓ's	SS nodes	SS mults	CM-SS nodes	CM-SS mults
$2^{31}-1$	18.95%	[3, 5]	26744	801140	22135	663068
$5\cdot 2^{32}-1$	24.9%	[3, 5]	65859	2038803	50721	1570170
$17\cdot 2^{36}-1$	16%	[3, 5, 7]	527837	17220577	439451	14337006
$19\cdot 2^{41}-1$	13%	[3, 5, 7]	3730842	125822468	3360424	113330131
$5\cdot 2^{48}-1$	26%	[3, 5, 7]	20261898	705620390	12860161	447854784

Table 4 shows that adding the CM terminal set consistently reduces the average number of visited vertices in the SuperSolver search. For the five tested parameter sets, the reductions are approximately $17.2\%, 23.0\%, 16.7\%, 9.9\%, 36.5\%$, respectively. The $\mathbb{F}_p$-multiplication counts show essentially the same reduction ratios. This is expected because, in both variants, the dominant per-vertex computations, including the 2-isogeny and modular-polynomial operations, are the same. Hence the multiplication count is nearly linear in the number of visited vertices.

The CM-hit column records the fraction of CM-enhanced SuperSolver runs that stop at a CM terminal. The observed CM-hit rates range from 13% to 26%. This shows that, even after SuperSolver's optimized odd-degree probing is enabled, the CM terminal set still contributes a non-negligible additional stopping source. In particular, the 50-bit parameter set has the largest CM-hit rate in the table, 26%, and also shows the largest reduction in both visited vertices and multiplication counts.

These data support the interpretation of the CM terminal set as a complementary stopping mechanism for SuperSolver. It does not replace the original subfield-neighbour tests; rather, it adds another efficiently recognizable terminal source and thereby shortens the search in the tested parameter ranges.

6.4 Summary of Measurements

The measurements above support the two implementation assumptions behind the CM-enhanced method. First, the CM table membership test behaves as a stable per-visit operation: at SQIsign-size parameter levels, the lookup time remains around $20\,\mu$s, despite the increase in table size. This supports the use of an expected $O(1)$ lookup cost in the search-stage model.

Second, the CM terminal set provides an effective additional stopping source. In the experiment isolating the stopping rule, CM terminals account for a substantial fraction of stopping events once their effective size is comparable to S_p. When the same table is integrated into SuperSolver, the CM-enhanced variant reduces both the average number of visited vertices and the corresponding $\mathbb{F}_p$-multiplication counts across all tested parameter sets.

Overall, the experiments indicate that the CM terminal set acts as a complementary stopping mechanism rather than a replacement for the SuperSolver subfield tests. Its search overhead is modest and stable, while the main practical cost lies in the preprocessing and storage required to construct large CM terminal tables.

7 Discussion

The CM-enhanced stopping rule should be viewed as a complementary mechanism, rather than as a replacement for existing search strategies. It does not modify the underlying non-backtracking walk in the 2-isogeny graph G_2, nor does it remove the subfield tests used in Delfs–Galbraith or the optimized odd-degree probing tests used in SuperSolver. Instead, it adds an additional input-independent source of recognizable terminal vertices, namely the precomputed CM terminal set $S_{\mathrm{CM}}(M)$.

It is useful to distinguish the CM terminal set from a generic meet-in-the-middle time–space tradeoff. A meet-in-the-middle method builds an instance-dependent collision table from one input vertex, stores the leaves together with their paths, and then searches from the other input vertex until a collision is found. If N vertices are stored, the expected cost is $\widetilde{O}\left(N + \frac{|S_{p^2}|}{N}\right)$. Balancing the

two terms with $N \approx |S_{p^2}|^{1/2} \approx p^{1/2}$ gives the usual $\widetilde{O}(p^{1/2})$-time and $\widetilde{O}(p^{1/2})$-space tradeoff. The CM terminal set has a different role. It is not built from one of the two input vertices, and it does not store paths from a specific endpoint. Instead, it depends only on p and the discriminant bound M, and can be reused across many independent search instances. For a single instance with the same amount of storage, a generic meet-in-the-middle table can be more direct, since a collision immediately provides a path through the stored tree. The purpose of $S_{\mathrm{CM}}(M)$ is therefore not to replace meet-in-the-middle search, but to provide a reusable arithmetic stopping source that can be combined with Delfs–Galbraith, SuperSolver, or other search procedures.

The main practical limitation is the size of the computable CM terminal set. The raw CM supply grows heuristically as $R_{\mathrm{CM}}(M) = \Theta(M^{3/2})$. Thus, in order to make $|S_{\mathrm{CM}}(M)|$ comparable to $|S_p| = \Theta(p^{1/2})$, one would need roughly $M \approx p^{1/3}$. For cryptographic-size primes, this can be far beyond the range that is currently practical for large-scale class-polynomial precomputation and storage.

In our current implementation and computational environment, CM data can be generated up to discriminants of size about 2^{20}. This is sufficient for the lookup benchmarks and the small search experiments reported above, but it is far from the range required to make $S_{\mathrm{CM}}(M)$ comparable to S_p at large cryptographic parameters. Although the computable CM range is limited, the CM perspective still suggests a useful sanity check for concrete cryptographic parameters. For schemes such as SQIsign, one should check whether the selected supersingular curves lie in a low-discriminant CM family; equivalently, whether their j-invariants occur as roots of $H_D(X)$ mod p for small fundamental discriminants D. Passing such a test does not constitute a security proof, but it rules out one class of special CM-derived curves that may be easier to recognize or connect using the methods studied here.

This point should be interpreted carefully. Our results do not imply an immediate change in the bit-security of SQIsign-size parameters. Rather, they indicate that low-discriminant CM structure is a concrete feature that can be tested during parameter generation and concrete security evaluation.

Several directions remain open. First, the construction of $S_{\mathrm{CM}}(M)$ and the storage of the CM terminal table could be optimized, for example by using compressed representations and memory-efficient hashing. Second, a conductor-aware extension to non-maximal quadratic orders could also be implemented and benchmarked, including the corresponding horizontal and vertical isogeny steps. Third, the CM table could be combined with other search strategies, including meet-in-the-middle methods, or local terminal tests.

Overall, the CM-enhanced method should be understood as a reusable stopping mechanism. Its value is not in replacing generic time–space tradeoffs, but in showing that CM supersingular vertices can be organized into a precomputed terminal set and used to complement existing subfield-search algorithms.

References

1. Apostol, T.M.: Introduction to Analytic Number Theory. Undergraduate Texts in Mathematics, Springer (1976)
2. Bach, E.: Explicit bounds for primality testing and related problems. Mathematics of Computation **55**(191), 355–380 (1990)
3. Bröker, R.: Constructing supersingular elliptic curves. J. Comb. Number Theory **1**(3), 269–273 (2009)
4. Bröker, R., Lauter, K., Sutherland, A.: Modular polynomials via isogeny volcanoes. Mathematics of Computation **81**(278), 1201–1231 (2012)
5. Brown, R.: The natural density of some sets of r-free numbers. Applied Mathematics E-Notes **23**, 309–315 (2023)
6. Buchmann, J.A.: A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In: Goldstein, C. (ed.) Séminaire de Théorie des Nombres, Paris 1988–89, Progress in Mathematics, vol. 91, pp. 27–41. Birkhäuser Boston, Boston, MA (1990)
7. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of CRYPTOLOGY **22**(1), 93–113 (2009). https://doi.org/10.1007/S00145-007-9002-X
8. Chen, M., Xue, J.: On $\mathbb{F}_p$-roots of the hilbert class polynomial modulo p. J. of Math **42**(2) (2022)
9. Cohen, H.: A course in computational algebraic number theory. Springer Science & Business Media (2013)
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press (2022)
11. Corte-Real Santos, M., Costello, C., Shi, J.: Accelerating the delfs–galbraith algorithm with fast subfield root detection. In: Annual International Cryptology Conference. pp. 285–314. Springer (2022). https://doi.org/10.1007/978-3-031-15982-4_10
12. Costello, C., et al.: SuperSolver. https://github.com/microsoft/SuperSolver (2023), gitHub repository
13. Cox, D.A.: Primes of the Form x^2+ny^2: Fermat, Class Field Theory, and Complex Multiplication. with Solutions, vol. 387. American Mathematical Soc. (2022)
14. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: International conference on the theory and application of cryptology and information security. pp. 64–93. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_3
15. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Designs, Codes and Cryptography **78**(2), 425–440 (2016). https://doi.org/10.1007/s10623-014-0010-1
16. Deuring, M.: Die typen der multiplikatorenringe elliptischer funktionenkörper. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg. vol. 14, pp. 197–272. Springer Berlin/Heidelberg (1941)
17. Granville, A., Soundararajan, K.: The distribution of values of $L(1, \chi_d)$. Geometric and Functional Analysis **13**(5), 992–1028 (2003). https://doi.org/10.1007/s00039-003-0438-3
18. Iwaniec, H., Kowalski, E.: Analytic number theory, vol. 53. American Mathematical Soc. (2021)
19. Kaneko, M.: Supersingular j-invariants as singular moduli mod p. Osaka J. Math **26**, 849–855 (1989)

20. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ-isogeny path problem. LMS Journal of Computation and Mathematics **17**(A), 418–432 (2014). https://doi.org/10.1112/S1461157014000151
21. Kohel, D.R.: Endomorphism rings of elliptic curves over finite fields. University of California, Berkeley (1996)
22. Leroux, A.: Computation of hilbert class polynomials and modular polynomials from supersingular elliptic curves. arXiv preprint arXiv:2301.08531 (2023)
23. Love, J., Boneh, D.: Supersingular curves with small noninteger endomorphisms. Open Book Series **4**(1), 7–22 (2020). https://doi.org/10.2140/obs.2020.4.7
24. Mestre, J.F.: La méthode des graphes. exemples et applications. In: Yamamoto, Y., Yokoi, H. (eds.) Proceedings of the International Conference on Class Numbers and Fundamental Units of Algebraic Number Fields (Katata, Japan, June 24–28, 1986), pp. 217–242. Department of Mathematics, Nagoya University, Nagoya (1986)
25. Mula, M., Murru, N., Pintore, F.: On random sampling of supersingular elliptic curves. Annali di Matematica Pura ed Applicata (1923-) **204**(3), 1293–1335 (2025). https://doi.org/10.1007/s10231-024-01528-x
26. Onuki, H.: On oriented supersingular elliptic curves. Finite Fields and Their Applications **69**, 101777 (2021). https://doi.org/10.1016/j.ffa.2020.101777
27. Pizer, A.K.: Ramanujan graphs and hecke operators. Bulletin of the American Mathematical Society **23**(1), 127–137 (1990). https://doi.org/S0273-0979-1990-15918-X
28. Python Software Foundation: Python Programming Language (2023), version 3.12, https://www.python.org/
29. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009). https://doi.org/10.1007/978-0-387-09494-6
30. Simon, D.: Solving norm equations in relative number fields using S-units. Mathematics of Computation **71**(239), 1287–1305 (2002)
31. Sutherland, A.: Computing hilbert class polynomials with the chinese remainder theorem. Mathematics of Computation **80**(273), 501–538 (2011)
32. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.7). https://www.sagemath.org (2025)
33. The SICQ Team: ApresSQI. https://github.com/TheSICQ/ApresSQI (2023), gitHub repository
34. The SQIsign Team: SQIsign: Algorithm specifications and supporting documentation. Version 2.0 (2025), available at https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/sqisign-spec-round2-web.pdf
35. Voight, J.: Quaternion algebras. Springer (2021)

A Detailed paths for the small-parameter example

This appendix records the vertex-by-vertex paths used in Section 5.1. All vertices are written in the form

$$a + b\beta \in \mathbb{F}_{p^2} = \mathbb{F}_p(\beta), \qquad a, b \in \mathbb{F}_p,$$

with $p = 2^{19} - 1$. Each edge label denotes the degree of the corresponding isogeny.

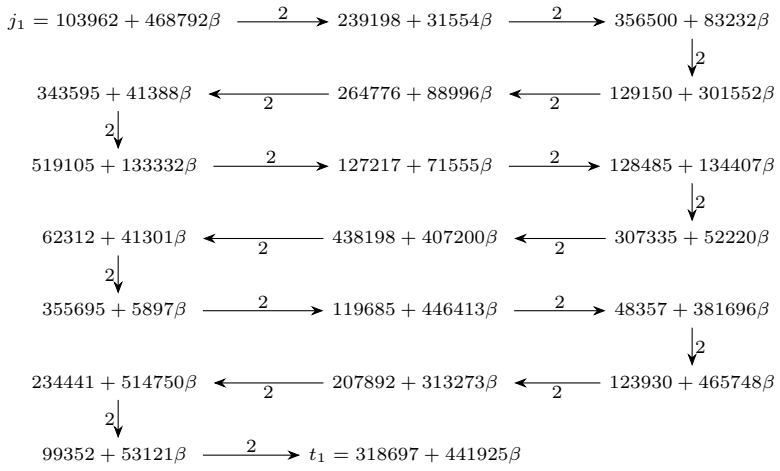


Fig. 1. The 2-isogeny search path in the toy example from $j_1 = 103962 + 468792\beta$ to the CM terminal $t_1 = 318697 + 441925\beta$. The lookup table returns the discriminant $D = -323$.

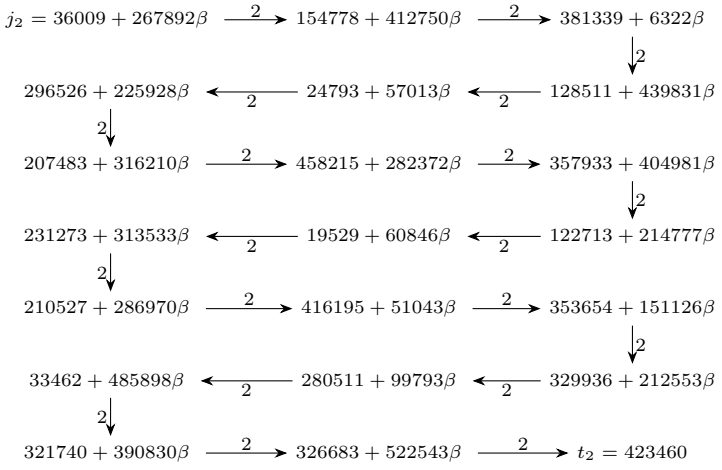


Fig. 2. The 2-isogeny search path in the toy example from $j_2 = 36009 + 267892\beta$ to the subfield terminal $t_2 = 423460 \in S_p$.

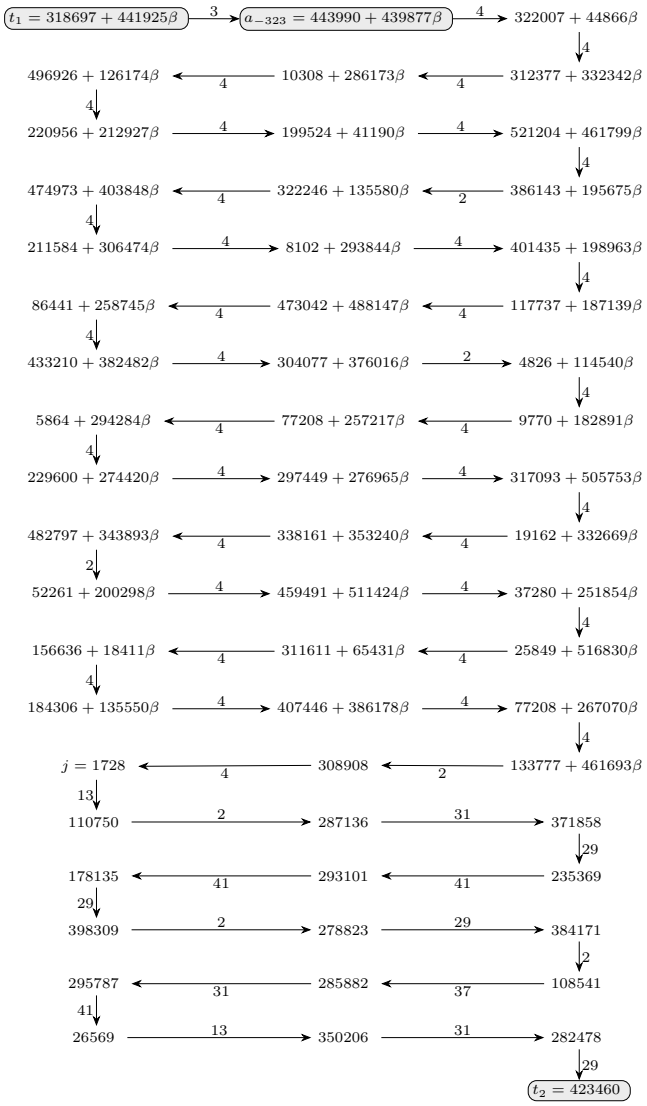


Fig. 3. Recovered mixed CM–subfield terminal path in the toy example. The CM terminal $t_1 = 318697 + 441925\beta$, tagged by the discriminant $D = -323$, is connected to the subfield terminal $t_2 = 423460$ through the fixed prime-field anchor $j = 1728$. Edge labels denote the degrees of the corresponding isogenies.