Experimentally studying path-finding problem between conjugates in supersingular isogeny graphs: Optimizing primes and powers to speed-up cycle finding

Madhurima Mukhopadhyay

Department of Mathematics, Indian Institute of Technology, Madras
ma24r008@smail.iitm.ac.in *

Abstract

We study the problem of finding a path between conjugate supersingular elliptic curves over $\mathbb{F}_{p^2}$ for a prime p, which is important for cycle finding in supersingular isogeny graphs. We see that for any given p, there is some l corresponding to p which accelerates the process of conjugate path-finding. Also, time-wise, the most efficient way of overviewing the graph is seeing $i(=3)$ steps at once. We have outlined methods in which the next vertex of any pseudo-random walk should be chosen to reach conjugate vertex faster. We have experimentally investigated the paths between frobenius conjugates for wide ranges of small prime l. We introduce sets to experimentally learn about the structure of the isogeny graphs when short cycles are present.

1 Introduction

The public key cryptosystems used today are based on integer factoring or the discrete logarithm problem in finite fields or elliptic curves. The presence of a quantum computing facility of a few thousand logical qubits would probably suffice to break these systems [RNSL17], given Shor's algorithm [Sho99]. Post-quantum cryptography is the study of cryptosystems that remain secure even if a quantum computer is available.

Let p be a large prime defining a finite field $\mathbb{F}_p$. The supersingular isogeny graph $\chi_l(\overline{\mathbb{F}_p})$ for a prime $l(\neq p)$ is a directed graph with the vertices being the isomorphic classes of supersingular elliptic curves, and the directed edges being equivalent l isogenies where both the supersingular elliptic curves and isogenies are defined over $\overline{\mathbb{F}_p}$. Isogeny based cryptography is a candidate of post-quantum cryptography, and it relies on the hardness of various [EHL+20, DFG19, BKV19, DFKL+20, CLM+18] types of path-finding problem in supersingular isogeny graphs between same or different base and final curves.

Endomorphisms being isogenies between same supersingular elliptic curves can be perceived as cycles within the graph $\chi_l(\overline{\mathbb{F}_p})$. The problem of finding paths in these graphs is equivalent to computing endomorphism rings [EHL+18] of supersingular elliptic curves which can be again reduced to computing one endomorphism [PW24, MW23].

*Personal email ID: mukhopadhyaymadhurima@gmail.com

Time complexity improvements in the path-finding problem are important from the viewpoint of choosing better cryptographic parameters [Ber20, CRSCS22, BDF+24, BBC+21]. Computational observations [ACNL+23] reveal that for $l = 2, 3$, vertices are generally closer to their frobenius conjugates in $\chi_l(\overline{\mathbb{F}_p})$, in the sense that if one performs a pseudo-random walk, lesser steps will be required when the base and final curves are frobenius conjugates of each other compared to the case when they are two arbitrary supersingular elliptic curves. The path-finding problem between conjugates is the problem of finding a path between a given supersingular elliptic curve E and it's frobenius conjugate curve $E^{(p)}$.
Paths between conjugate supersingular elliptic curves are used to construct cycles which aid in computing endomorphism rings [EHL+20]. Relying on efficient computation of cycles, Kohel [Koh96] first studied the problem of finding endomorphism rings, which was later continued by Galbraith *et. al.* [GPS20] along a different line. The runtime of Kohel's algorithm [Koh96], which finds subring of finite index in the endomorphism ring is $O(p^{1+\epsilon})$ while [GPS20] calls cycle finding algorithm $O(\log p)$ times. The algorithm of Eisenträger *et. al.*, [EHL+20] is probabilistic and depends on certain heuristics and constructs two cycles to recover endomorphism ring from a Bass suborder in time $O((\log p)^2 p^{\frac{1}{2}})$. An essential component of the above algorithms is that they are all based on finding cycles. Recent endomorphism ring computation algorithms are also available [FIK+23, ES24, XZQ24]. Despite these newer algorithms which remove some conditions in [EHL+20], the algorithm in it remains relevant due to optimized complexity, especially as p increases.
The cycle computation problem is also important as an algorithmic number theory problem and it's hardness can be utilized for devising secure cryptographic schemes. Important applications beside endomorphism ring computation [EHL+20] are in constructing collision resisting hash functions [CLG09], SQIsign [DFKL+20], etc.

1.1 Motivation and our contributions

In this section, we list our contributions(which we have itemized) and the motivations behind them. We have referred to some experiments. Our code is available at:

https://github.com/Madhurima11/cycle-computation-practical-methods

As conjugate elliptic curves are equal in the subfield $\mathbb{F}_p$ of $\mathbb{F}_{p^2}$, a related problem is finding a supersingular elliptic curve in $\mathbb{F}_p$ from an arbitrary supersingular elliptic curve in $\mathbb{F}_{p^2}$. The improvement in [CRSCS22] showed that choosing different l's for each step of the pseudo-random walk leads to less time. In our case, the value of l has to be fixed. But there lies an underlying possibility that a prime l other than $2, 3$ may lead to better timing estimates.

- We choose various primes p of sizes around 35 to 45 bits and l upto 59. Our aim was to initiate from any vertex of the graph $\chi_l(\overline{\mathbb{F}_p})$ and find a mirror path to it's frobenius conjugate. We see that the l for which the time to find a path to conjugate is minimum is not always the smaller primes like $2, 3$. It varies for each value of p. We have tabulated[Table 3] some values of p, l which correspond to each other.

- The above observations imply that for practical parameters l has to be pre-computed when just given p, supersingular elliptic curves over $\mathbb{F}_{p^2}$ and the aim is to find a mirror path. We have outlined a procedure[Section 6.2] for choosing l with the goal of minimizing the total cost.

The above discussion makes it necessary to study the nature of the path between conjugates in $\chi_l(\overline{\mathbb{F}_p})$ as l varies. Experimental data[Section 5,[ACNL+23]] is available for $l = 2, 3$.

- We have investigated [Table 1] the paths between frobenius conjugates for $l > 2, 3$. Our studies partly fulfill the requirement specified in Section 4.1 of [ACNL+23], which states that "Further studies on a broader sample of primes would be required to fully explore the differences between the distributions of distances between conjugate and arbitrary vertices."

The related sets to study the paths between conjugates is the set of all supersingular j-invariants in $\mathbb{F}_p$(denoted by S_0) and
$S_i = \{j \in \mathbb{F}_{p^2} : j$ is supersingular and is connected to it's conjugate j^p by an isogeny of degree $l^i\}$ for $i \geq 1$. The cardinality of sets S_i have been well studied for $i = 0, 1$. We note that increasing the cardinality of sets S_i [Proposition1] leads to a decrease in the length of pseudo-random walk, in our path finding problem between conjugates.

- Paths induced by j invariants in S_i may contain short cycles, depending on the value of l, p, which introduces cryptographic vulnerablities. We introduce sets $\overline{S_i} = \{j \in S_i (i > 1) | j_{\frac{i}{2}} \in S_0$ if i is even and $j_{\lfloor \frac{i}{2} \rfloor} \in S_1$ if i is odd, where j_k represents any descendant at the k-th step$\}$. Each set $\overline{S_i} = S_i$, when short cycles of corresponding lengths are absent. However, when such cycles are present, the sets $\overline{S_i}$ and S_i may not be equal for some i. It is interesting to delve into the structure of isogeny graphs in these cases. We have studied the frequency distribution[Table 1 and Table 2] of vertices in $\overline{S_i}$ and S_i for small values of i, and several primes l, p.

The path-finding between conjugates is a specialised case of the general path-finding problem between any two arbitrary supersingular elliptic curves. The connectedness of the graph $\chi_l(\overline{\mathbb{F}_p})$ implies that for large i, about $O(\log p)$ isogenies between two given supersingular elliptic curves is guaranteed [Theorem 79, [Koh96]]. But for small i, the existence of isogenies is not guaranteed. Hence a relevant problem is that, given small i, enquiring existence of an isogeny of degree l^i between two supersingular curves which are frobenius conjugates of each other. The important aspect here is that when we are performing a psuedo-random walk, increasing i at each step means we can view a larger portion of the graph. This gives the advantage of choosing a direction that would help us to land with our desired vertex quickly. On the other hand, the cost of finding isogeny grows on increasing i.

- Inspired by remark 3.5 of [EHL+20] which says we can tinker their algorithm[Algorithm 3.4 [EHL+20]] with the cases when the distances between frobenius conjugates "in the graph is bounded by some fixed integer B". Asymptotically, we prove that [Section 5.6] when solving the path finding problem between conjugates, taking $i = 3$ balances the cost of finding isogenies with the advantage of broader view. We have described optimal methods for each bound up to 3. Connecting it with the sets introduced, the relevant structure to concentrate on is $S_0 \cup S_1 \cup \overline{S_2} \cup \overline{S_3}$. We achieve the same asymptotic cost of $O(l^2 \log p)$ as in [EHL+20], which is the most optimal till now, but with the advantage of practical improvements.

Some recent works [ABC+25, KKA+24] deal with cycle computation using several prime degrees of isogenies. The bottleneck still lies in the choice of primes that would be necessary to generate cycles for constructing endomorphism rings of the general case of curves, *i.e.*, supersingular elliptic curves over $\mathbb{F}_{p^2}$. We stick to one prime l and the necessary pre-computations that would minimize the time.

1.2 Paper organization

We introduce the context of the problem in Section 1. We specifically point out our contributions in subsection 1.1. In Section 2, we give some background regarding elliptic curves and isogenies. In Section 3, we present the specific problem of path finding to conjugates and the motivation to work with it. In Section 4, we analyse the relative cardinalities of some sets which arise when short cycles are present. We calculate the cost of finding prime power isogenies in Section 5, where we also devise the optimal way in which we can do the test for finding path to conjugate. We outline the importance of choosing l beforehand and the pre-computation strategies to do so in Section 6. We list some future works in Section 7.

2 Preliminaries

We present a brief exposure to understand this work. A detailed background on elliptic curves, isogenies and related information is available in literature [Sil86, Was08].

Elliptic curve: Let $p > 3$ be a prime. An elliptic curve E over a finite field $\mathbb{F}_{p^n}$ (for some positive integer n) can be given by a (short) Weierstrass form $E(\mathbb{F}_{p^n}) : y^2 = x^3 + ax + b$ where $a, b \in \mathbb{F}_{p^n}$. The points on E are the points $x, y \in \mathbb{F}_{p^n}$ satisfying the equation of E along with a point $(0 : 1 : 0)$ on the projective curve $y^2z = x^3 + axz^2 + bz^3$, which form an abelian group [Chapter III,[Sil09]]. The discriminant $\Delta(E) = -16(4a^3 + 27b^2) \neq 0$ and j-invariant is defined by $j(E) = -1728\frac{(4a)^3}{\Delta(E)}$. For any arbitrary $j_0 \in \mathbb{F}_{p^n}$, there exists an elliptic curve E_0 over $\mathbb{F}_{p^n}$ such that $j(E_0) = j_0$. Two elliptic curves are isomorphic over $\overline{\mathbb{F}_p}$ if and only if they have the same j-invariant. The cardinality of E is given by $E(\mathbb{F}_{p^n}) = p^n + 1 - t$ where t is the trace of the p^n-th frobenius map[Chapter V, [Sil09]]. A curve $E(\mathbb{F}_{p^n})$ is supersingular if the characteristic p divides the trace t, otherwise, it is ordinary[1]. Every supersingular elliptic curve over $\overline{\mathbb{F}_p}$ has it's j-invariant defined over $\mathbb{F}_{p^2}$ [Theorem 3.1, Chapter V, [Sil09]][2]. There are $\frac{p}{12}$ of isomorphism classes of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.

Isogenies: An *isogeny* $\phi : E_1 \to E_2$ between two elliptic curves E_1, E_2 is a non-constant rational map which is also a group homomorphism. This implies that an isogeny is surjective on points over algebraic closure, has a finite kernel, and can be expressed in terms of rational functions $(\frac{p_1(x)}{q_1(x)}, \frac{p_2(x)}{q_2(x)}.y)$ where $p_1, q_1, p_2, q_2 \in \mathbb{F}_{p^n}[x]$ with $\gcd(p_1, q_1) = 1$. The degree $\deg(\phi) = \max(\deg p_1, \deg q_1)$ and ϕ is *separable* if $\left(\frac{p_1(x)}{q_1(x)}\right)' \neq 0$. An example of any isogeny on any elliptic curve is the multiplication by some integer m, denoted by $[m]$ which takes any point P to the point mP obtained by m times adding P. All isogenies of some prime degree $l \neq p$ are separable, which can be described by it's kernel [Proposition 4.12, Chapter III, [Sil09]]. Isogenies with the same kernel are identified as same[3] and we can compute the isogeny from kernel by using Velu's formula [Vél71]. Given any isogeny $\phi : E_1 \to E_2$, there is an isogeny $\hat{\phi} : E_2 \to E_1$ such that $\phi \circ \hat{\phi} = \hat{\phi} \circ \phi = [\deg(\phi)]$. This map $\hat{\phi}$ is called the dual of ϕ.
An isogeny from any elliptic curve E to itself is called an *endomorphism* of E. The set of endomorphisms

[1] There are equivalent definitions in terms of Weierstrass equation, torsion points, multiplication map, endomorphism rings or zeta function.

[2] Henceforth in our discussions from now on, we will assume the elliptic curves to be defined over $\mathbb{F}_{p^2}$.

[3] We say that they are equivalent if one isogeny can be obtained from another by composing an automorphism.

defined over $\overline{\mathbb{F}_{p^n}}$ together with the zero map, form a ring under the operations of addition and composition, which is called the *endomorphism ring* of E and is denoted by $End(E)$. For a supersingular elliptic curve, the endomorphism ring is isomorphic to an order in a quaternion algebra.

Modular Polynomials: The *modular polynomial* $\phi_l(x, y)$ for a prime l is a symmetric polynomial of degree $(l + 1)$, whose roots over $\mathbb{F}_{p^2}$ correspond to pair of l-isogeneous j-invariants of elliptic curves over $\mathbb{F}_{p^2}$ [Exercise 2.18, Chapter II, [Sil94]] and is of the form $\phi_l(x, y) = x^{l+1} - x^l y^l + y^{l+1} + \sum_{k,m \leq l, k+m<2l} a_{km} x^k y^m$ for $a_{km} \in \mathbb{Z}$. It requires $O(l^3 \log l)$ bits to represent ϕ_l. Given an elliptic curve E, to compute l isogeneous curve, we evaluate $\phi_l(j(E), x) \in \mathbb{F}_{p^n}[x]$ and then compute roots in algebraic closure. An algorithm due to Elkies [E+98] allows to compute the isogeny in time exponential in l.

Isogeny graphs: We have already defined the supersingular isogeny graph $\chi_l(\overline{\mathbb{F}_p})$ in Section 1. The number of edges from any j-invariant is the $(l + 1)$ of roots of the modular polynomial evaluated at that point. The graph can be a multi-graph and for each edge $(j(E_1), j(E_2))$, the concept of dual isogeny implies that $(j(E_2), j(E_1))$ is also an edge. When $j(E_1), j(E_2) \neq 0, 1728$, the multiplicities of $(j(E_1), j(E_2))$ and $(j(E_2), j(E_1))$ are same, which means barring these two vertices, we can view it as an undirected graph[4]. This graph is connected, $(l + 1)$-regular(except at $0, 1728$ due to extra automorphisms). Any two elliptic curves in this graph is connected by m isogenies of degree l where $m = \log p$ [Theorem 79,[Koh96]]. For a fixed prime $l = O(\log p)$, any l-isogeny in the just mentioned chain can be specified by rational maps or kernel, and so have representations of size polynomial in $\log p$. The cardinality of the set V of vertices of this graph is given by $\#V = [\frac{p}{12}] + \epsilon_p$ where $\epsilon_p = 0, 1, 2$ accordingly as $p \equiv 1, \{3, 5, 7\}, 11 \mod 12$ respectively. This graph is a Ramanujan graph [Piz90], an optimal expander graph. This means any random walk mixes rapidly, and it is widely used in endomorphism ring computation and path-finding problems.

Isogeny cycles: An isogeny cycle, which we shall refer as simply cycle from now on, is a closed walk containing the given set of vertices of elliptic curves, which contains no backtracking and is not a power of another closed walk. By backtracking we mean an edge is dual to any other[5]. The cycle $\{a_1, a_2, a_3, \ldots, a_N\}$, through some elliptic curve E, corresponds to an endomorphism $End(E)$ of degree l^N.

Conjugates: Given an elliptic curve $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^2}$, let $E^{(p)} : y^2 = x^3 + a^p x + b^p$. The Frobenius morphism $Frob : E \to E^{(p)}$ is given by $(x, y) \mapsto (x^p, y^p)$. The Galois group $Gal(\mathbb{F}_{p^2}/\mathbb{F}_p)$ of the frobenius automorphism $\pi : \mathbb{F}_{p^2} \to \mathbb{F}_{p^2}$ such that $\pi(\alpha) = \alpha^p$ acts on the graph $\chi_l(\overline{\mathbb{F}_p})$ and sends a j-invariant j_1 to j_1^p(the conjugate). The fixed points under this automorphism are the supersingular j-invariants in $\mathbb{F}_p$. Also, π is an involution as $j_1^p = j_1 \forall j_1 \in \mathbb{F}_{p^2}$. Any isogeny $\Phi : E_1 \to E_2$ such that $\Phi = (f, g)$ between two elliptic curves results in another isogeny between the conjugate curves given by $\Phi^p : E_1^p \to E_2^p$ where $\Phi^p = (f^p, g^p)$. Also, $\left(\Phi^p\right)^p = \Phi$.
The mirror involution [ACNL+23] sends each element of the graph $\chi_l(\overline{\mathbb{F}_p})$ to it's conjugate, resulting in some important structural properties. The algebraic number theoretical condition of E and E^p being l-

[4]By identifying isogenies with it's dual and edges as equivalent upto post composition with an automorphism.

[5]Two edges are dual if the dual of the isogeny corresponding to it is equal to the composition of automorphism of it with the isogeny corresponding to the other.

isogeneous for some supersingualr elliptic curve E over $\mathbb{F}_{p^2}$, is that $\mathbb{Z}[\sqrt{-lp}]$ embeds into $End(E)$ [Lemma 6,[CLG09]].

3 Path finding to conjugates in supersingular isogeny graphs

In this section we discuss the concept of mirror paths and certain sets which help to study path finding to conjugates. Next, we introduce a modified version of these sets, which are useful from the cryptographic security viewpoint.

Mirror paths[Definition 2.6 [ACNL+23]]: Cycles are build with one side the the mirror image under the frobenius map, of the other side as,

$$\phi_l(j_1, j_2) = 0 \implies \phi_l(j_1^p, j_2^p) = \phi_l(j_1, j_2)^p = 0 \tag{1}$$

This now implies that for l-isogeneous vertices j_1 and j_2, if any of them is in $\mathbb{F}_p$ or isogeneous to it's conjugate, then there exists a path between the other vertex and it's conjugate. When this event happens in a pseudo-random walk, then, the path can be traced backwards to reach the starting vertex. Mirror paths are either of the form $j_1 \to j_2 \to \dots j_m \to j \to j_m^p \to j_{m-1}^p \to j_2^p \to j_1^p$ (where the intermediate vertex j is in $\mathbb{F}_p$), which makes it of even length or it is of the form $j_1 \to j_2 \to \dots j \to j^p \to j_2^p \to j_1^p$ (where j is l-isogeneous to j^p) of odd length.

An essential subroutine of cycle finding [Algorithm 3.4,[EHL+20]] is to find a path between conjugate vertices in $\chi_l(\overline{\mathbb{F}_p})$ for fixed l. This makes it relevant to study values of l and minimum value of B(for fixed l), so that finding l^B isogeny to the conjugate j^p takes optimal time.

3.1 Length of pseudo-random walk

The length of the pseudo-random walk to find a cycle, can be connected to the cardinality of the target vertex set as below.

Proposition 1. *[Proposition 3.6, [EHL+20]] Let us consider a random walk in $\chi_l(\overline{\mathbb{F}_p})$ ($p > 3$ and $l \neq p$) to reach a set S of vertices, not containing 0 or 1728. Then the minimum length L of the random walk is*

$$L = \frac{\log\left(\frac{p}{6|S|^{\frac{1}{2}}}\right)}{\log\left(\frac{l+1}{2\sqrt{l}}\right)}.$$

The corresponding probability Pr_S to reach S is

$$Pr_S \geq \frac{6|S|}{p}$$

3.2 Cardinality of the set S of target vertices for various choices of isogeny degree

The set S can be chosen as the set of all supersingular j-invariants in the subfield $\mathbb{F}_p$. In that case, the cardinality of S is $O(p^{\frac{1}{2}})$, where the actual value [Equation 1, [DG16]] can be calculated from the class number of some associated imaginary quadratic field and the value of p modulo 8.

The above value is independent of the choice of the prime l, as it does not deal with any isogeneous curve. The other choice of the set S arises when l^i isogeny to the conjugate curve is considered for various values of the integer $i \geq 1$. Let us call this set as S_i. More formally, let

$$S_i = \{j \in \mathbb{F}_{p^2} : j \text{ is supersingular and is connected to it's conjugate } j^p \text{ by an isogeny of degree } l^i\} \quad (2)$$

for $i \geq 1$. Additionally, let us define

$$S_0 = \{j \in \mathbb{F}_{p^2} : j \text{ is supersingular and } j = j^p\} = \{j \in \mathbb{F}_{p^2} : j \text{ is supersingular and } j \in \mathbb{F}_p\} \quad (3)$$

For example, [EHL+20] considers l isogeny to the conjugate. They have derived a lower bound[Theorem 3.9, [EHL+20]] on the cardinality of the corresponding set S_1 as $\#S_1 > \frac{C\sqrt{p}}{\log(\log(p))}$ where $l < \frac{p}{4}$ and $C > 0$ is a constant depending upon the value of l.
In general, asymptotically, the cardinality [[CLG09], Lemma 6] of S_i is $l^{\frac{i}{2}}O(\sqrt{p})$.

3.3 Necessity to choose l and i optimally

A vital issue here is to test membership of an element in S_i. For relatively large $i(i = O(\log p))$, the existence of isogeny for any element of $\chi_l(\overline{\mathbb{F}_p})$ is guaranteed [Theorem 79, [Koh96]]. But for smaller values of i, the isogeny of degree l^i between conjugates may not exist. Computing isogenies is the only way in this case. This cost of isogeny computation grows bigger as the value of l or i increases. The necessity of relatively larger l and i may arise as it increases the cardinality of S_i, which decreases the length of the pseudo-random walk. This motivates to study ways to choose l and i so that the total cost which depends both on the length of the pseudo-random walk and the cost of isogeny computation is optimally balanced, which has the possibility of adding practical improvements to the most optimized algorithm [EHL+20] for finding paths between conjugates.

3.4 Paths leading to mirror paths

The vertices in S_0 or S_1 give rise to mirror paths which aid in finding cycles. For $i > 1$, the paths corresponding to vertices in S_i may be mirror paths or non-mirror paths. Non-mirror paths may happen when a vertex is l-isogeneoous to the conjugate of some vertex $\overline{j}$, whose immediate ancestor is a part of the pseudo-random walk, but $\overline{j}$ itself is not a part of the walk. Let us call such paths as *generalised sibling paths.*

Definition 1. *Sibling Path: Suppose j is a vertex which appears in a pseudo-random walk with l-isogeneous supersingular j-invariants (leaving the one already chosen as it's ancestor) $j_{i_1}, j_{i_2}, \ldots, j_{i_l}$. We call a path sibling path if $\exists\ i_1 \neq i_2$ such that $j_{i_1}^p$ and j_{i_2} are l-isogeneous.*

In case of sibling paths j_{i_1} and j_{i_2} have a common immediate ancestor. The notion of generalised sibling paths captures the cases when the immediate common ancestor may not be the same. Obviously, any sibling path is also a generalised sibling path.
For example, let, $j, j_{11}, j_{21}, j_{31}, j_{41}, j_{52}, j_{61}, j_{72}, j_{81}$ be a path such that j_{km}(obtained as a root of $j_{k-1,.}$) signifies a vertex at the k-th level and is the m-th of the l isogenous vertices. Following this notation, the vertex j_{32} is l-isogenous to j_{21} and it is not a part of the pseudo-random walk. Let the conjugate j_{32}^p be l isogeneous to j_{81}, *i.e.*, $\phi_l(j_{81}, j_{32}^p) = 0$. We then obtain a short cycle[6] as $j_{21}, j_{31}, j_{41}, j_{52}, j_{61}, j_{72}, j_{81}, j_{32}^p, j_{21}^p, j_{31}^p, j_{41}^p, j_{52}^p, j_{61}^p, j_{72}^p, j_{81}^p, j_{32}, j_{21}$.

[6] We express this as an isogeny with initial and terminal vertex j_{21}

Remark 1. *Any sibling path or generalised sibling path leads to a short cycle.*

Let us recall the relation between isogeny and endomorphism ring.

Theorem 2. *[Theorem 3.1 [MMP24]] Let the Generalised Riemann Hypothesis hold true. Also, let E, E' be two supersingular elliptic curves with an isogeny of degree l between E and E'. Furthermore, let $End(E)$ and $End(E')$ be the endomorphism rings of E and E' respectively.*
If $End(E)$ and $End(E')$ are known, then the l-isogeny can be computed in polynomial time.
If the l-isogeny is known, along with one of the endomorphism rings, then the other endomorphism ring can be computed in polynomial time.

The above theorem implies that if a short cycle through any vertex of $\chi_l(\overline{\mathbb{F}_p})$ is known along with an isogeny from any other vertex, it may introduce vulnerabilities, which leads to knowledge about the endomorphism ring, which again aids in computing isogenies. This disturbs the foundational hardness assumptions of isogeny-based cryptography. Hence, we are interested in studying paths in $S_i(i > 1)$, which fall in the category of non-sibling paths. This implies the study of sets $S_i(i > 1)$, such that vertex $j \in S_i$ is only characterized by the fact that it has a descendant vertex in either of S_0 and S_1. We exclude those vertices which may lead to generalised sibling paths. Since a new vertex is selected at every point of the pseudo-random walk, choosing such a vertex j earlier in the pseudo-random walk will save the cost of computing isogenies which may not lead to mirror paths or the paths which may be insecure for cryptographic purposes. Thus our focus will be to study the sets

$$\overline{S_i} = \{\ j \in S_i(i > 1) |\ j \text{ has a descendant vertex in } S_0 \text{ or } S_i\}$$

By the theory of mirror paths,

$$\overline{S_i} = \{j \in S_i(i > 1) |\ j_{\frac{i}{2}} \in S_0 \text{ if } i \text{ is even and } j_{\lfloor \frac{i}{2} \rfloor} \in S_1 \text{ if } i \text{ is odd}\}$$

[7] We thus discard elements of the set $S_i \setminus \overline{S_i}$ of those j-invariants which have descendants at any two levels(may be same also) such that one is isogeneous to the frobenius conjugate of the other. Technically,

$S_i \setminus \overline{S_i} = \{j \in S_i(i > 1) |\ j$ has no descendant j_{k_1,i_1} and j_{k_2,i_2} such that j_{k_1,i_1} is isogeneous to $j^p_{k_2,i_2}\}$

[8]

4 Experimental observations with small distance

Previous observations [ACNL+23] point out that paths between conjugate vertices are much more common than those between arbitrary vertices. In this section, we report the relative frequency of vertices in $\overline{S_2}, \overline{S_3}, \overline{S_4}$ from the experiments of performing pseudo-random walks.
To do this, we focussed on three primes p equal to $70001, 90001, 100003$ and l ranging from primes 2 to 59. The difference from the earlier [ACNL+23] observations in literature is that, we consider a larger set for primes l, instead of only $2, 3$ and deal with child nodes in $\mathbb{F}_p$ or S_1. We note that precise bounds on the estimate of cardinality of S_0, S_1 are known, whereas the same for S_i is absent in literature for $i > 1$.

[7] where j_k represents any descendant at the k-th step
[8] where j_{t,i_1} represents a descendant at the t-th step

We considered the graph $\chi_l(\overline{\mathbb{F}_p})$, where we took random supersingular j-invariants, j_0 and continued pseudo-random walks by initially choosing $j = j_0$ and then j as a random root of $\phi_l(j, x)$.
The aim was to see when can we get a hint of a mirror path to j_0^p. This could happen when some intermediate j and j^p were connected by l^i isogeny. We varied $i = 2, 3, 4$ and stopped when a desired isogeny was found. To do this, we also labeled each of the l roots (leaving out the root that has occurred previously) as $j_{i_1}, j_{i_2}, \ldots, j_{i_l}$.
We halted when there was some $i_1, i_2 \in \{1, 2, \ldots, l\}$ such that either $j_{i_1} = j_{i_2}^p$, or $\phi_l(j_{i_1}, j_{i_2}^p) = 0$ or $\deg\left(\gcd(\phi_l(j_{i_1}, x), \phi_l(x, j_{i_2}^p))\right) \geq 1$. This corresponds to the cases where $j \in S_2$ or S_3 or S_4 respectively, j being the parent of j_{i_1} and j_{i_2}. When $i_1 = i_2$, the child $j_{i_1} \in S_0$ (for $i = 2, 4$) or S_1 (for $i = 3$). These lead to mirror paths that can be utilised to construct cycles. Contrarily when $i_1 \neq i_2$, the paths lead to short cycles which are discarded from the viewpoint of cryptographic security.
In Table 1, we have noted l, p along with the cases for which we get mirror paths. In the third column, we record the total number of supersingular j invariants. For each prime and each isogeny l, we have noted the number of elements in S_i and $\overline{S_i}$ for $i = 2, 3, 4$. The figures under $S \setminus \overline{S_i}$ comes from the number of supersingular j invariants such that j_{i_1} and $j_{i_2}^p$ are isogeneous for $i_1 \neq i_2$. We see that the number of elements in this set $S_i \setminus \overline{S_i}$ which leads to cycles is quite small when compared with the total elements in S_i. This is reflected in the ratio $\frac{|S_i \setminus \overline{S_i}|}{|S_i|}$.
We have separately computed the relative cardinality of $\overline{S_i}$'s in terms of % when compared with the total number of j invariants in Table 2. This reveals the importance of checking the presence of the element in $\overline{S_i}$ for small i, as they are the dominant isogeny that finds the path to the conjugate. This practical idea of the comparison of the proportional cardinalities is new and is a step further towards studying the distances between conjugates and arbitrary vertices as was suggested in Section 4.1 of [ACNL+23].

l	p	Total supersingular j-inv.	$i=2$	Distribution in S_2		$i=3$	Distribution in S_3		$i=4$	Distribution in S_4	
			$\lvert S_2\rvert$	$\lvert S_2 \setminus \overline{S_2}\rvert$, $\frac{\lvert S_2\setminus\overline{S_2}\rvert}{\lvert S_2\rvert}$	$\lvert\overline{S_2}\rvert$	$\lvert S_3\rvert$	$\lvert S_3 \setminus \overline{S_3}\rvert$, $\frac{\lvert S_3\setminus\overline{S_3}\rvert}{\lvert S_3\rvert}$	$\lvert\overline{S_3}\rvert$	$\lvert S_4\rvert$	$\lvert S_4\setminus\overline{S_4}\rvert$, $\frac{\lvert S_4\setminus\overline{S_4}\rvert}{\lvert S_4\rvert}$	$\lvert\overline{S_4}\rvert$
2	70001	616	44	0,0	44	108	9,0.083	99	464	0,0	464
	90001	2194	55	0,0	55	1169	33,0.028	1136	970	0,0	970
	100003	4594	119	0,0	119	3323	117,0.035	3206	1152	0,0	1152
3	70001	616	42	0,0	42	288	13,0.045	275	286	0,0	286
	90001	2194	81	14,0.173	67	1122	43,0.038	1079	991	0,0	991
	100003	4594	201	41,0.204	160	2157	97,0.045	2060	2236	0,0	2236
5	70001	616	78	0,0	78	175	29,0.166	146	363	0,0	363
	90001	2194	79	0,0	79	1353	73,0.054	1280	762	0,0	762
	100003	4594	298	40,0.134	258	3257	324,0.099	2933	1039	0,0	1039
7	70001	616	122	11,0.090	111	277	28,0.101	249	217	0,0	217
	90001	2194	166	14,0.084	152	975	70,0.072	905	1053	0,0	1053
	100003	4594	391	39,0.100	352	1250	70,0.056	1180	2953	0,0	2953

	70001	616	158	0,0	158	197	20,0.083	177	261	0,0	261
11	90001	2194	217	0,0	217	1653	189,0.114	1464	324	0,0	324
	100003	4594	462	0,0	462	2336	157,0.067	2179	1796	0,0	1796
	70001	616	182	0,0	182	276	35,0.127	241	158	32,0.203	126
13	90001	2194	260	12,0.046	248	743	58,0.078	685	1191	260,0.218	931
	100003	4594	583	36,0.062	547	3276	387,0.118	2889	735	189,0.257	546
	70001	616	249	10,0.040	239	154	7,0.045	147	213	94,0.441	119
17	90001	2194	356	13,0.037	343	1417	142,0.100	1275	421	206,0.489	215
	100003	4594	642	0,0	642	3461	377,0.109	3084	491	266,0.541	225
	70001	616	259	10,0.039	249	349	49,0.140	300	8	5,0.625	3
19	90001	2194	414	12,0.029	402	1414	138,0.0.098	1276	366	281,0.768	85
	100003	4594	816	41,0.050	775	1998	151,0.076	1847	1780	1177,0.661	603
	70001	616	299	9,0.030	290	285	23,0.081	262	32	15,0.469	17
23	90001	2194	450	12,0.027	438	1673	156,0.093	1517	71	54,0.761	17
	100003	4594	903	0,0	903	3292	285,0.087	3007	399	301,0.754	98
	70001	616	303	0,0	303	291	110,0.378	181	22	9,0.41	13
29	90001	2194	527	12,0.023	515	1625	339,0.209	1286	42	36,0.857	6
	100003	4594	1153	36,0.031	1117	3327	341,0.102	2986	114	74,0.649	40
	70001	616	328	0,0	328	288	83,0.288	205	0	0,0	0
31	90001	2194	575	10,0.017	565	1617	401,0.248	1216	2	1,0.5	1
	100003	4594	1239	37,0.030	1202	3189	991,0.311	2198	166	119,0.717	47
	70001	616	382	0,0	382	234	122,0.521	112	0	0,0	0
37	90001	2194	675	14,0.021	661	1503	752,0.500	751	16	11,0.688	5
	100003	4594	1424	35, 0.025	1389	3166	1260,0.398	1906	4	4,1	0
	70001	616	422	8, 0.019	414	194	98,0.505	96	0	0,0	0
41	90001	2194	719	14,0.019	705	1475	445,0.302	1030	0	0,0	0
	100003	4594	1476	0,0	1476	3116	731,0.234	2385	2	2,1	0
	70001	616	429	7,0.016	422	187	95,0.508	92	0	0,0	0
43	90001	2194	676	0,0	676	1500	912,0.608	588	18	14,0.777	4
	100003	4594	1549	0,0	1549	3039	2145,0.705	894	6	6,1	0
	70001	616	464	6,0.012	458	152	91,0.598	61	0	0,0	0
47	90001	2194	719	0,0	719	1475	517,0.350	958	0	0,0	0
	100003	4594	1735	31,0.017	1704	2859	1075,0.376	1784	0	0,0	0
	70001	616	482	5,0.010	477	134	114,0.850	20	0	0,0	0
53	90001	2194	824	0,0	824	1370	1018,0.743	352	0	0,0	0
	100003	4594	1820	27,.014	1793	2774	1048,0.377	1726	0	0,0	0
	70001	616	496	5,.010	491	120	91,0.758	29	0	0,0	0
59	90001	2194	903	7,0.007	896	1291	577,.446	714	0	0,0	0
	100003	4594	2010	29,0.014	1981	2584	1574,0.609	1010	0	0,0	0

Table 1: Analysis of conjugate paths and detection of l^i isogenies for small i.

Table 2: Relative %'s of the set $\overline{S_i}$ for $i = 2, 3, 4$ when compared with the total number of j invariants considered.

$\overline{S_i}$	% of $\overline{S_i}$ from Table 1
$\overline{S_2}$	$7, 3, 3; 7, 3, 3; 13, 4, 6; 18, 7, 8; 26, 10, 10; 30, 11, 12; 39, 16, 14; 41, 18, 17; 48, 20, 20; 50, 24, 24;$ $54, 26, 26; 63, 30, 30; 68, 32, 32; 69, 31, 34; 75, 33, 37; 78, 38, 39; 80, 41, 43$
$\overline{S_3}$	$16, 52, 70; 45, 49, 45; 24, 58, 64; 41, 41, 26; 29, 67, 47; 40, 31, 63; 24, 58, 67; 49, 58, 40; 43, 69, 66;$ $30, 59, 65; 34, 56, 48; 18, 34, 41; 16, 47, 52; 15, 27, 19; 10, 44, 39; 3, 16, 38; 5, 33, 22$
$\overline{S_4}$	$75, 44, 25; 47, 45, 49; 60, 35, 22; 36, 48, 64; 42, 15, 39; 21, 42, 12; 20, 10, 5; 0, 4, 13; 3, 1, 2; 2, 0, 1;$ $0, 0, 1; 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0$

This once again reassures that paths between conjugate j-invariants are much more common than any other paths. Along with that, it points to the new direction that paths arising due to vertices in $\overline{S_2}, \overline{S_3}, \overline{S_4}$ are quite common and so efficient detection strategies for $i \geq 2$ would be helpful for obtaining mirror paths.

We have experimentally studied the issue of mirror paths arising from vertices in S_2, S_3, S_4 for various values of l. We see that the derivation of better theoretical estimates of the cardinality of sets $\overline{S_i}$ is important from the viewpoint of obtaining paths between conjugate vertices in the supersingular isogeny graph.
Also, this suggests searching for optimal distance and better l to find mirror paths. We have done this in Sections 5.6 and 6 respectively.

5 Cost of finding an isogeny of degree l^i

The aim of this section is to find the cost of testing whether a given vertex j is l^i isogeneous to it's conjugate j^p in each case. The intuition is that, for a fixed l, we would like to reduce the length of the pseudo-random walk and accelerate the process of arriving at a suitable vertex that would give a path to the conjugate. We would like to minimize root computations, if possible. For smaller values of i, we consider the modular polynomial while for larger i, we use minimal polynomial generated from a system of modular polynomials. We calculate corresponding costs in terms of the multiplications needed[9]. There is a trade-off between increasing the value of i and testing whether elements are in $\overline{S_i}$. We compare the methods and find which one would be best according to the computational resource of time.

5.1 $i = 0$:

This consist of the set S_0. Given a j-invariant, the cost of testing whether it is in the subfield $\mathbb{F}_p$ or not, will take constant time $O(1)$.

[9]We have neglected the costs of addition and subtraction in our analysis as they are quite less than those of multiplication.

5.2 $i = 1$:

Given a j-invariant, the cost of finding whether it is a member of S_1 or not will comprise taking into account three issues.

1. Finding the conjugate invariant j^p.
2. The cost of evaluating the modular polynomial $\phi_l(x, y)$ at $x = j, y = j^p$.
3. The cost of testing whether $\phi_l(j, j^p) = 0$ or not.

5.2.1 Finding conjugates:

Given a finite field $\mathbb{F}_{p^2}$ it is expressed as $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$, where β is a primitive element of the extension. The element β can be chosen such that it is -1 or the first non-square in the sequence $\pm n$, $n \geq 2$. This means that for any element $j = a + \beta b \in \mathbb{F}_{p^2}$, it's conjugate $j^p = a - \beta b$. This operation costs $O(1)$ so we can ignore it henceforth.

5.2.2 Evaluating the modular polynomial:

The modular polynomial $\phi_l(x, y)$ modulo prime p is of the form $x^{l+1} - x^l y^l + y^{l+1} + \sum_{k,m \leq l, k+m<2l} a_{km} x^k y^m$ where the integers a_{km} are reduced modulo p. To evaluate this polynomial at $x = j, y = j^p$, we initially need the values of $j^k, (\tilde{j})^k \ \forall k = 0, 1, \ldots, l$ where $\tilde{j} = j^p$ is already known. Once these values are available, we need to multiply several elements in the field $\mathbb{F}_{p^2}$. We note that one multiplication in $\mathbb{F}_{p^2}$ costs three multiplications in $\mathbb{F}_p$.
For computing these powers, the cost is $O(l)$ multiplications in $\mathbb{F}_{p^2}$ as each of the powers j_1^k for $j_1 = j, \tilde{j}$, can be re-used to compute the next power j_1^{k+1}. We can now consider this cost as $O(l)$ multiplications in $\mathbb{F}_p$.
Once these powers are computed we can move with the evaluation of the bivariate polynomial of degree $O(l)$ in each variable. For each $k \leq l$, the coefficient of x^k consists of terms of the form $a_0 + a_1 y^1 + \ldots + a_l y^l$. The cost of each power of x is $O(l)$ multiplications in $\mathbb{F}_p$. Now considering all the l powers of x, the entire bivariate polynomial can be evaluated in $O(l^2)$ multiplications in $\mathbb{F}_p$.

Considering all of the above, the total cost is $O(l^2)$ multiplications over $\mathbb{F}_p$.

5.2.2.1 Cost of testing whether the evaluation is zero or not

Once the polynomial is evaluated for the conjugates, the cost of knowing whether it is zero or not can be neglected as it is constant, $O(1)$.

Remark 2. *The above implies that the cost of testing whether vertices are in S_1 or not will take time complexity of $O(l^2 \log p)$.*

5.3 $i = 2$:

Given a j-invariant j, it is a member of S_2, if it is connected with it's conjugate by an isogeny of degree l^2. By the theory of mirror paths, this connection exists if there is a vertex j_s which is a member of

the subfield $\mathbb{F}_p$, and it appears as a neighbour of both j and j^p. Let $f = \phi_l, p(j, x)$. Clearly, from Section 5.2.2, the cost of getting f is $O(l^2)$ multiplications in $\mathbb{F}_p$.
From Proposition 1 of [CRSCS22], the problem reduces to finding gcd of two polynomials g_1, g_2 where,

$$g_1 = f + \pi(f),\ g_2 = f - \pi(f)$$

where π is the p-power frobenius endomorphism. We note that π fixes each element of $\mathbb{F}_p$ and $\pi(\beta)+\beta = 0$ for the primitive element β. This means that $\pi^k(\beta) = (-1)^k\beta$, which is something we shall use repeatedly now.

5.3.1 Simplifying expression of g_1 and g_2

The essential aim of Proposition 1 of [CRSCS22] is to find whether g_1, g_2 have common roots. Considering the fact that multiplications by constants do not change the roots of the polynomials, we can modify the polynomials g_1, g_2 as:

$$g_1 = \frac{1}{2}[\phi_l, p(j, x) + \pi(\phi_l, p(j, x))],\ g_2 = \frac{-\beta}{2}[\phi_l, p(j, x) - \pi(\phi_l, p(j, x))]$$

If we consider the polynomial f of the form $f = \sum_{i=0}^{n} a_i x^i$ for the coefficients a_i reduced modulo the prime p, where a_i's are elements of $\mathbb{F}_{p^2}$,

$$g_1 = \frac{1}{2}[f + \pi(f)] = Re(a_n)x^n + Re(a_{n-1})x^{n-1} + Re(a_{n-2})x^{n-2} + \ldots + Re(a_1)x^1 + Re(a_0) \quad (4)$$

$$g_2 = \frac{-\beta}{2}[f - \pi(f)] = Im(a_n)x^n + Im(a_{n-1})x^{n-1} + Im(a_{n-2})x^{n-2} + \ldots + Im(a_1)x^1 + Im(a_0) \quad (5)$$

where by Re and Im we mean, the coefficients of β^0, β^1 respectively for an element $(a + \beta b) \in \mathbb{F}_{p^2}$. Using $(a + \beta b)^p = (a - \beta b)$ and substituting $Re(a + \beta b) = a$ and $Im(a + \beta b) = b$ when each of the a_i's are expressed in terms of β, we get

$$g_1 = x^n + \ldots + Re(a_0),\ g_2 = Im(a_n)x^{n-1} + \ldots + Im(a_0) \quad (6)$$

5.3.2 Testing whether the modular polynomials have a root in $\mathbb{F}_p$

We can compute the inverse-free gcd of g_1 and g_2 by using Algorithm 1 of [CRSCS22]. The value of n is the degree of the modular polynomial, which is $(l + 1)$. So the above procedure will take time $O(n^2) = O(l^2)$ [Proposition 2, [CRSCS22]], multiplications in $\mathbb{F}_p$.
This analysis sums up the total cost of checking for an l^2 isogeny as $O(l^2)$ multiplications in $\mathbb{F}_p$.

5.4 Problems in using modular polynomials for $i = 3$ or $i = 4$ and higher without root computation

Case for $i = 3$: Given a j-invariant j, to test whether it is connected with it's frobenius conjugate j^p by an isogeny of degree l^3, we need to know whether the modular polynomials $\phi_l(j, x)$ and $\phi_l(x, x^p)$ have a common root. If there is a common root α, then this will imply a path of length 3 given by $j \to \alpha \to \alpha^p \to j^p$. But, the polynomial $\phi_l(x, x^p)$ is a polynomial of degree pl and hence the complexity to compute the gcd will be exponential in $\log p$. So using simply univariate modular polynomials is not suitable for $i = 3$.

Case for $i = 4$: In this case to know whether a mirror path exists, we need to know the intermediate roots. This is not possible without root computation.
For higher value of i, the same situations holds true and so we need several modular polynomials for these cases.

5.5 Approach for $i \geq 3$ using a system of modular polynomials

We assume we are given a fixed positive integer $i \geq 3$ and a j-invariant j. Our job is to find whether there is a small isogeny of degree l^i between j and j^p. As the paths between j and j^p are mirror paths, it is sufficient to test whether there is some intermediate j-invariant in $\mathbb{F}_p$ or S_1. We consider $d = \lfloor \frac{i}{2} \rfloor$, our aim is then to know whether a vertex j_d is in $\mathbb{F}_p$ or S_1, that satisfies the system below.

$$\begin{aligned} \phi_l(j, j_1) &= 0 \\ \phi_l(j_1, j_2) &= 0 \\ &\vdots \\ \phi_l(j_{d-1}, j_d) &= 0 \end{aligned} \tag{7}$$

5.5.1 Computing roots of modular polynomials at each level

In this case, we focus on calculating the complexity when the roots $j_1, j_2, \ldots, j_d$ of modular polynomials are computed at each level. For the given modular polynomial, we exclude the root at the previous level and compute the possible roots at the next level. This implies that for each modular polynomial, there are l options for j_1, l^2 options for j_2, and so on, with l^d options for j_d. The last job is to test whether any of the j_d's are in $\mathbb{F}_p$ or S_1.
Initially, we need to compute the root of the polynomial $\phi_l(j, x)$. Let R_l be the cost of root computation for a degree l polynomial. Depending on the number of options for each of $j_1, j_2, \ldots, j_d$, the total cost of root computation is $(1 + l + l^2 + \ldots + l^{d-1})R_l$[10]. The standard algorithms for these are Berlekamp's algorithm [Ber70] with complexity $O(l^3 + l^2 \log l \log(p^2))$ operations in $\mathbb{F}_{p^2}$ or Cantor-Zassenhaus algorithm [CZ81] with complexity $O(l^3 \log(p^2))$ operations in $\mathbb{F}_{p^2}$. We can consider an optimized complexity of $O(l^3 + l^2 \log(p^2))$ in $\mathbb{F}_{p^2}$ [Sho09]. The deterministic variants are more costly.
Replacing operations in $\mathbb{F}_{p^2}$ with those in $\mathbb{F}_p$ and also considering the total number of roots we pointed out, the total complexity when the root finding method is used is $\frac{1}{l-1}(l^d - 1)O(l^3 + l^2 \log p)$. The dominating factor in this case is $O(l^{(d+1)} \log p)$ operations in $\mathbb{F}_p$.

5.5.2 Gröbner basis algorithms

In this method, we consider a system of equations which we solve by creating the Gröbner basis of related ideals and then obtaining the minimal polynomial.

5.5.3 Constructing the ideal satisfying system of equations

The system of equations 7 will generate an ideal I in the multivariate polynomial ring.

$$I = \langle \phi_l(j, j_1), \phi_l(j_1, j_2), \ldots, \phi_l(j_{d-1}, j_d) \rangle \subseteq \mathbb{F}_{p^2}[j_1, j_2, \ldots, j_d]$$

[10] We compute roots till the $(d-1)$-th level and obtain all supersingular ellipic curves at the d-th level.

The modular polynomial for a prime $l \neq p$, can have at most $(l+1)$ distinct roots. Due to this, the dimension of the vector space $\mathbb{F}_{p^2}[j_1, j_2, \ldots, j_d]/I$ is bounded above by $(l+1)^d$. By the finiteness theorem [Chapter 2, [CLO05]] and considering the fact that $j \in \mathbb{F}_{p^2}$, I is zero-dimensional. Consequently, there is a non-zero polynomial in $I \cap \mathbb{F}_{p^2}[j_k]$ for each $k = 1, 2, \ldots, d$.

5.5.3.1 The minimal polynomial for the last variable

Let us consider the *lex* ordering $j_d > j_{d-1} > \ldots > j_2 > j_1$ and a corresponding Gröbner basis. Let us recall that by Hilbert's basis theorem [Chapter 1, [CLO05]] and Buchberger's criteria [Chapter 1, [CLO05]], the existence of a Gröbner basis is valid. Again by the FGLM algorithm [Chapter 2, [CLO05]], we can construct the Gröbner basis for the particular monomial ordering, and then find the minimal polynomial of j_d with respect to the ideal I. Let m_d denote the minimal polynomial. We can then check whether the roots of this minimal polynomial is in $\mathbb{F}_p$ or S_1 to ensure that a mirror path between j and j^p exists.
Considering the number of variables d and the dimension of $\mathbb{F}_{p^2}[j_1, j_2, \ldots, j_d]/I$ over $\mathbb{F}_{p^2}$ being at most $(l+1)^d$, the complexity of using the FGLM algorithm is $O(d(l+1)^{3d})$ [Proposition 4.1, [FGLM93]]. Fortunately, we can take advantage of some generic relation and forego this cost.

5.5.3.2 Pre-computing the minimal polynomials

We notice that we can actually avoid computation of this minimal polynomial at each step, by using the generic relation [Remark 3.1,[TKF+20]] that exists between modular polynomials and corresponding minimal polynomials. The authors claimed to have verified it. We have also run codes for some test cases, where we found this to hold true in all the cases. Technically for $d \geq 3$:

$$m_d(j, x) = \phi_{l^{d-2}}(j, x)\phi_{l^d}(j, x)$$

We can pre-compute minimal polynomials m_d for some d from the knowledge of modular polynomials already pre-computed before. One possible source of such pre-computed modular polynomials is a database maintained by Sutherland. This cost is then a one-time cost. This will remove the cost of d at each step. Instead, we need to substitute the given j each time in $m_d(x, y)$, where $(m_d(x, y) = \phi_{l^{d-2}}(x, y)\phi_{l^d}(x, y)$ as modular polynomials are symmetric). The degree of this minimal polynomial in each variable is the sum of the degrees of the component modular polynomials in each variable which is $(l+1)l^{d-3}$ for $\phi_{l^{d-2}}(x, y)$ and $(l+1)l^{d-1}$ for $\phi_{l^d}(x, y)$. Clearly, this sum is $O(l^d)$. Using the same strategies as in Section 5.2.2, to evaluate modular polynomials of degree $l+1$, we can evaluate $m_d(x, y)$ at $x = j$ in $O(l^d)$ operations in $\mathbb{F}_p$.
This method is cost-wise much better than constructing the modular polynomial at each step which multiplies the complexity by $O(dl^{2d})$.

5.5.3.3 Testing for mirror paths

The next aim would be to test whether the minimal polynomials m_d have a root that gives rise to a mirror path between j and j^p.
The naive method over here is to construct a system of $(i-1)$ equations like 7, and then test whether j^p is the root of this system by evaluating the corresponding minimal polynomial of degree $O(l^{i-1})$ at

$x = j$ and $y = j^p$. The cost of this evaluation would be $O(l^{2(i-1)})$. Considering the fact that $i = 2d$ if it is even and $i = 2d + 1$ if it is odd, the total cost of this evaluation would be $O(l^{2i-2})$ which is $O(l^{4d-2})$ if i is even and $O(l^{4d})$ if i is odd.
The above cost is quite high for larger values of i. We thus, try to see if modifications could lead to better complexities. The situation of mirror paths indicates that intermediate roots are either in $\mathbb{F}_p$ for even i and over S_1 for odd i.

Roots in $\mathbb{F}_p$ We notice that when our aim is to know whether any root of the minimal polynomial is in $\mathbb{F}_p$ or not, we can formulate the easier problem of checking gcd by constructing polynomials g_1 and g_2 of equations 4 and 5. The analysis is same as in Section 5.3.1, when we replace f by $m_d(j, x)$ of degree $O(l^d)$.

The analysis is almost the same, except the fact that now the degree of $m_d(j, x)$ is $O(l^d)$. The corresponding polynomials g_1, g_2 are

$$g_1 = \frac{1}{2}[m_d(j,x)+\pi(m_d(j,x))] = Re(a_n)x^n+Re(a_{n-1})x^{n-1}+Re(a_{n-2})x^{n-2}+\ldots+Re(a_1)x^1+Re(a_0) \quad (8)$$

$$g_2 = \frac{-\beta}{2}[m_d(j,x)-\pi(m_d(j,x))] = Im(a_n)x^n+Im(a_{n-1})x^{n-1}+Im(a_{n-2})x^{n-2}+\ldots+Im(a_1)x^1+Im(a_0) \quad (9)$$

for $m_d(j, x) = \sum_{i=0}^{n} a_i x^i$. We can now obtain the polynomials g_1, g_2 over $\mathbb{F}_p$ as in equation 6, by using the relations between any element and it's conjugate and adding the coefficients of β^0, β^1. Algorithm 1 of [CRSCS22], will now take time $O(n^2)$ which in this case can be obtained as n is the degree of the modular polynomial $O(l^d)$ as in Section 5.5.3.2. Thus the cost of evaluating the gcd without inverse computation will be $O(l^{2d})$.
By Proposition 1 of [CRSCS22], if the degree of this gcd is 1, then a root over the subfield exists. We note that the range of l, d that we wish to consider is not too big, and so this is comparable to the analysis for the range of values of l considered in [CRSCS22], so that the probability of the degree of the gcd being 1 is not negligible. The complete cost of evaluating a pre-computed minimal polynomial and knowing whether there is a root over $\mathbb{F}_p$ or not, is the sum of $O(l^d \log p)$ [Section 5.5.3.2] and $O(l^{2d} \log p)$ which is $O(l^{2d} \log p)$. For $d \geq 2$, the above cost exceeds the cost to compute individual roots.

Roots in S_1 In this section, we present Algorithm 1 to test whether there are l^d isogenies resulting in a root in S_1. We assume that we are given a j-invariant j and our task is to know whether $m_d(j, x)$ has a root in S_1. Technically, we aim to see whether the polynomials $m_d(j, x)$ and $\phi_l(x, x^p)$ have common roots. We consider the resultant R of these polynomials with respect to the variable a after noting that each polynomial can be considered as a bivariate polynomial to remove p from the exponent.
The degrees of the polynomial M and Φ are $O(l^d)$ and $O(l)$ respectively. Thus the construction of the Sylvester matrix will take about $O(l^{d+1})$ operations and the determinant will require about $O(l^{3d} \log p)$ operations. The total cost, which will also include the cost of later steps of gcd and root computation thus exceeds the cost of individual root computation as in Section 5.5.1.

Algorithm 1: Testing for vertices in S_1.

Input: A prime l, a positive integer d and a supersingular j-invariant $j_{00} \in \mathbb{F}_{p^2}$.
Output: Whether there exists a vertex $j_d \in S_1$ such that j_{00} and j_d are l^d-isogeneous.

$M \leftarrow m_d(j_{00}, a + \beta b)$.//Bivariate polynomial in $\mathbb{F}_{p^2}[a, b]$.
$\Phi \leftarrow \phi_l(a + \beta b, a - \beta b)$.//Bivariate polynomial in $\mathbb{F}_p[a, b]$.
$R = Resultant(M, \Phi, a)$.//Resultant of above polynomials with respect to a.
if *R has a root b_0 in $\mathbb{F}_p$* **then**
    $g = GCD(M(a, b_0), \Phi(a, b_0))$.
    if *g has a root b_0 in $\mathbb{F}_p$* **then**
        Return *Yes*.
Return *False*.

5.6 Optimal method to detect mirror paths

Given the previous discussions, we observe that the cost of testing whether vertices are in $\mathbb{F}_p$ or S_1 is $O(l^2 \log p)$ (Section 5). To search whether the j-invariant at some level leads to a mirror path within a few steps, we have to detect whether there exists a small integer $i \geq 0$ such that j and j^p are l^i isogeneous. The cost for $i = 0$ and $i = 1$ is the cost of detecting vertices in $\mathbb{F}_p$ and S_1 respectively. This implies that the minimum cost is $O(l^2 \log p)$. Also for $i = 2, 3$ the value of $d(= \lfloor \frac{i}{2} \rfloor)$ is 1, so the cost in these cases is also $O(l^2 \log p)$ (Section 5.5).

Best value of i and corresponding cardinality The above analysis indicates that the best value of i to balance the cost of computing isogenies with the advantage of viewing a longer path is $i = 3$. Beyond $i = 3$ the cost incurred exceeds $O(l^2 \log p)$ and grows further with an increase in i. Keeping the notations of equation 7, the set of our interest is $S_0 \cup S_1 \cup \{j \in S_2 | j_d \in S_0\} \cup \{j \in S_3 | j_d \in S_1\} = S_0 \cup S_1 \cup \overline{S_2} \cup \overline{S_3}$..

The steps: Given a j-invariant we test first whether it belongs to $\mathbb{F}_p$. The cost of this is $O(1)$. If it is found, then the mirror path is already found. Else we now test whether it is in S_1 which will take time $O(l^2 \log p)$. On the occasion, it is not in S_1, we can test whether any neighbour of j (*i.e.*, root of $\phi_l(j, x)$) is in $\mathbb{F}_p$, without actually computing the roots of it by computing the gcd of two polynomials, as in Section 5.3. This will also require the same cost of $O(l^2 \log p)$. Beyond, this point root computation is essential, and the best cost can be approximated by $O(l^2 \log p)$, with the roots being computed with probabilistic algorithms (Section 5.5.1). This cost is asymptotically equal to the cost in [EHL+20], with the gain being practical. We are able to view a larger proportion of the graph at each step, which will provide better options to choose favourable routes to reach the frobenius conjugate earlier.

6 Choice of l

The number of supersingular elliptic curves in $\mathbb{F}_p$ or S_1 increases, in the graph $\chi_l(\overline{\mathbb{F}_p})$ on increasing l. By Lemma 6 of [CLG09] for mirror paths of length i, it is given by $l^{\frac{i}{2}} O(\sqrt{p})$. The number of isogeny cycles also increases with the increase in l [Theorem 7.1 of [ACL+24]]. This favour towards a higher value of l is also aggravated by the fact that using only $l = 2$ may result in some amount of

non-randomness, short cycles, or walks.
However, the cost of root finding also increases with the increment of l. This means that to choose l, suitable for our purpose, this balance so that the cost of increase in root finding is counter-acted by the expansion of suitable vertices to yield mirror paths is an important factor that we should keep in mind.

We performed experiments with various values of p and small values of l to find which l takes least time to find a mirror path. We also discuss how to calculate the cost and fix l suitably.

6.1 Experimental observations

We choose primes of size 35 to 45 bits and for each prime, we generated a list of supersingular j-invariants randomly. We aimed to perform a pseudo-random walk for each j-invariant until we found some isogeneous (isogeny being some power of l) j-invariant for each, which would lead to a mirror path. This ensured that the number of vertices we would walk through would be substantial enough for inference. Let j_0 be any j-invariant chosen from the list. Initially, we set $j = j_0$. At each step, we computed the root of the classical modular polynomial $\phi_l(j, x)$. If a root was found, in $\mathbb{F}_p$ or S_1 then the algorithm halted for this j_0. Else choosing j as any random root we again continued the process.
We performed the above experiment for different values of small primes $l(\leq 59)$. We noted the l for which time to find the mirror path was least, considering all the vertices. We call this l optimal, which balances the cost of obtaining roots with the increase in number of roots for that prime in the most efficient manner. The optimal l differed for values of p chosen. We have listed them in Table 3. These observations indicate that given p, we need to choose l after calculating the cost at each point. This is more important as our observations were for creating a small number of cycles, although we walked over a large number of j-invariants. The difference in time required by various values of l should increase when larger number of cycles have to be constructed in practical situations. This makes it necessary to choose l suitably.

Table 3: Table showing that for a given prime, there is a prime l, which finds a requisite vertex for mirror path in shorter time.

p	Best l
34359738337	7 and 3
274877906857	7
1099511627689	3
2199023255521	13
8796093022141	17
35184372088777	23

6.2 Cost to optimize to pre-compute l in preprocessing phase

We focus on the specific scenario on how to choose a small l more appropriate for the problem. Let us denote the cost per node by c_l. We can compute c_l for the first few small primes l. Considering the entire random walk, the total cost is obtained by multiplying the cost c_l with the number of steps. We can then fix l as that prime for which the total cost is minimum.
We keep in mind that each modular polynomial $\phi_l(x, y)$ is a degree $(l + 1)$ polynomial in each of x, y

and we can approximate one $\mathbb{F}_{p^2}$ multiplications by three multiplications in $\mathbb{F}_p$. Considering multiplications as the dominant cost, and our ultimate aim is to check the membership of a given j-invariant in $S_0 \cup S_1 \cup \overline{S_2} \cup \overline{S_3}$, we can compute the exact cost at any given j-invariant j as the sum total of the exact costs of evaluating $\phi_l(j, x)$ at $x = j^p$, the cost of finding whether $\phi_l(j, x)$ has a root in $\mathbb{F}_p$ and the cost of root computation of $\phi_l(j, x)$. The cost c_l is obtained by dividing the sum of these exact costs by the degree $(l + 1)$ of the modular polynomial. We can pre-compute this costs $c_{l_i}, 1 \leq i \leq t$, for some small primes $l_1 < l_2 < \ldots < l_t$.
The exact cardinality of the sets S_i is difficult to compute, given it is associated with the cardinality of class groups. Estimations will be possible in special cases, when the cardinality of the associated class group, or a lower bound is available. There are methods [GJ16, MS24, Gél18] present in literature to compute class groups.
We can estimate the length L of a random walk from Proposition 1, and calculate the total cost $T_l = c_l \times L$. Both c_l and L being functions of the isogeny degree l, we can minimize the function T_l with respect to l and choose $l = l_i$ when $T_{l_i} = \min(T_{l_1}, T_{l_2}, \ldots, T_{l_t})$. We choose l, keeping in mind the splitting of p in an associated number field, so that short cycles [Section 5.3.4 [CLG09]] are not present.

Our experiments conclude that it is essential to choose l optimally, when we are searching for mirror paths. The optimization is possible when we find the cost c_l suitably and balance it by minimizing the total cost. This will help in adding practical gain to the frobenius conjugate path-finding algorithm.

7 Future Work

Two important venues of future work arise as a result of the discussions in this paper.
We have proved that to find path between conjugates in the graph $\chi_l(\overline{\mathbb{F}_p})$ by seeing l^i isogenies the optimal value of i is $i = 3$ and the optimal l depends on the cardinality of the target set $S_0 \cup S_1 \cup \overline{S_2} \cup \overline{S_3}$. There exists precise estimations of bounds on S_1[Theorem 3.9 [EHL+20]] and S_0[Equation 1, [DG16]]. This implies it remains a later work to derive estimations, especially the lower bounds on the cardinality of $\overline{S_i}$ up to $i \leq 3$.
Another future scope of work is to test membership of elements in $\overline{S_3}$ without finding the roots exactly. If it leads to a speed-up, a similar approach for other $\overline{S_i}$ can also help in improving run-time by raising the value of i, which would aid in viewing a larger portion of the graph.

8 Conclusion

The problem of finding a path between supersingular elliptic curves which are frobenius conjugates of each other, is a specialised case of the general path-finding problem on which supersingular isogeny-based cryptography is based. Considering the graph of all supersingular elliptic curves for a given prime p, the important aspect to keep in mind to solve the path finding problem between conjugates is wisely choosing the values of l, i when we are overviewing the graph i steps at a time.
The issue here is that conjugate vertices are closer [ACNL+23] to each other than arbitrary vertices of the graph $\chi_l(\overline{\mathbb{F}_p})$ of all supersingular elliptic curves so it is just sufficient to search for special vertices which are either equal to their frobenius conjugates or have it as a neighbour. The search complexity of such special vertices is exponential in p and polynomial in l. As optimally choosing l in general path-finding problem reduces run-time [CRSCS22], we were interested to see the application in this scenario. Our ex-

periments reveal that for different primes p, the value of l that leads to the smallest time to find appropriate paths for our problem is variable [Table 3]. We have framed a procedure[Section 6.2] to pre-compute l. Connected to this, we experiment the nature of the path between frobenius conjugates when $l > 2, 3$, which partially answers the issue of exploring paths as was mentioned in Section 4.1 of [ACNL+23] between frobenius conjugates and arbitrary vertices for a larger set of primes. To experimentally learn about the structure of isogeny graphs when short cycles are present, we introduce sets $\overline{S_i} = \{j \in S_i (i > 1) | \ j_{\frac{i}{2}} \in S_0$ if i is even and $j_{\lfloor \frac{i}{2} \rfloor} \in S_1$ if i is odd, where j_k represents any descendant at the k-th step$\}$ where the set of all supersingular j-invariants in $\mathbb{F}_p$ is denoted by S_0 and
$S_i = \{j \in \mathbb{F}_{p^2} : j$ is supersingular and is connected to it's conjugate j^p by an isogeny of degree $l^i\}$ for $i \geq 1$. For various primes l, p, we have recorded[Table 1 and Table 2] the frequency distribution of elements in $\overline{S_i}$ for small i.
We recall the question of choosing l and i suitably, so that instead of seeing just the immediate neighbour arising from 2−isogenies, we can expand our point of view and choose the path that accelerates the process of finding a mirror path. The connectedness of $\chi_l(\overline{\mathbb{F}_p})$ means for large i close to $O(\log p)$, the existence of isogenies is always guaranteed. We can broadly observe the graph for large i, but that increases the cost of computing isogenies. We proved[Section 5.6] $i = 3$ is optimal, which means the corresponding set is $S_0 \cup S_1 \cup \overline{S_2} \cup \overline{S_3}$. Also, $i = 3$ implies that the issue of bounding "by some fixed integer B"[Remark 3.5, [EHL+20]] is possible by taking $B = 3$. The literature contains estimations of sizes of S_0, S_1 and different modes of testing membership. This implies a future work is to compute cardinalities of $\overline{S_i}$ for i up to 3, and test membership in $\overline{S_3}$ without computing the roots.
The considerations on l and i will accelerate the process of pathfinding among frobenius conjugate, which will result in a speed-up of cycle finding. Concrete application scenarios like general endomorphism ring computation [EHL+20], SQIsign [DFKL+20] or construction of collision resistant hash functions [CLG09] which depends on the hardness of cycle finding will benefit from this practical improvement.

References

[ABC+25] Sarah Arpin, Ross Bowden, James Clements, Wissam Ghantous, Jason T LeGrow, and Krystal Maughan, *Cycles and cuts in supersingular l-isogeny graphs*, Cryptology ePrint Archive (2025).

[ACL+24] Sarah Arpin, Mingjie Chen, Kristin E Lauter, Renate Scheidler, Katherine E Stange, and Ha TN Tran, *Orientations and cycles in supersingular isogeny graphs*, Research Directions in Number Theory: Women in Numbers V, Springer, 2024, pp. 25–86.

[ACNL+23] Sarah Arpin, Catalina Camacho-Navarro, Kristin Lauter, Joelle Lim, Kristina Nelson, Travis Scholl, and Jana Sotáková, *Adventures in supersingularland*, Experimental Mathematics **32** (2023), no. 2, 241–268.

[BBC+21] Gustavo Banegas, Daniel J Bernstein, Fabio Campos, Tung Chou, Tanja Lange, Michael Meyer, Benjamin Smith, and Jana Sotáková, *Ctidh: faster constant-time csidh*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2021** (2021), no. 4, 351–387.

[BDF+24] Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski, *Sqisign2d–west: The fast, the small, and the safer*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2024, pp. 339–370.

[Ber70] Elwyn R Berlekamp, *Factoring polynomials over large finite fields*, Mathematics of computation **24** (1970), no. 111, 713–735.

[Ber20] D Bernstein, *Faster computation of isogenies of large prime degree*, Tech. report, IACR Cryptology ePrint Archive, 2020: 341, 2020.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren, *Csi-fish: efficient isogeny based signatures through class group computations*, International conference on the theory and application of cryptology and information security, Springer, 2019, pp. 227–247.

[CLG09] Denis X Charles, Kristin E Lauter, and Eyal Z Goren, *Cryptographic hash functions from expander graphs*, Journal of CRYPTOLOGY **22** (2009), no. 1, 93–113.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes, *Csidh: an efficient post-quantum commutative group action*, Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24, Springer, 2018, pp. 395–427.

[CLO05] David A Cox, John Little, and Donal O'shea, *Using algebraic geometry*, vol. 185, Springer Science & Business Media, 2005.

[CRSCS22] Maria Corte-Real Santos, Craig Costello, and Jia Shi, *Accelerating the delfs–galbraith algorithm with fast subfield root detection*, Annual International Cryptology Conference, Springer, 2022, pp. 285–314.

[CZ81] David G Cantor and Hans Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Mathematics of Computation (1981), 587–592.

[DFG19] Luca De Feo and Steven D Galbraith, *Seasign: compact isogeny signatures from class group actions*, Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38, Springer, 2019, pp. 759–789.

[DFKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *Sqisign: compact post-quantum signatures from quaternions and isogenies*, Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26, Springer, 2020, pp. 64–93.

[DG16] Christina Delfs and Steven D Galbraith, *Computing isogenies between supersingular elliptic curves over f _p f p*, Designs, Codes and Cryptography **78** (2016), 425–440.

[E+98] Noam D Elkies et al., *Elliptic and modular curves over finite fields and related computational issues*, AMS IP Studies in Advanced Mathematics **7** (1998), 21–76.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit, *Supersingular isogeny graphs and endomorphism rings: reductions and solutions*, Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part III 37, Springer, 2018, pp. 329–368.

[EHL+20] Kirsten Eisenträger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park, *Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs*, Open Book Series **4** (2020), no. 1, 215–232.

[ES24] Kirsten Eisentraeger and Gabrielle Scullard, *Connecting kani's lemma and path-finding in the bruhat-tits tree to compute supersingular endomorphism rings*, arXiv preprint arXiv:2402.05059 (2024).

[FGLM93] Jean-Charles Faugere, Patrizia Gianni, Daniel Lazard, and Teo Mora, *Efficient computation of zero-dimensional gröbner bases by change of ordering*, Journal of Symbolic Computation **16** (1993), no. 4, 329–344.

[FIK+23] Jenny Fuselier, Annamaria Iezzi, Mark Kozek, Travis Morrison, and Changningphaabi Namoijam, *Computing supersingular endomorphism rings using inseparable endomorphisms*, arXiv preprint arXiv:2306.03051 (2023).

[Gél18] Alexandre Gélin, *Reducing the complexity for class group computations using small defining polynomials*, arXiv preprint arXiv:1810.12010 (2018).

[GJ16] Alexandre Gélin and Antoine Joux, *Reducing number field defining polynomials: an application to class group computations*, LMS Journal of Computation and Mathematics **19** (2016), no. A, 315–331.

[GPS20] Steven D Galbraith, Christophe Petit, and Javier Silva, *Identification protocols and signature schemes based on supersingular isogeny problems*, Journal of Cryptology **33** (2020), no. 1, 130–175.

[KKA+24] Yuta Kambe, Akira Katayama, Yusuke Aikawa, Yuki Ishihara, Masaya Yasuda, and Kazuhiro Yokoyama, *Computing endomorphism rings of supersingular elliptic curves by finding cycles in concatenated supersingular isogeny graphs*, COMMENTARII MATHEMATICI UNIVERSITATIS SANCTI PAULI **72** (2024), no. 1, 19–42.

[Koh96] David Russell Kohel, *Endomorphism rings of elliptic curves over finite fields*, University of California, Berkeley, 1996.

[MMP24] Marzio Mula, Nadir Murru, and Federico Pintore, *On random sampling of supersingular elliptic curves*, Annali di Matematica Pura ed Applicata (1923-) (2024), 1–43.

[MS24] Madhurima Mukhopadhyay and Palash Sarkar, *Pseudo-random walk on ideals: practical speed-up in relation collection for class group computation*, Cryptography and Communications (2024), 1–21.

[MW23] Arthur Herlédan Le Merdy and Benjamin Wesolowski, *The supersingular endomorphism ring problem given one endomorphism*, arXiv preprint arXiv:2309.11912 (2023).

[Piz90] Arnold K Pizer, *Ramanujan graphs and hecke operators*, Bulletin of the American Mathematical Society **23** (1990), no. 1, 127–137.

[PW24] Aurel Page and Benjamin Wesolowski, *The supersingular endomorphism ring and one endomorphism problems are equivalent*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2024, pp. 388–417.

[RNSL17] Martin Roetteler, Michael Naehrig, Krysta M Svore, and Kristin Lauter, *Quantum resource estimates for computing elliptic curve discrete logarithms*, Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23, Springer, 2017, pp. 241–270.

[Sho99] Peter W Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM review **41** (1999), no. 2, 303–332.

[Sho09] Victor Shoup, *A computational introduction to number theory and algebra*, Cambridge university press, 2009.

[Sil86] Joseph H Silverman, *Heights and elliptic curves*, Arithmetic geometry, Springer, 1986, pp. 253–265.

[Sil94] ———, *Advanced topics in the arithmetic of elliptic curves*, vol. 151, Springer Science & Business Media, 1994.

[Sil09] ———, *The arithmetic of elliptic curves*, vol. 106, Springer, 2009.

[TKF+20] Yasushi Takahashi, Momonari Kudo, Ryoya Fukasaku, Yasuhiko Ikematsu, Masaya Yasuda, and Kazuhiro Yokoyama, *Algebraic approaches for solving isogeny problems of prime power degrees*, Journal of Mathematical Cryptology **15** (2020), no. 1, 31–44.

[Vél71] Jacques Vélu, *Isogénies entre courbes elliptiques*, Comptes-Rendus de l'Académie des Sciences **273** (1971), 238–241.

[Was08] Lawrence C Washington, *Elliptic curves: number theory and cryptography*, Chapman and Hall/CRC, 2008.

[XZQ24] Guanju Xiao, Zijian Zhou, and Longjiang Qu, *Endomorphism rings of supersingular elliptic curves and quadratic forms*, arXiv preprint arXiv:2409.11025 (2024).