CORAL
Faster Isogeny Group Action for Post-Quantum NIKE

Andrea Basso[1], Giacomo Borin[1,2], Ryan Rueger[1,3], and Sina Schaeffler[1,4]

[1] IBM Research Europe, Switzerland
andrea.basso@ibm.com
[2] University of Zurich, Switzerland
coral@gbor.in
[3] Technische Universität München, Germany
ryan@rueg.re
[4] ETH Zurich, Switzerland
sschaeffle@ethz.ch

Abstract. There are two kinds of cryptographic group actions: restricted and unrestricted. While unrestricted actions like (qt-)PEGASIS are needed for more advanced constructions, restricted ones like dCTIDH are sufficient for instantiating a NIKE and usually much more efficient.
In this work, we propose CORAL, a significantly faster algorithm to evaluate the same action as (qt-)PEGASIS, but in a restricted fashion; CORAL only computes two-dimensional 2-isogenies to evaluate the action and outperforms both recent unrestricted (KLaPoTi, (qt-)PEGASIS) and (restricted) CSIDH-based approaches (SQALE, dCTIDH). In essence, CORAL trades off unrestrictedness for efficiency.
Our unoptimised C implementation evaluates a group-action in 240 ms with a 2032-bit prime. When used to construct a non-interactive key exchange, CORAL yields an actively secure post-quantum NIKE with compact public keys (e.g. 256 bytes for 2032-bit primes).

Keywords: Group Actions · Isogenies · NIKE

1 Introduction

Cryptographic group actions are a powerful framework for constructing public-key primitives in the post-quantum setting. Recall that group actions $* : G \times X \to X$ are compatible with the group law, i.e. satisfy $g * (g' * x) = (gg') * x$ for all g, g', x, and are said to be *cryptographic* if it is hard to recover g from the pair $g * x, x$.

This framework, first proposed by Brassard and Yung [BY91] and later developed by Alamati, De Feo, Montgomery, and Patranabis [ADMP20], naturally

May 2026
Authors listed alphabetically ams.org/profession/leaders/CultureStatement04.pdf.

generalises group exponentiation and so gives rise to many cryptographic primitives known from discrete-log assumptions. Indeed, in direct analogy to their discrete-log counterparts, one can construct non-interactive key exchange [Cou06, RS06,CLM+18], signatures [DG19,BKV19], blind signatures [KLLQ23,HLM+25], threshold protocols [DM20], oblivious pseudorandom functions [BKW20,HHM+24], verifiable random functions [Lai24], and more [ADMP20] using cryptographic group actions. Importantly, these primitives are secure in a post-quantum setting when instantiated from the *class-group action* on elliptic curves, the only known post-quantum (commutative) group action. A notable absence from this list of constructions are Schnorr-type signatures [Sch90], precisely because there is no (efficiently computable) group structure on public keys.

In 1996, Couveignes [Cou06] and later, independently, in 2006 Rostovtsev and Stolbunov [RS06] suggested the class-group action on *ordinary* elliptic curves for cryptographic applications; and in 2018, by translating these ideas to *supersingular* curves, Castryck, Lange, Martindale, Panny, and Renes [CLM+18] were able to construct the first *practical* post-quantum cryptographic group action CSIDH with runtimes measured in milliseconds instead of minutes. This sparked a long line of research into high-performance, constant-time variations of CSIDH, with parameters tuned for post-quantum key-exchange (e.g. [BBC+21,CSCJR22, CHMR25]).

The parameters underlying CSIDH inherently define a polynomially-sized set of class-group elements $g_1, \ldots, g_n$ by which the action can be efficiently evaluated in practice. By forming small products $g_e = g_1^{e_1} \cdots g_n^{e_n}$ and acting iteratively, one obtains a practically efficient evaluation algorithm for an exponentially-sized subset of the class-group. The runtime of this algorithm is linear in the ℓ_1-norm $e_1 + \cdots + e_n$ of the *exponent vector* $e = (e_1, \ldots, e_n)$: there is no notion of square-and-multiply for group actions.

Only acting with group elements in this small-product form is sufficient for non-interactive key-exchange (and more, e.g. [ADMP20]), but becomes a limitation for signatures (and more, e.g. [DM20]). Indeed, CSIDH cannot evaluate the action of an element g if it is not written as the product of the g_i; and even when it is $g = g_e$, the $\Theta(\ell_1(e))$ runtime prohibits large e_i. This makes CSIDH a *restricted* cryptographic action [ADMP20].

As already noted by CSIDH, it is possible to reduce arbitrary group elements g_e (i.e. with potentially large exponent-vector entries) into small products at runtime if the *structure* of the class-group is known. Although (pre-)computing the structure generally has subexponential complexity [HM89], Beullens, Kleinjung, and Vercauteren [BKV19] demonstrated that it is feasible for the CSIDH-512 parameters through a 170 GHz-core-year (pre-)computation. Using the class-group structure, they could "unrestrict" CSIDH-512 with minimal overhead to deliver the first *unrestricted* cryptographic group action, and construct from this the signature CSI-FiSH.

Since the class-group is commutative, its action is susceptible to subexponential quantum attacks based on Kuperberg's algorithm [Kup05, Kup13]; careful analysis of which has led to the concrete quantum security of CSIDH-512 be-

ing called into question [BS20, Pei20]. Since class-group-structure computations for larger CSIDH parameters are out of reach, there was interest for alternative *scalable* constructions.

The SCALLOP family [DFK+23, CLP24, ABE+25] realises this goal, albeit with significantly degraded concrete performance, by considering (class-groups of) imaginary quadratic orders of carefully chosen *large conductor*. The structure of these class-groups could be computed with modest resources in practice, despite the (pre-)computation still being asymptotically sub-exponential. While there are possible time-complexity tradeoffs between the class-group structure pre-computation and the runtime decomposition into small products, no configuration allows for a fully polynomial-time evaluation algorithm [Pan23].

In 2023, Page and Robert finally developed a polynomial-time *unrestricted* algorithm Clapoti to evaluate the class-group action [PR23], using the higher-dimensional techniques developed in the SIDH attacks [CD23, MMP+23, Rob23] central to which was *Kani's lemma* [Kan97, Th. 2.3; Rob23, Lem. 6]. Notably, this algorithm does not require decomposing class-group elements into small products, and in particular does not need to know the class-group structure. From this polynomial-time algorithm sprung concretely efficient implementations, KLaPoTi [PPS25] and PEGASIS [DEF+25]; and later the improvements qt-PEGASIS [DEIV25, DD26], which vastly outperform all SCALLOP-like constructions, but are still significantly slower than CSIDH.

While these works unlock the full range of constructions from (unrestricted) group actions, they each have their practical bottlenecks which are especially expensive at the parameters considered necessary for quantum security (2048 and 4096 bits [Pei20, BLMP19]): (qt-)PEGASIS uses four-dimensional isogenies, while KLaPoTi works in dimension two, but requires very large base-field sizes with respect to the class-group size.[5]

Despite the large body of work on unrestricted actions, we point out that the fastest implementations of restricted actions [WLLZ26, ZLL+26] still outperform their fastest unrestricted counterparts [DEIV25, DD26] by a factor of around 32 (at 4096 bit discriminants, see Table 4). While much faster, these restricted actions are fundamentally incompatible with the unrestricted ones, e.g. if one uses a (faster) restricted action for NIKE, one would still need to use a (different, slower) unrestricted action to do signatures with.

Our contributions. We present CORAL,[6] one of the fastest restricted class-group evaluation algorithms to date, that is still compatible with the unrestricted (qt-)PEGASIS actions. More precisely

– By picking the same field characteristic p and same action of $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$ as (qt-)PEGASIS, CORAL may be viewed as a lower-dimensional (and thus more efficient) restricted version of (qt-)PEGASIS. As such, CORAL can be combined with (qt-)PEGASIS to accelerate more advanced primitives constructed from group-actions.

[5] The cost of isogeny computations in dimension g is exponential in g.

[6] COmmutative Restricted Action from Lower-dimensional isogenies.

- Our unoptimised C implementation performs one group-action evaluation in 240 ms for a 2032-bit discriminant and 1 780 ms for a 4084-bit discriminant.
- The key ingredient of our construction is an algorithm to sample elements of the class-group whose action can be computed concretely efficiently using a single chain of two-dimensional 2-isogenies, inspired by techniques of [NO24].
- The distribution of the sampling algorithm appears not to admit a meet-in-the-middle attack, and so we are able to sample from a set of size 2^λ inside a significantly larger class-group, similar in flavour to [CSCJR22, CHMR25, Hou25]. This provides classical security against brute-force search, while the large class-group resists quantum attacks of subexponential-complexity based on Kuperberg's algorithm.

Technical overview. Similarly to CSIDH, CORAL is a restricted algorithm to compute the action induced by the Frobenius orientation, but does so for the exact same parameters as (qt-)PEGASIS.

The Orientation. Like (qt-)PEGASIS, CORAL evaluates the action of $\mathrm{Cl}(\mathbb{Z}[(\pi+1)/2])$ on $\mathbb{Z}[(1+\pi)/2]$-oriented curves with $p = c2^f - 1$, where c is a small odd cofactor. Recall that $\mathbb{Z}[(1+\pi)/2]$-oriented curves are defined over $\mathbb{F}_p$ with $\mathbb{F}_p$-rational 2-torsion and are on the *surface* of the 2-isogeny volcano [CD20, FM02, Sut13, DG16]; moreover, $\mathrm{Cl}(\mathbb{Z}[(\pi+1)/2]) \cong \mathrm{Cl}(\mathbb{Z}[\pi])$ when $p \equiv 7 \pmod 8$ [FHL+26, Lem. 4.3], allowing for a slightly simpler representation of class-group elements (i.e. requiring only integer coefficients). Similarly to (qt-)PEGASIS, computing the action on the surface allows us to work over $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$ which significantly improves performance.

Two-dimensional evaluation. As shown in Clapoti [PR23, Prop. 2.1], given two equivalent ideals $\mathfrak{a}, \mathfrak{b}$ of coprime norms inside an imaginary quadratic order $\mathcal{O}$, one can apply *Kani's lemma* to embed the one-dimensional isogenies $\varphi_\mathfrak{a}, \varphi_\mathfrak{b}$ into a two-dimensional N-isogeny $\Phi\colon E^2 \to E_\mathfrak{a} \times E_{\overline{\mathfrak{b}}}$, where $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$ (See Section 3.2). When N is smooth and the N-torsion defined over a small extension (i.e. *accessible*), the codomain of Φ can be computed in time polynomial in $\log(p), \log(\mathrm{disc}(\mathcal{O}))$ to obtain $E_\mathfrak{a}$, i.e. the action of $[\mathfrak{a}] = [\mathfrak{b}]$ on E.

Prior unrestricted algorithms. Finding (equivalent) representatives $\mathfrak{a}, \mathfrak{b}$ of a *given* class $[\mathfrak{c}]$, with suitable $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$ (i.e. smooth and $E[N]$ accessible) appears to be difficult. Clapoti unconditionally circumvented this issue by employing eight-dimensional isogenies to obtain a polynomial-time algorithm; and suggested a four-dimensional heuristic version which was later refined in (qt-)PEGASIS to obtain an efficient algorithm using (four-dimensional) 2-isogenies. KLaPoTi [PPS25] took an alternative approach: by translating the problem of finding two equivalent ideals with suitable N into a problem of finding a quaternion of a smooth norm, they were able to apply the KLTP algorithm [KLPT14] to find $\mathfrak{a}, \mathfrak{b}$. However, in order to obtain a suitable $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$ from KLPT, p must be of size $\log(p) \approx |\mathrm{disc}(\mathcal{O})|^3$ (as opposed to $\approx |\mathrm{disc}(\mathcal{O})|$ in (qt-)PEGASIS), increasing

the cost of field arithmetic, and preventing the use of the Frobenius orientation, which is more efficient (in both time and bandwidth).

Our restricted algorithm. Despite suitable $\mathfrak{a}, \mathfrak{b}$ being difficult to find for a *predetermined* class $[\mathfrak{c}]$, it turns out that the set $\mathcal{C}_e$ of all classes represented by pairs $\mathfrak{a}, \mathfrak{b}$ satisfying $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$ is heuristically of size $\approx 2^e p^{-1/2}$ (See Section 4.1). By solving a relatively straightforward norm-equation, we are able to directly sample equivalent ideals $\mathfrak{a}', \mathfrak{b}'$ with $\mathrm{N}(\mathfrak{a}') + \mathrm{N}(\mathfrak{b}') = 2^e$, representing some *a priori* unknown class $[\mathfrak{c}']$ in $\mathcal{C}_e$. Precisely because we cannot arbitrarily choose the class $[\mathfrak{c}']$ in $\mathcal{C}_e$, we obtain a *restricted* group action.

Our ideal-sampling algorithm. Letting $\mathfrak{a}, \mathfrak{b} \subseteq \mathbb{Z}[\pi]$ be equivalent ideals with $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$, we observe that $\mathfrak{a}\bar{\mathfrak{b}}$ is generated by an element $\theta = x + y\pi$ of norm

$$x^2 + y^2 p = \mathrm{N}(\theta) = \mathrm{N}(\mathfrak{a}\bar{\mathfrak{b}}) = \mathrm{N}(\mathfrak{a})\mathrm{N}(\mathfrak{b}) = 2^{2(e-1)} - a^2, \tag{1}$$

where $a = 2^{e-1} - \mathrm{N}(\mathfrak{a}) = \mathrm{N}(\mathfrak{b}) - 2^{e-1}$. Importantly, the converse also holds: any θ in $\mathbb{Z}[\pi]$ of norm $\mathrm{N}(\theta) = 2^{2(e-1)} - a^2$ yields equivalent ideals

$$\mathfrak{a} = (2^{e-1} - a, \theta)\mathbb{Z}[\pi], \quad \mathfrak{b} = (2^{e-1} + a, \bar{\theta})\mathbb{Z}[\pi]$$

of norms $\mathrm{N}(\mathfrak{a}) = 2^{e-1} - a, \mathrm{N}(\mathfrak{b}) = 2^{e-1} + a$ so that $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$ (see Lemma 3.1). Moreover, if $\mathrm{N}(\theta)$ is odd (i.e. a is odd), the norms $\mathrm{N}(\mathfrak{a}), \mathrm{N}(\mathfrak{b})$ are coprime.

As such, it suffices to find integer solutions (x, y, a) with y, a odd to Eq. (1) to find θ and consequently write down coprime-normed equivalent ideals $\mathfrak{a}, \mathfrak{b}$ representing a class in $\mathcal{C}_e$. We can do this efficiently by sampling random (odd) y values until it is possible to decompose $2^{2(e-1)} - y^2 p$ into the sum of two squares using Cornacchia's algorithm [Cor08, MN90]. Indeed, rearranging Eq. (1), we obtain the central norm-equation to be solved

$$2^{2(e-1)} - y^2 p = x^2 + a^2.$$

Smaller key space. Recalling that $\mathcal{C}_e$ is of size $\approx 2^e p^{-1/2}$, it suffices to randomly sample from $\mathcal{C}_e$ with $e \approx \log(p)/2 + \lambda$ to obtain a key-space of size 2^λ. Notably, it is not necessary to iteratively act with an exponent-vector of elements to obtain a large key space, as the CSIDH family does. In fact, unlike the CSIDH-family, a key-space of size 2^λ suffices because the $\mathcal{C}_e$ sets apparently do not decompose into non-trivial products of sets $\mathcal{C}_e = \mathcal{L}_e\mathcal{R}_e$, and so meet-in-the-middle attacks do not apply. As argued by SQALE [CSCJR22, Sec. 4] the complexity of Kuperberg's algorithm [Kup05, Kup13] for solving the group action inverse problem is $\exp(O(|\mathrm{Cl}(\mathbb{Z}[\pi])|^{1/2}))$, i.e. sub-exponential in the size of the ambient class-group and not the key-space. As such, we may choose $e \approx \log(p)/2 + \lambda$ and $\log(p)$ in $\Theta(\lambda^2)$ to obtain both classical and quantum security (precise bit-complexities of Kuperberg's algorithm are difficult to evaluate [BLMP19, BS20, Pei20]).

A small and efficient post-quantum NIKE. As with CSIDH, CORAL can be used to construct an actively-secure NIKE directly from the axioms of a (restricted) commutative group action: key generation consists of sampling an ideal class $[\mathfrak{s}_\mathsf{A}]$ and evaluating it on a fixed public curve E_0 to obtain a public key $E_\mathsf{A} = [\mathfrak{s}_\mathsf{A}]E_0$; the shared secret derivation is done the same way, but starting from another party's public key i.e. the shared secret with Bob is given by $\mathsf{ss}_\mathsf{AB} = [\mathfrak{s}_\mathsf{A}]E_\mathsf{B} = [\mathfrak{s}_\mathsf{B}]E_\mathsf{A}$. Both parties obtain the same shared secret precisely because the class-group is commutative.

Organization. We first present the CORAL group action in Section 3, describing the ideal sampling algorithm and the restricted action. Section 4 provides a thorough security analysis, including a theoretical and experimental analysis of the distribution of sampled ideals in the class group. Finally, we discuss our implementation results in Section 5.

2 Preliminaries

Notation. We use standard notation from algebraic number theory and isogeny-based cryptography. In addition, Pr is the probability function. The logarithm log is in base 2 and the natural logarithm is denoted by ln. Random sampling is denoted by $\xleftarrow{\$}$. Ordered pairs are denoted by parentheses.

2.1 Elliptic Curves and Isogenies

We refer the reader to [Sil09] for a thorough treatment of elliptic curves and isogenies, and [De 17] for a concise introduction with a view towards cryptography. Moreover, we refer to [CLM+18] for an introduction to the isogeny class-group action on curves defined over $\mathbb{F}_p$, and to [CK20, Onu20] for arbitrary orientations. Finally, we refer to to [ADMP20] for cryptographic applications of group actions.

Elliptic Curves. We recall that elliptic curves are abelian groups described by polynomial equations, and are exactly the set of one-dimensional principally polarised Abelian varieties. They have both arithmetic structure (they are groups) and geometric structure (they are projective varieties). An elliptic curve defined over a field with positive characteristic p is *supersingular* if the p-torsion is trivial [Sil09, V.3].

Isogenies. An *isogeny* between elliptic curves (or more generally between Abelian varieties of equal dimension) is a morphism of both groups and varieties with finite kernel. In particular, this means every isogeny can be locally described as a tuple of quotients of polynomials [Sil09, I.3]. More specifically, isogenies between elliptic curves defined over a field k of characteristic $\neq 2, 3$ can be put into a standard form $(f_1/f_2 : yf_3/f_4 : 1)$ whereby f_i are polynomials in $\overline{k}[x]$ and the pairs f_1, f_2 and f_3, f_4 each coprime.

For example, over $\mathbb{F}_7$

$$\begin{aligned} \{zy^2 = x^3 + xz^2\} &\to \{zy^2 = x^3 + 3xz^3\} \\ (x, y, 1) &\mapsto \big((x^2+1)/x : y(x^2-1)/x : 1\big) \end{aligned} \tag{2}$$

is an isogeny on the affine patch $\{z = 1\}$ between (supersingular) elliptic curves. Its kernel is $\{(0:1:0), (0:0:1)\}$, where $(0:1:0)$ is the neutral group element.

The *degree* of an isogeny is its degree as a morphism of varieties and can be read off its standard form as $\max(\deg(f_1), \deg(f_2))$. So the degree of our example isogeny (Equation (2)) is 2. An isogeny is *separable* if its degree is equal to the cardinality of its kernel. Separable isogenies are defined uniquely by their kernel up to post-composition with isomorphisms. We say an isogeny of Abelian varieties $A \to B$ is d-dimensional if A (and therefore B) has dimension d.

An *endomorphism* is an isogeny from an elliptic curve to itself, or the zero map. The prototypical examples are *scalar multiplication* (i.e. the maps sending $P \to nP$, for a non-zero integer n). We denote by $\mathrm{End}(E)$ the ring of endomorphisms of an elliptic curve E, where the ring structure is given by point-wise addition and composition.

Higher-dimensional embedding. We recall a consequence of the *Reducibility Theorem* [Kan97] known as *Kani's Lemma*, an important tool in isogeny-based cryptography to embed isogenies between elliptic curves into isogenies between abelian surfaces. Here we present a modern interpretation of this result due to Robert [Rob23, Lem. 6].

Lemma 2.1 (Kani). *Commuting diagrams*

$$\begin{array}{ccc} E & \xrightarrow{\varphi} & E' \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi'} \\ F & \xrightarrow[\varphi']{} & F' \end{array} \tag{3}$$

of isogenies between elliptic curves, satisfying $\deg(\varphi) = \deg(\varphi') = a$ *and* $\deg(\psi) = \deg(\psi') = b$*, induce* $(a+b)$*-isogenies of (principally polarized) abelian surfaces*

$$\Phi = \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix} : E \times F' \to E' \times F.$$

Moreover, when a, b *are coprime, the kernel of* Φ *is given by*

$$\ker(\Phi) = \{(aP, \psi'\varphi(P)) \mid P \in E[a+b]\}.$$

2.2 Restricted effective group actions

In [ADMP20], the authors introduced the framework of (restricted) effective group actions ((R)EGA) to formalize the intuitions behind CSIDH-like schemes.

The action of a group G on a set X is said to be *effective* if there are efficient algorithms for representing elements in X, computing the group operation in G, and computing the group action, which means computing $g \star x$ given $g \in G$ and $x \in X$. An action is said to be *restricted* if the group action can only be computed for a subset of G, which we can efficiently sample from with a distribution $\mathcal{D}$. To stress the difference between the two, we refer to an effective group action as *unrestricted*. We refer to [OZ24, ADMP20] for more details on the definitions and properties of (R)EGAs, for now we only state the main security assumption associated to these primitives.

Problem 2.2 ((Restricted)-Group Action Inversion Problem ((R)GAIP)). Given a group action $(G, X, \star)$ and a distribution $\mathcal{D}$ over G, the (R)GAIP is the following problem: Given $x_0 \in X$ and $x = g \star x_0$ for a random $g \leftarrow \mathcal{D}$, compute any $g' \in G$ such that $x = g' \star x_0$.

If $(G, X, \star)$ is an EGA, we refer to this problem just as the Group Action Inversion Problem (GAIP). In this case, the distribution $\mathcal{D}$ is uniform over G.

2.3 Class-group actions

We immediately specialise to our case of interest, namely $p \equiv 7 \pmod 8$ and the action of $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$ acting on $\mathbb{F}_p$-rational supersingular (primitively) $\mathbb{Z}[(\pi + 1)/2]$-oriented elliptic curves (i.e. those with $\mathbb{F}_p$-endomorphism ring $\mathbb{Z}[(\pi+1)/2]$, or equivalently with $\mathbb{F}_p$-rational 2-torsion). This matches precisely the setting of (qt-)PEGASIS [DEF+25, DEIV25], and is similar to [CLM+18]. For a more general treatment of *orientations* we refer to [CK20, Onu20]. In a wider context, the relationship between the action of class groups of imaginary quadratic orders and elliptic curves is known as *complex multiplication* [Cox13, Sil94, Lan87].

Imaginary quadratic orders. Let $p \equiv 3 \pmod 4$ be a prime and let $K = \mathbb{Q}(\sqrt{-p})$ be the imaginary quadratic number field of discriminant $\Delta_K = -p$, whose non-trivial field isomorphism we denote by $a \mapsto \overline{a}$. The element $\omega = (1 + \sqrt{-p})/2$ generates the *ring of integers* $\mathcal{O}_K = \mathbb{Z}[\omega] = \mathbb{Z} + \omega\mathbb{Z}$ of K. Every *order* of $\mathcal{O}_K$ is given by $\mathcal{O}_f = \mathbb{Z} + f\omega\mathbb{Z}$ for an integer f called the *conductor*, and has discriminant $\Delta_f = f^2\Delta_K$. As such, the orders of a number field are all contained inside the $\mathcal{O}_K$, the *maximal order*. We will be interested in the order $\mathcal{O}_2 = \mathbb{Z}[\sqrt{-p}]$ of conductor 2. Every ideal $\mathfrak{a}$ of $\mathcal{O}_f$ may be written as $\mathfrak{a} = ma\mathbb{Z} + m(b + f\omega)\mathbb{Z}$ for some integers a, b, m and has finite index m^2a inside $\mathcal{O}_f$ which we call its *norm* $\mathrm{N}(\mathfrak{a})$. We refer to [Coh93, Ch. 5] for further details.

Class-groups. We say an ideal $\mathfrak{a}$ of an order $\mathcal{O}_f$ is *invertible* if there exists another ideal $\mathfrak{b}$ such that $\mathfrak{ab}$ is principal and denote the set of invertible ideals by $\mathcal{I}(\mathcal{O}_f)$; moreover we say that two invertible ideals $\mathfrak{c}, \mathfrak{d}$ are *equivalent* $\mathfrak{c} \sim \mathfrak{d}$ if $\mathfrak{c}\overline{\mathfrak{d}}$ is principal. Clearly principal ideals are invertible. Moreover, all non-zero ideals of the maximal order $\mathcal{O}_K$ are invertible. Ideal equivalence is stable under multiplication ($\mathfrak{c} \sim \mathfrak{d}, \mathfrak{c}' \sim \mathfrak{d}'$ implies $\mathfrak{cc}' \sim \mathfrak{dd}'$) and so the quotient $\mathcal{I}(\mathcal{O}_f)/\sim$

is a well-defined multiplicative monoid whose neutral element is represented by principal ideals. This is a finite group, called the *class-group* $\mathrm{Cl}(\mathcal{O}_f)$ of $\mathcal{O}_f$. We refer to [Cox13, Ch. 7] for further details.

Isogenies from ideals. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathbb{F}_p$ and recall that in this case the Frobenius isogeny $\pi\colon E \to E^{(p)} : y^2 = x^3 + a^p x + b^p; (x, y) \mapsto (x^p, y^p)$ is an endomorphism. The minimal polynomial of π is $X^2 + p = 0$, and so we can identify $\mathbb{Z}[\sqrt{-p}]$ with a subring of the endomorphism ring $\mathrm{End}(E)$ by mapping $\sqrt{-p} \mapsto \pi$; denoting the induced map $\mathbb{Z}[\sqrt{-p}] \hookrightarrow \mathrm{End}(E)$ by ι.

Given a (primitive) ideal $\mathfrak{a} = \ell\mathbb{Z} + (\lambda + \sqrt{-p})\mathbb{Z}$ inside $\mathbb{Z}[\sqrt{-p}]$, and an elliptic curve $E/\mathbb{F}_p$, we can write down the (finite) subgroup

$$E[\mathfrak{a}] := \bigcap_{\alpha \in \mathfrak{a}} E[\iota(\alpha)] = E[\ell] \cap E[\pi + \lambda]$$

which is the kernel of a separable isogeny $\varphi_{\mathfrak{a}}\colon E \to E_{\mathfrak{a}}$ of degree $\mathrm{N}(\mathfrak{a})$. Precisely because $\ker(\varphi_{\mathfrak{a}}) = E[\mathfrak{a}]$ is stable under the action of Frobenius, we know that $\varphi_{\mathfrak{a}}$ is $\mathbb{F}_p$-rational and so $E_{\mathfrak{a}}$ is also defined over $\mathbb{F}_p$. As such, we can repeat this procedure on $E_{\mathfrak{a}}$ with another ideal $\mathfrak{b}$ to yield a new isogeny $\varphi_{\mathfrak{b}}\colon E_{\mathfrak{a}} \to (E_{\mathfrak{a}})_{\mathfrak{b}}$. We refer to [CLM+18] for further details.

The class-group action. The $\mathbb{F}_p$-isomorphism class $[E_{\mathfrak{a}}]$ of $E_{\mathfrak{a}}$ does not depend on the choice of representative of $[\mathfrak{a}]$ in $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$, and the mapping $([\mathfrak{a}], [E]) \to [E_{\mathfrak{a}}]$ constitutes a free group action. When restricting to (primitively) $\mathbb{Z}[(\pi + 1)/2]$-oriented curves (i.e. curves with $\mathbb{F}_p$-endomorphism ring $\mathbb{Z}[(\pi + 1)/2]$), the action is also transitive [CD20]. As is customary in isogeny-based cryptography, we abuse notation and write $E_{\mathfrak{a}}$ for the $\mathbb{F}_p$-isomorphism class $[E_{\mathfrak{a}}]$. This notion can be generalised on supersingular curves via the theory of *orientations* [CK20, Onu20], with original results due to Waterhouse [Wat69, Th. 4.5].

Class numbers. Due to a result of Littlewood [Lit28, Th. 1], the *class number* $h(\mathcal{O}_K) = |\mathrm{Cl}(\mathcal{O}_K)|$ is known to have asymptotic behaviour

$$\frac{\pi + o(1)}{12e^{\gamma}} \cdot \frac{1}{\log\log(\Delta_K)} < \frac{h(\mathcal{O}_K)}{\sqrt{-\Delta_K}} < \frac{2e^{\gamma} + o(1)}{\pi} \cdot \log\log(\Delta_K),$$

where $\gamma \approx 0.6$ is the *Euler-Mascheroni* constant; this refines Siegel's result $\log(h(\mathcal{O}_K)) \sim \log(\sqrt{-\Delta_K})$ [Sie35]. Since $K = \mathbb{Q}(\sqrt{-p})$ has discriminant $\Delta_K = -p$ when $p = 3 \pmod 4$, the class-group action is suitable for cryptographic purposes when p is sufficiently large.

For example, CSI-FiSH [BKV19] computed the class number of the class-group attached to CSIDH [CLM+18] to have approximately 257.136 bits. Recall that in CSIDH, p has approximately 510.668 bits and so the class number slightly exceeds $\sqrt{-\Delta_K} = \sqrt{p}$. This phenomenon can be heuristically explained by the fact that the CSIDH primes have many many small split primes by design [Cas21].

We are also able to relate the class number of a conductor-f sub-order $\mathcal{O}_f$ to the class number of $\mathcal{O}_K = \mathcal{O}_1$ via the formula

$$h(\mathcal{O}_f) = h(\mathcal{O}_K) f \prod_{q \text{ prime} | f} \left(1 - \frac{1}{q}\left(\frac{\Delta_K}{q}\right)\right)$$

when $\Delta_K < -3$ [Cox13, Th. 7.24, Ex. 5.9], where $(\cdot/\cdot)$ is the Kronecker symbol. For the orders $\mathbb{Z}[\sqrt{-p}]$ of conductor 2 inside the quadratic imaginary number field $K = \mathbb{Q}(\sqrt{-p})$ with $p = 7 \pmod 8$ prime, we observe that $h(\mathbb{Z}[\sqrt{-p}]) = h(\mathcal{O}_K)$ (in fact, the class groups are isomorphic [FHL+26, Lem. 4.3]).

2.4 Clapoti, (qt-)PEGASIS, and KLaPoTi

In Clapoti [PR23], Page and Robert introduced a new framework for computing isogeny class-group actions time polynomial in $\log(p), \log(\mathrm{disc}(\mathcal{O}))$.

The central building block of the Clapoti construction [PR23, Prop. 2.1], lies in showing that the isogeny class-group action of the ideal $[\mathfrak{c}]$ can be computed efficiently using two-dimensional N-isogenies, as long as we can find equivalent ideals $\mathfrak{a}, \mathfrak{b} \in [\mathfrak{c}]$ such that the $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$-torsion is smooth and accessible.

However, since a general procedure to find such ideals does not appear to be available, they instead formally prove in [PR23, Prop. 2.7] that the isogeny class-group action can always be computed in asymptotically polynomial time using eight-dimensional N-isogenies.

The core idea is to compose $\varphi_\mathfrak{a}, \psi_\mathfrak{a}$ with $\mathcal{U}$ and $\varphi_\mathfrak{b}, \psi_\mathfrak{b}$ with $\mathcal{V}$, where $\mathcal{U}$ and $\mathcal{V}$ are 4×4 polarized integer matrices obtained by *Zarhin's trick* [Zar74] with determinants u, v respectively. The resulting four-dimensional Kani-square yields an eight-dimensional isogeny of polarized degree $N = u\mathrm{N}(\mathfrak{a}) + v\mathrm{N}(\mathfrak{b})$. Since the determinants u, v are expressed as sums of 4 squares of the entries of $\mathcal{U}, \mathcal{V}$, any $N = u\mathrm{N}(\mathfrak{a}) + v\mathrm{N}(\mathfrak{b}) \geq \mathrm{N}(\mathfrak{a})\mathrm{N}(\mathfrak{b})$ can be achieved. In particular, powersmooth N such that the N-torsion is accessible can be obtained.

Even though this construction is asymptotically efficient, its reliance on eight-dimensional isogenies makes it impractical for concrete parameters. Two relevant approaches have been given in the literature to circumvent this issue, giving the following constructions.

PEGASIS and qt-PEGASIS. By employing smooth one-dimensional isogenies and 2×2 polarized integer matrices, PEGASIS could efficiently compute in practice the action by $[\mathfrak{a}]$ in dimension-4, by finding an equivalent $\mathfrak{b}$ such that $(a^2 + b^2)s\mathrm{N}(\mathfrak{a}) + (a'^2 + b'^2)s'\mathrm{N}(\mathfrak{b}) = 2^e$, where s, s' are smooth and $e < \log(p)$. Finally, qt-PEGASIS [DEIV25] removed the need for the smooth one-dimensional isogenies, by directly finding norm equation solutions to

$$\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) + \mathrm{N}(\mathfrak{c}) + \mathrm{N}(\mathfrak{d}) = 2^e,$$

with $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{d}$ equivalent ideals, resulting in an algorithm that is roughly twice as fast as PEGASIS.

KLaPoTi. Another approach to finding equivalent ideals $\mathfrak{a}, \mathfrak{b}$ such that $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$ is to directly solve this norm equation using the KLPT [KLPT14] algorithm, as done in KLaPoTi [PPS25]. However, this approach can only find solutions for quite large values e thus requiring a larger field $\mathbb{F}_p$ than in other class-group action instantiations to achieve the same security level. These larger field impacts the efficiency of the resulting isogeny evaluation. For example, an optimized Rust implementation for the smallest security level of 512-bit primes requires 2.5 s according to [PPS25].

3 The CORAL group action

We now describe the CORAL group action. Let us fix $p = c2^f - 1$ to be a large prime with c a small odd cofactor. The CORAL group action is a (restricted) group action whose set elements are the supersingular elliptic curves defined over $\mathbb{F}_p$ with rational 2-torsion. In the $\mathbb{F}_p$-isogeny graph, whose shape is described as a volcano (e.g. [CD20, Fig. 2]), these are the curves that are on the surface. We thus denote such a set by $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$, for they are the curves with $\mathbb{F}_p$-endomorphism ring $\mathbb{Z}[(\pi+1)/2]$.

These curves, since they are defined over $\mathbb{F}_p$, all have the Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$. Thus, the quadratic order $\mathbb{Z}[\pi]$ embeds into $\mathrm{End}(E)$, which means every curve in $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$ is $\mathbb{F}_p$-oriented. As a consequence, the class group $\mathrm{Cl}(\mathbb{Z}[\pi])$ acts on $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$ via the oriented isogeny class-group action, as described in Section 3.3. The group in the CORAL group action is a subset of $\mathrm{Cl}(\mathbb{Z}[\pi])$ (as described in the following section), acting on the curves in $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$ via the isogeny class-group action.

To efficiently compute the action corresponding to a class $[\mathfrak{c}]$ in $\mathrm{Cl}(\mathbb{Z}[\pi])$, [PR23, Prop. 2.1] requires finding two integral $[\mathfrak{c}]$-representatives $\mathfrak{a}, \mathfrak{b}$ with coprime norms, such that the $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$ torsion on E is defined over $\mathbb{F}_{p^2}$. For concrete efficiency, we further impose that $N = 2^e \mid p + 1$, so that the corresponding two-dimensional isogeny can be decomposed into a chain of 2-isogenies, while the N-torsion is still $\mathbb{F}_{p^2}$-rational. [7] Efficiently finding $\mathfrak{a}, \mathfrak{b}$ for a given $[\mathfrak{c}]$, subject to these constraints, appears to be difficult.

Nevertheless, writing down the sets

$$\begin{aligned}
\mathcal{P}_e &= \left\{(\mathfrak{a}, \mathfrak{b}) \mid [\mathfrak{a}] = [\mathfrak{b}],\, \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e,\, \mathrm{N}(\mathfrak{a}) < \mathrm{N}(\mathfrak{b})\right\} \\
\mathcal{I}_e &= \left\{\mathfrak{a} \mid \exists \mathfrak{b} : (\mathfrak{a}, \mathfrak{b}) \in \mathcal{P}_e \text{ or } (\mathfrak{b}, \mathfrak{a}) \in \mathcal{P}_e\right\}, \\
\mathcal{C}_e &= \left\{[\mathfrak{a}] \mid \mathfrak{a} \in \mathcal{I}_e\right\},
\end{aligned}$$

we parametrise the elements of $\mathrm{Cl}(\mathbb{Z}[\pi])$ that can be efficiently evaluated in practice by [PR23, Prop. 2.1], once $(\mathfrak{a}, \mathfrak{b})$ in $\mathcal{P}_e$ representing $[\mathfrak{c}]$ is found. However, instead of finding $\mathfrak{a}, \mathfrak{b}$ to evaluate a given $[\mathfrak{c}]$, we can directly sample pairs $(\mathfrak{a}, \mathfrak{b})$ from $\mathcal{P}_e$, to evaluate $[\mathfrak{a}]$; because we no longer control the class $[\mathfrak{c}] = [\mathfrak{a}] = [\mathfrak{b}]$, we obtain a *restricted* group action.

[7] By working over an $\mathbb{F}_p$-curve and its twist, it is possible to represent the full N-torsion over $\mathbb{F}_p$. See Section B on computation over $\mathbb{F}_p$.

3.1 Sampling in $\mathcal{P}_e$

We now give a high-level overview on the ideal sampling in CORAL. We start by showing that sampling a CORAL ideal is equivalent to sampling an $\mathbb{F}_p$-endomorphism of the right degree.

Lemma 3.1. *Every pair $(\mathfrak{a}, \mathfrak{b})$ in $\mathcal{P}_e$ corresponds to an endomorphism γ of norm $\mathrm{N}(\mathfrak{a})\mathrm{N}(\mathfrak{b}) = \mathrm{N}(\mathfrak{a})(2^e - \mathrm{N}(\mathfrak{a}))$ via $\mathfrak{a}\overline{\mathfrak{b}} = \gamma\mathbb{Z}[\pi]$. Conversely, every endomorphism γ in $\mathbb{Z}[\pi]$ of odd norm $q(2^e - q)$, yields a pair of equivalent ideals $\mathfrak{a} = (q, \gamma)\mathbb{Z}[\pi], \mathfrak{b} = (2^e - q, \overline{\gamma})\mathbb{Z}[\pi]$ in $\mathcal{P}_e$ with coprime norms $\mathrm{N}(\mathfrak{a}) = q, \mathrm{N}(\mathfrak{b}) = 2^e - q$.*

Proof. The norm of $\mathfrak{a} = (q, \gamma)\mathbb{Z}[\pi]$ is verified to be q by first noting that $q\mathbb{Z}[\pi] \subseteq (q, \gamma)\mathbb{Z}[\pi] \subseteq \mathbb{Z}[\pi]$ and so $(\mathbb{Z}[\pi]/q\mathbb{Z}[\pi])/((q, \gamma)\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]) \cong \mathbb{Z}[\pi]/(q, \gamma)\mathbb{Z}[\pi]$; then by computing $\#\mathbb{Z}[\pi]/q\mathbb{Z}[\pi] = q^2$ and $\#((q, \gamma)\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]) = q$, we conclude that $\#\mathbb{Z}[\pi]/(q, \gamma)\mathbb{Z}[\pi] = \mathrm{N}((q, \gamma)\mathbb{Z}[\pi]) = q$. The equality $\#((q, \gamma)\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]) = q$ comes from observing that the multiplication-by-γ map on $\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]$ has image $\gamma\mathbb{Z}[\pi]/q\mathbb{Z}[\pi] \cong (q, \gamma)\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]$. Since `SampleIdeal` enforces that q divides the norm of γ which equals $\gamma\overline{\gamma}$, the multiplication-by-γ map on $\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]$ has kernel containing $\overline{\gamma}\mathbb{Z}[\pi]/q\mathbb{Z}[\pi] \cong \gamma\mathbb{Z}[\pi]/q\mathbb{Z}[\pi]$. By counting, both kernel and image have cardinality q. Moreover, $\mathfrak{a}\overline{\mathfrak{b}} = (q(2^e - q), q\gamma, (2^e - q)\gamma, \gamma^2)\mathbb{Z}[\pi] \supseteq \gamma\mathbb{Z}[\pi]$ since q is prime to $2^e - q$. Since $\mathrm{N}(\mathfrak{a}\overline{\mathfrak{b}}) = \mathrm{N}(\gamma)$, we now conclude that $\gamma\mathbb{Z}[\pi] = \mathfrak{a}\overline{\mathfrak{b}}$ and so $\mathfrak{a}, \mathfrak{b}$ represent the same class in $\mathrm{Cl}(\mathbb{Z}[\pi])$.

Therefore, to sample from $\mathcal{P}_e$ is to sample a $\gamma = x + y\pi$ in $\mathbb{Z}[\pi]$ with $\mathrm{N}(\gamma) = x^2 + y^2 p = q(2^e - q)$. Changing variables $q = 2^{e-1} - a$ and rearranging, we obtain the norm equation

$$2^{2(e-1)} - y^2 p = x^2 + a^2, \tag{4}$$

We can find solutions to this norm equation by sampling a random y until $2^{2(e-1)} - y^2 p$ is a sum of two squares. We sample y to be odd, so that one of a, x is odd when a solution is found. In turn, this allows us to enforce a to be odd, which guarantees q is also odd and $\mathfrak{a}, \mathfrak{b}$ have coprime norm. This approach is represented in the `SampleIdeal` algorithm.

Algorithm 1 `SampleIdeal`(p, e)

Input: A prime p, an integer $e > \log(p)/2$
Output: An element $(\mathfrak{a}, \mathfrak{b})$ in $\mathcal{P}_e$

repeat
  $y' \xleftarrow{\$} \{0, \ldots, \lfloor 2^{e-3}/\sqrt{p} \rfloor - 1\}$ ▷ *Non-empty since $e > \log p/2$*
  $y \leftarrow 2y' + 1$ ▷ *Sample an odd y in $[1, \lfloor 2^{e-2}/\sqrt{p} \rfloor]$*
  $x, a \leftarrow$ `SumOfTwoSquares`$(2^{2(e-1)} - y^2 p)$
until $(x, a) \neq \perp$
if $a \equiv 0 \pmod 2$ **then** ▷ *Recall, one of a, x must be odd*
  $x, a \leftarrow a, x$ ▷ *Ensure a odd*
$q \leftarrow 2^{e-1} - a$ ▷ *Hence q is odd*
return $\mathfrak{a} = (q, x + y\pi)\mathbb{Z}[\pi], \mathfrak{b} = (2^e - q, x - y\pi)\mathbb{Z}[\pi]$ ▷ *Odd coprime norms*

At the core of SampleIdeal, there is the SumOfTwoSquares algorithm (a special case of *Cornacchia's algorithm* [Cor08]), which decomposes suitable odd $L = 2^{2(e-1)} - y^2 p$ into the sum of two squares. The choice of SumOfTwoSquares determines the efficiency of SampleIdeal: besides being the most computational demanding subroutine, its success probability determines the number of y samples.

Decomposing a number into a sum of two squares requires factoring the number itself. Obtaining a full factorization is infeasible for integers of size $2 \log p$; our SumOfTwoSquares thus needs to work with a partial factorization. If SumOfTwoSquares only worked for prime $L \equiv 1 \pmod 4$, the SampleIdeal algorithm would already be guaranteed to succeed;[8] however, the number of iterations until L is prime would be considerable, which would slow down SampleIdeal.

Our approach relies on a partial factorisation: the odd integer L is decomposed into the product $L = L_{\mathsf{good}} L_{\mathsf{bad}} L'$, where L_{good} is the product of small primes that are 1 (mod 4) and L_{bad} is the product of small primes that are 3 (mod 4). If L_{bad} is not a square, the algorithm rejects y and L and starts a new iteration; otherwise we know L' is 1 (mod 4). Thus, we finally check L' is 1 (mod 4) and a prime and if so, we decompose it into a sum of two squares. We detail our approach in Section A, where we also describe additional improvements.

Remark 3.2. In QFESTA, [NO24, Alg. 2] samples arbitrary-degree isogenies from a curve whose full endomorphism ring is known, by finding endomorphisms of a suitable norm and splitting it to recover the desired isogeny. Our sampling of ideal pairs in $\mathcal{P}_e$ follows an analogous strategy in the quadratic order $\mathbb{Z}[\pi]$: we sample endomorphisms $\gamma \in \mathbb{Z}[\pi]$ of the right norm and extract the corresponding ideal pair via Lemma 3.1. The key difference is that working within $\mathbb{Z}[\pi]$ rather than the full endomorphism ring avoids the need to know $\mathrm{End}(E)$, and thus our approach works for any supersingular curve over $\mathbb{F}_p$.

Since specific implementations of SumOfTwoSquares affect which ideals can be sampled, we define the set $\mathcal{I}_e^{\mathtt{sots}}$ of possible output ideals, and then we show that SampleIdeal produces ideals that are uniformly distributed in $\mathcal{I}_e^{\mathtt{sots}}$.

Definition 3.3. *For a given exponent e, prime p, and a specific instantiation of* SumOfTwoSquares*, we denote by $\mathcal{I}_e^{\mathtt{sots}} \subset \mathcal{I}_e$ the set of ideals $\mathfrak{a}$ such that $(\mathfrak{a}, \mathfrak{b})$ is a possible output of* SampleIdeal*.*

Lemma 3.4. *The distribution of the output of* SampleIdeal *is uniform over $\mathcal{I}_e^{\mathtt{sots}} \subseteq \mathcal{P}_e$ for any given deterministic instantiation of* SumOfTwoSquares*.*

Proof. First, we see that in the initial loop of SampleIdeal, from line 1 to line 5, we are performing rejection sampling on the set of y such that $2^{2(e-1)} - y^2 p$ is representable as a sum of two squares by the SumOfTwoSquares instantiation

[8] By Jacobi's two-square theorem an odd integer is decomposable into a sum of two squares if and only if its prime factorizations includes only primes which are 1 (mod 4) or the prime divisors that are 3 (mod 4) appear with an even exponent.

used. Given such a y, since we are considering a deterministic instantiation, we obtain a unique solution (x, a) to the norm equation from `SumOfTwoSquares` that gives a unique pair $(\mathfrak{a}, \mathfrak{b})$ in $\mathcal{P}_e$.

Because y is sampled uniformly, the distribution of y after the loop is uniform over $\mathcal{I}_e^{\texttt{sots}}$, as long as the map $y \mapsto (\mathfrak{a}, \mathfrak{b})$ from $\mathcal{I}_e^{\texttt{sots}}$ to $\mathcal{P}_e$ is injective. We now show that this is the case.

Let y_1, y_2 be such that the corresponding pairs of ideals $(\mathfrak{a}_1, \mathfrak{b}_1), (\mathfrak{a}_2, \mathfrak{b}_2) \in \mathcal{P}_e$ are equal. Let x_1, a_1 and x_2, a_2 be the corresponding solutions of the norm equation, with a_1, a_2 odd. This implies $\mathrm{N}(\mathfrak{a}_1) = \mathrm{N}(\mathfrak{a}_2)$, and so $q_1 = q_2$ and $a_1 = a_2$. From Equation (4), we have $x_1^2 + py_1^2 = x_2^2 + py_2^2$, and so $x_1 = \pm x_2 \pmod p$ since p is prime. Now, since $2^{2(e-1)} \leq (p/2)^2$ by Equation (4) again we have $0 < x_1, x_2 < p/2$. This implies that also $x_1 = x_2$, and so $y_1 = y_2$ since $y_1, y_2 > 0$[9].

3.2 Computing the CORAL group action

The `SampleIdeal` algorithm produces pairs $(\mathfrak{a}, \mathfrak{b})$ of equivalent ideals, and by projecting $(\mathfrak{a}, \mathfrak{b}) \to [\mathfrak{a}]$ we obtain a distribution in the class-group $\mathrm{Cl}(\mathbb{Z}[\pi])$. For fixed inputs p, e we denote this distribution $\mathcal{D}_e^{\mathrm{Cl}}$.

To evaluate the action by $[\mathfrak{a}] = [\mathfrak{b}]$ obtained from $(\mathfrak{a}, \mathfrak{b}) \leftarrow \texttt{SampleIdeal}(p, e)$, we embed into the two-dimensional $(\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e)$-isogeny

$$\Phi = \begin{pmatrix} \varphi_{\mathfrak{a}} & \widetilde{\psi'_{\overline{\mathfrak{b}}}} \\ -\psi_{\overline{\mathfrak{b}}} & \widetilde{\varphi'_{\mathfrak{a}}} \end{pmatrix} : E \times E \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}} \quad \text{via the Kani-square} \quad \begin{array}{ccc} E & \xrightarrow{\varphi_{\mathfrak{a}}} & E_{\mathfrak{a}} \\ {\scriptstyle \psi_{\overline{\mathfrak{b}}}}\downarrow & & \downarrow{\scriptstyle \psi'_{\overline{\mathfrak{b}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\varphi'_{\mathfrak{a}}]{} & E \end{array}$$

By construction, $\mathfrak{a}\overline{\mathfrak{b}} = (x + y\pi)\mathbb{Z}[\pi]$ and so we can efficiently compute

$$\ker(\Phi) = \{(qP, (x + y\pi)(P)) \mid P \in E[2^e]\};$$

from which we can efficiently compute Φ to obtain $E_{\mathfrak{a}} = [\mathfrak{a}] * E$ from its codomain.

The isogeny Φ decomposes as a chain $\Phi = \Phi_e \circ \cdots \Phi_1$ of e two-dimensional 2-isogenies Φ_i [Rob25, Prop. 1]: it begins with an endomorphism Φ_1 of $E \times E$; is followed by a sequence of $\delta \geq 1$ *diagonal isogenies* between products of elliptic curves $E_1^{(i)} \times E_2^{(i)} \to E_1^{(i+1)} \times E_2^{(i+1)}$; then a *gluing* into a simple Abelian surface $E_1^{(\delta+1)} \times E_1^{(\delta+1)} \to A^{(\delta+2)}$; which is followed by $e-2-\delta$ *generic* isogenies between simple Abelian varieties $A^{(i)} \to A^{(i+1)}$; and finally a splitting $A^{(e-1)} \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$. This is depicted in Fig. 1. The number δ of diagonal isogenies is given by

$$\delta = \max(v_2(x + y \pm q)) - 1 = \max(v_2(x - y \pm q)) - 1 \tag{5}$$

Notably, this depends only on the secret key and not the curve being acted upon.

[9] This also implies that the ideals $\mathfrak{a}_1, \mathfrak{a}_2 \in \mathcal{I}_e$ are uniquely identified by their norms, once obtained as an output of `SampleIdeal`.

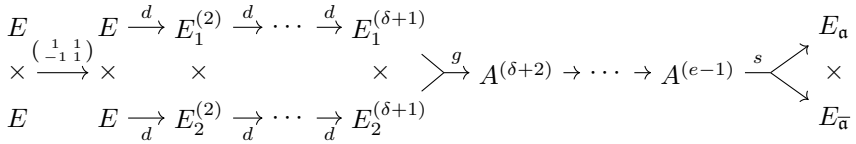

Fig. 1: Decomposition of the two-dimensional 2^e-isogeny $E \times E \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$ into e two-dimensional 2-isogenies. Here d, g, s denote "diagonal", "gluing" and "splitting isogenies" respectively.

Compatibility with (qt-)PEGASIS. We emphasise the compatibility between CORAL and (qt-)PEGASIS: unlike other isogeny REGA [CLM+18,Hou25], CORAL and (qt-)PEGASIS compute the same isogeny class-group action on $\mathcal{E}(\mathbb{Z}[(\pi + 1)/2])$ and use the same parameters (both constructions use a prime p where $p + 1$ is divisible by a large power of 2, which is suitable for efficient isogeny computation in higher dimensions). They can thus be combined to exploit the efficiency of CORAL and the unrestrictedness of (qt-)PEGASIS: for example, in the standard group-action based identification scheme, the signer can compute the commitment $[\mathfrak{c}] * E_{\mathsf{pk}}$ using CORAL, while the verifier can recompute the commitment using CORAL and evaluate the $[\mathfrak{c}]^{-1}[\mathsf{sk}]$ action using (qt-)PEGASIS.

3.3 More general orientations

For simplicity, we have only considered the Frobenius orientation until now. To extend our techniques to an arbitrary orientation, we note that CORAL relies only on two properties of the Frobenius endomorphism:

- The trace of π is zero, which means $\mathrm{N}(x + y\pi) = x^2 + y^2\mathrm{N}(\pi) = x^2 + y^2 p$ has no mixed xy terms. Thus, we obtain Eq. (4), which can be solved easily to sample CORAL ideals.
- The ratio $2^{e-1}/\sqrt{\mathrm{N}(\pi)}$ is exponentially large in λ for some e such that $2^e \mid p + 1$, which guarantees that we are sampling from an exponentially large subset of the class group, up to a polynomial factor given by number of repetitions (this is discussed in detail in Section 4.1).

Thus, our techniques can be generalized to work with any K-orientation encoded by a trace-zero endomorphism ω of norm $< p^2/2^{2\lambda}$. Even more generally, if ω encodes the primitive $\mathcal{O}$-orientation ι, then the trace-zero endomorphism $\omega_0 = 2\omega - \mathrm{tr}(\omega)$ also encodes ι.

In the remainder of the paper, we treat the case of the Frobenius orientation, which is particularly beneficial for implementation efficiency.

4 Security analysis

Recall that we consider $p = c2^f - 1$ for some small odd integer c and let $e \in (1/2 \log p + 2, f)$ be a parameter. Since we restrain our analysis to the Frobenius

orientation, CORAL computes the same class group action as in (qt-)PEGASIS or CSIDH, so the security of our group action relies fundamentally on the same assumptions. However, we are considering a smaller set of ideals to sample from, as done for example in SQALE [CSCJR22], thus we consider the following problem:

Problem 4.1 (Group Action Inversion in $\mathcal{I}_e^{\mathtt{sots}}$). On input an elliptic curve E in $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$ and an elliptic curve $E' = [\mathfrak{a}] * E$ for $\mathfrak{a} \xleftarrow{\$} \mathcal{I}_e^{\mathtt{sots}}$, return the ideal class $[\mathfrak{a}]$.

In Section 4.1, we study the distribution $\mathcal{D}_e^{\mathrm{Cl}}$ of the classes $[\mathfrak{a}] \in \mathrm{Cl}(\mathbb{Z}[\pi])$ for $\mathfrak{a} \leftarrow \mathcal{I}_e^{\mathtt{sots}}$. We argue that $\mathcal{D}_e^{\mathrm{Cl}}$ heuristically samples at random from an unstructured and exponentially-large subset of the whole group, which provides no obvious structure to exploit for an attack specific to CORAL. Thus, CORAL provides the same security as the class group action, as computed in CSIDH and (qt-)PEGASIS. We discuss potential classical and quantum attacks in Sections 4.2 and 4.3, then we show how to set parameters according to the previous discussion in Section 4.4. We then provide experimental evidence for these claims in Section 4.5.

4.1 Distribution in the class group

We first analyse the size of the set $\mathcal{I}_e^{\mathtt{sots}}$, which depends on the number of valid y-samples for the instantiation of the algorithm `SumOfTwoSquares` in `SampleIdeal`. Recall that we are sampling y from the set of odd integers smaller than $2^{e-1}/\sqrt{p}$. Under the heuristic assumption that $2^{2(e-1)} - y^2 p$ behaves like a random number of $2(e-1)$ bits that is 1 (mod 4), we can estimate the expected size for $\mathcal{I}_e^{\mathtt{sots}}$ by looking at the probability that $L = 2^{2(e-1)} - y^2 p$ returns a valid output for `SumOfTwoSquares`. We consider two choices of `SumOfTwoSquares`, an optimal one and a pessimistic one to give bounds on the size of $\mathcal{I}_e^{\mathtt{sots}}$.

- Lower bound. `SumOfTwoSquares` can always efficiently enforce that $L = 2^{2(e-1)} - y^2 p$ is a prime number. In this case, with the Prime Number Theorem we can estimate the probability that a random odd $2(e-1)$ bits integer L is prime as $((e-1)\ln(2))^{-1}$, and so the expected size of $\mathcal{I}_e^{\mathtt{sots}}$ is at least

$$\frac{2^{e-2}}{\sqrt{p} \cdot (e-1)\ln(2)} \,. \tag{6}$$

- Upper bound. Even the best imaginable `SumOfTwoSquares` algorithm cannot compute a sum of two squares decomposition of an odd integer L if no such decomposition exists. Indeed, the number of integers $\leq X$ representable as a sum of two squares is known to be $\approx \gamma X/\sqrt{\ln(X)}$ for $\gamma \approx 0.76$ the Landau–Ramanujan constant. Therefore, we can upper-bound the size of $\mathcal{I}_e^{\mathtt{sots}}$ as:

$$\gamma \frac{2^{e-2}}{\sqrt{p \cdot 2(e-1)\ln(2)}} \,. \tag{7}$$

In practice, we found that for small values of e, when we can factor integers easily, the size of $\mathcal{I}_e^{\texttt{sots}}$ is slightly bigger than the bound in Eq. (7), due to the fact that we sample integers $1 \pmod 4$, which are slightly more likely to be representable as a sum of two squares.

We give a precise instantiation of `SumOfTwoSquares` in Section A, in which we estimate the number of repetitions needed to find a valid y-sample, that allows us to estimate the size of $\mathcal{I}_e^{\texttt{sots}}$.

Distribution of $\mathcal{I}_e^{\texttt{sots}}$ in the class group $\mathrm{Cl}(\mathbb{Z}[\pi])$. In principle, it could be the case that a large number of ideals in $\mathcal{I}_e^{\texttt{sots}}$ lie in a small number of classes. We give a heuristic argument relying on the theory of representations of integers by binary quadratic forms, that this is not the case.

First, as shown in the proof of Lemma 3.4, the ideals in $\mathcal{I}_e^{\texttt{sots}}$ are uniquely identified by their norm. Also, the classes that can potentially contain a specific ideal $\mathfrak{a} \in \mathcal{I}_e^{\texttt{sots}}$ are those whose associated class of binary quadratic forms represents $\mathrm{N}(\mathfrak{a}) = q$, but also $2^e - q$, by the definition of $\mathcal{I}_e$ and $\mathcal{P}_e$. Heuristically, a class $[\mathfrak{c}]$ can potentially contain more ideals in $\mathcal{I}_e$ if its associated class of quadratic form represents more pairs of integers $(q, 2^e - q)$ with q odd and smaller than 2^{e-1}. As shown in [BG06], given a fixed class $[\mathfrak{c}] \in \mathrm{Cl}(\mathbb{Z}[\pi])$, the number of distinct integers below a bound $X = p^{\varepsilon}$ represented by (the form of) $[\mathfrak{c}]$ is $N_{[\mathfrak{c}]}(X) = \Theta\left(X/\sqrt{p}\right)$, where the implied constants bounding $c_\pi < N_{[\mathfrak{c}]}(X)\sqrt{p}/X < C_\pi$ depend only on the order $\mathbb{Z}[\pi]$ [BG06, Eq. 1.3] and not on the class $[\mathfrak{c}]$. So, under the heuristic assumption that the probabilities of q and $2^e - q$ being represented by $[\mathfrak{c}]$ are independent, we can expect that no class $[\mathfrak{c}]$ is more likely to contains ideals in $\mathcal{I}_e$ than any other. Note: it is not possible to define a proper notion of probability here, since for fixed parameters p, e, these events are deterministic. However, we can still use this heuristic to give an intuition on the way ideals in $\mathcal{I}_e^{\texttt{sots}}$ are distributed in the class group. So, by the previous discussion, we can formulate the following heuristic:

Heuristic 4.2. *Fix e, p such that $e - 2 > \log(p)/2$ and $2^e \mid p + 1$. Then the classes of the ideals in $\mathcal{I}_e^{\texttt{sots}}$ distribute uniformly and independently in* $\mathrm{Cl}(\mathbb{Z}[\pi])$.

With Heuristic 4.2, we infer the characteristics of the distribution $\mathcal{D}_e^{\mathrm{Cl}}$ using combinatorial arguments. Let $h \approx \sqrt{p}$ be the class number of $\mathbb{Z}[\pi]$ and $t = |\mathcal{I}_e^{\texttt{sots}}|$.

1. Since the sampled classes are uniform and independent, the support of $\mathcal{D}_e^{\mathrm{Cl}}$ has expected size

$$C_1 = h\left(1 - \left(1 - \frac{1}{h}\right)^t\right) \approx h\left(1 - e^{-t/h}\right), \tag{8}$$

which is the expected number of unique coupons collected when collecting t coupons from a box of h coupons.

2. Also, the probability that there exists no "collision class", that is a class $[\mathfrak{c}]$ such that $[\mathfrak{c}] = [\mathfrak{a}_1] = [\mathfrak{a}_2]$ for two distinct ideals $\mathfrak{a}_1, \mathfrak{a}_2 \in \mathcal{I}_e^{\mathtt{sots}}$, is

$$C_2 = \prod_{i=0}^{t-1} \left(1 - \frac{i}{h}\right) \approx e^{-t(t-1)/2h}, \tag{9}$$

which is the probability of *no* collision when collecting t coupons from a box of h coupons.

We get the important consequence that if $t \ll \sqrt{h}$, then with high probability there are no collision classes, and so the support of $\mathcal{D}_e^{\mathrm{Cl}}$ is of size t and $\mathcal{D}_e^{\mathrm{Cl}}$ is uniform on its support.

When $\sqrt{h} < t$, we cannot rule out the existence of collision classes; we need to refine the previous results with a classical binomial distribution argument. The support of $\mathcal{D}_e^{\mathrm{Cl}}$ can be partitioned into a finite number of blocks, where the k-th block contains the classes containing k different ideals in $\mathcal{I}_e^{\mathtt{sots}}$, with a probability mass of k/t in $\mathcal{D}_e^{\mathrm{Cl}}$. The expected size of the k-th block is given by

$$C_3(k) = h \cdot \binom{t}{k} \cdot \left(\frac{1}{h}\right)^k \left(1 - \frac{1}{h}\right)^{t-k}. \tag{10}$$

4.2 Classical security

We now discuss the classical security of Problem 4.1 as a function of the size of the support of $\mathcal{D}_e^{\mathrm{Cl}}$, which, as we have seen in the previous discussion, is closely related to the size of $\mathcal{I}_e^{\mathtt{sots}}$ and the parameters p, e. The best known classical attacks to solve the more general Problem 2.2 are based on meet-in-the-middle graph walks or memory-bounded variants such as the van Oorschot–Wiener (vOW) or Golden Collision Search (GCS) [ACC+19, vOW99].

This thus indicates that the naive choice of a key-space of size larger than $2^{2\lambda}$ is enough to achieve λ bits of security. More arguments on the trade-offs between memory and complexity are given in [CSCJR22, Section 5.2]. However, we argue that meet-in-the-middle attacks cannot leverage the additional structure of the set $\mathcal{I}_e$: that is, they still run in time $\sqrt{h}$, where h is the class number of $\mathbb{Z}[\pi]$. To argue this we revisit the meet-in-the-middle attack in the context of the isogeny class group action. Assuming $E' = [\mathfrak{a}] * E$ for some $\mathfrak{a} \in \mathcal{I}_e^{\mathtt{sots}}$, the natural way to perform a meet-in-the-middle attack is to consider two subsets A, B of the class group $\mathrm{Cl}(\mathbb{Z}[\pi])$, and proceed as follows:

1. Start from E and perform random walks in the isogeny graph, computing $[a] * E$ for $a \in A$, and store the results in a list $\mathcal{L}_A$.
2. Start from E' and perform random walks in the isogeny graph, computing $[b] * E'$ for $b \in B$, and store the results in a list $\mathcal{L}_B$, then check whether $\mathcal{L}_A \cap \mathcal{L}_B \neq \emptyset$. From any collision we extract a, b such that $[a] * E = [b] * E'$, and so $E' = [ab^{-1}] * E$.[10]

[10] Technically, we can avoid storing the whole list $\mathcal{L}_B$ and instead check for collisions on the fly, but this does not change the argument.

By the freeness of the group action, if there is a collision it means that $[ab^{-1}] = [\mathfrak{a}]$, or equivalently a is in the same class as $\mathfrak{a}b$. In a generic meet-in-the-middle attack, the sets A, B are random subsets of the class group, and so is $\mathfrak{a}B$; thus, they intersect with constant probability when $|A| \cdot |\mathfrak{a}B| \approx h$, that is when $|A| = |B| \approx \sqrt{h}$. To leverage that $\mathfrak{a} \in \mathcal{I}_e$ and find collisions, we would need to find an efficiently sampleable set B such that $\mathfrak{a}B$ is still contained in a significantly smaller subset C of the class group, for a large fraction of $\mathfrak{a} \in \mathcal{I}_e^{\texttt{sots}}$, such that collision can be found with less than $\sqrt{h}$ operations by sampling from $A = C$ for $\mathcal{L}_A$.

We could consider the naive choice of $B = \{id\}$, or some other very small set, so that $\mathfrak{a}B$ is contained in $\mathcal{I}_e$. However, this forces A to be nearly as large as $\mathcal{I}_e$, thus giving the same complexity as a key-enumeration attack. Apart from the naive choice, we could try to consider B, C as subsets of $\mathcal{I}_{e'}$,$\mathcal{I}_{e''}$ for some e', e'', but under Heuristic 4.2, there is no reason for $\mathcal{I}_e \cdot \mathcal{I}_{e'}$ to intersect with $\mathcal{I}_{e''}$. In general, the structure of $\mathcal{P}_e$ seems to be linked more to the norm relations between two equivalent fractional ideals than to some multiplicative structure. Since no other choice of B appears to leverage the structure of $\mathcal{I}_e$ to give a better attack, we conclude that among *meet-in-the-middle* attacks the generic one is still the best, with complexity $\sqrt{h}$. A similar argument is used to justify the usage of prime-degree isogenies in [DKL+20, Section 8.3]: prime-degree isogenies cannot be decomposed and therefore it is impossible to perform a meet-in-the-middle attack. The best known attacks are thus generic ones or exhaustive search over the key-space.

We mention two more possible attack strategies, which we show cannot apply to CORAL. The set $\mathcal{I}_e$ is a contained in the set of all ideals of norm at most 2^e: an attacker may thus choose to work within this set. In our instantiations, we choose $e > \log(p)/2$; by a direct application of the Minkowski bound, we see that all ideal classes have a representative of norm at most 2^e. Hence, the set of all classes with at least one ideal of norm at most 2^e is the entire class group, and thus this attack approach cannot offer an advantage.

Another potential attack could exploit the fact that the isogeny-class group action is *symmetric* around $E_0 : y^2 = x^3 + x$ by the twisting. That is, $[\mathfrak{a}]^{-1} * E_0 = ([\mathfrak{a}] * E_0)^t$. As such we know that $E_0 \times E_0$ and $E_\mathfrak{a} \times E_\mathfrak{a}^t$ are connected by a two-dimensional isogeny of degree $N = 2^e$, which, in turn, begs the question whether a meet-in-the-middle attack in dimension-2 is feasible. Alas, the length of such walks needs to be at least $e/2 > \log(p)/2$, and in dimension-2 there are multiple possible directions per step [CDS20]; we thus conclude that this attack is not more efficient than the generic ones. Similar arguments could be repeated for other curves close to E_0 in the isogeny graph, but the same conclusion holds.

4.3 Quantum security

For an exponentially large key-space, the best known quantum attacks to solve Problem 4.1 are based on Kuperberg's algorithm [Kup05,Kup13] for the dihedral hidden subgroup problem, as explained in [CLM+18].

By the previous discussion, and Heuristic 4.2 in particular, we assume that the sampled ideals from $\mathcal{D}_e^{\mathrm{Cl}}$ behave like random elements of the class group and are not contained in any smaller subgroup. Thus, for the same arguments given in [CSCJR22], we expect that Kuperberg's algorithm does not perform better on Problem 4.1 than on the group action inversion problem (Problem 2.2) for (qt-)PEGASIS, even if the support of $\mathcal{D}_e^{\mathrm{Cl}}$ is a strict subset of the full class group. Thus, assuming a large enough key space, the only parameter that affects the quantum security of our scheme is the size of the class group, which depends only on the base prime p.

4.4 Parameter Selection

We finally discuss the selection of the parameters p, e to securely instantiate our constructions. We remark that it is still an open question to give an precise complexity estimate of Kuperberg's algorithm for the Clapoti and (qt-)PEGASIS constructions, as done in [BLMP19, Pei20] for CSIDH. It is out of the scope of this work to give such an analysis. As usual in the literature, we therefore consider different possible choices for the bit size of the prime p defining the base field $\mathbb{F}_p$, from the most optimistic 512 bits to the more conservative 4096 bits. More precisely, we consider the same primes as in [DEIV25] of the form $p = c2^f - 1$. Such large bit sizes ensure that any generic classical attacks have negligible probability of success for $\lambda = 128$.

Thus we only need to set e so that exhaustive search attacks have negligible probability of success. To do this, we start by considering $e = \log(p)/2 + \lambda$, then:

1. We estimate the size t of $\mathcal{I}_e^{\mathtt{sots}}$ using the lower bound in Eq. (6).
2. We estimate the size of the support of $\mathcal{D}_e^{\mathrm{Cl}}$ with Eq. (8), which we require to be at least 2^λ. If not, we increase e.
3. We then estimate the probability of having collision classes with Eq. (9), and if this is negligible, we conclude that $\mathcal{D}_e^{\mathrm{Cl}}$ is uniform on an exponentially large support.
4. If the probability of a collision is not negligible, we instead compute the min-entropy of $\mathcal{D}_e^{\mathrm{Cl}}$, that is the negative logarithm of the probability mass of the most likely class in $\mathcal{D}_e^{\mathrm{Cl}}$, since min-entropy can be used as a lower bound for the security against exhaustive search attacks. Since the probability mass of a class with k ideals is k/t, we have by the Markov inequality

$$\Pr[\text{classes with at least } k \text{ ideals} \geq 1] \leq C_3(k). \tag{11}$$

We thus lower bound the min-entropy of $\mathcal{D}_e^{\mathrm{Cl}}$ by $\log t - \log k$ with k the largest integer such that $C_3(k)$ is non-negligible in λ. If the min-entropy is smaller than λ, we increase e and repeat the process.

We thus obtain the following parameters:

$$\begin{aligned} 2^{500} \cdot 27 - 1 \quad & \text{with } e = 392, \\ 2^{503} \cdot 33 - 1 \quad & \text{with } e = 394, \\ 2^{1004} \cdot 15 - 1 \quad & \text{with } e = 642, \\ 2^{2026} \cdot 51 - 1 \quad & \text{with } e = 1155, \\ 2^{4084} \cdot 63 - 1 \quad & \text{with } e = 2185. \end{aligned}$$

We remark that for all our parameter choices, with the exception of the smallest one, we expect no collision classes.

Remark 4.3 (Conservative choices). At a small additional cost (around 15%), we can make the conservative choice of setting e such that the support of $\mathcal{D}_e^{\mathrm{Cl}}$ is larger than $2^{2\lambda}$, to avoid potential new *meet-in-the-middle* attacks.

Remark 4.4 (CSIDH-like construction). It remains possible to sample secret keys as *products* of elements in $\mathcal{I}_e^{\mathtt{sots}}$ (e.g. with smaller e) to grow the key-space, *à la* CSIDH. Under the heuristic that these products behave as random classes, we get an ideal distribution that is statistically close to the uniform one on the class group, as long as $|\mathcal{I}_e^{\mathtt{sots}}|^k$ is bigger than the class group. This construction would be susceptible to meet-in-the-middle again (in the same way CSIDH is), but would likely not suffer from the same constant-time implementation issues as CSIDH, because computing evaluating the action of ideals in $\mathcal{I}_e^{\mathtt{sots}}$ costs the same for each ideal (modulo the number of diagonal isogenies). Ultimately, though, this construction is not faster, because computing one two-dimensional isogeny of length $\log(p)/2 + \lambda$ is cheaper than computing k two-dimensional 2-isogenies of length $\log(p)/2 + \lambda/k$.

4.5 Experimental analysis

We tested Heuristic 4.2 experimentally[11] for some small primes p with $\log p$ between 32 and 64 and for different values of e between $1/2 \log p$ and $\log p$. We applied the sampling procedure as in `SampleIdeal`, for all possible y-samples sequentially, and we stored the classes of the sampled ideals in $\mathcal{D}_e^{\mathrm{Cl}}$ with the corresponding, potentially multiple values of y for each class. Then we computed the size of the support of $\mathcal{D}_e^{\mathrm{Cl}}$, the number of collision classes, distinguished also by the multiplicity of the collision. We then compare the expected values $C_1, C_3(k)$ for $k = 2, 3$ with the actual values obtained from the experiment. We also computed the probability of no collision C_2 for each e and the expected size for $\mathcal{I}_e^{\mathtt{sots}}$ according to the upper bound in Eq. (7).

We also performed a classical χ^2 test to check the uniformity of $\mathcal{D}_e^{\mathrm{Cl}}$ on its support, and we computed the corresponding p-value. For each test, we sampled around $|\mathcal{I}_e^{\mathtt{sots}}|$ ideals per parameter choice, then computed the classes of the sampled ideals. As expected, as soon as we have some collisions, the distribution

[11] https://github.com/CORAL-nike/CORAL

Table 1: Experimental results for $\mathcal{D}_e^{\mathrm{Cl}}$ for $p = 2^{64} \cdot 3 - 1$ and $e \in [40, 62]$

e	$\lvert\mathcal{I}_e^{\mathtt{sots}}\rvert$ exp.	$\lvert\mathcal{I}_e^{\mathtt{sots}}\rvert$ actual	$\lvert\mathcal{D}_e^{\mathrm{Cl}}\rvert$	collisions actual	collisions exp.	C_2	$C_3(1)$ exp.	$C_3(1)$ actual	$C_3(2)$ exp.	$C_3(2)$ actual	$C_3(3)$ exp.	$C_3(3)$ actual
40	4	8	8	0	0.0	1.0	8.0	8	0.0	0	0.0	0
45	115	328	328	0	0.0	1.0	328.0	328	0.0	0	0.0	0
50	3489	4596	4596	0	0.0	1.0	4596.0	4596	0.0	0	0.0	0
51	6908	9142	9142	0	0.0	0.99	9142.0	9142	0.0	0	0.0	0
52	13680	17397	17397	0	0.0	0.98	17397.0	17397	0.0	0	0.0	0
53	27095	34328	34328	0	0.1	0.91	34327.8	34328	0.1	0	0.0	0
54	53677	38549	38548	1	0.1	0.89	38548.8	38547	0.1	1	0.0	0
55	106355	266824	266816	8	5.5	0.0	266813.1	266808	5.5	8	0.0	0
56	210768	260109	260106	3	5.2	0.01	260098.6	260103	5.2	3	0.0	0
57	417754	510332	510319	13	20.0	0.0	510292.0	510306	20.0	13	0.0	0
58	828148	1003786	1003711	75	77.3	0.0	1003631.4	1003636	77.3	75	0.0	0
59	1641955	1969073	1968781	292	297.4	0.0	1968478.2	1968489	297.4	292	0.0	0
60	3255960	3869878	3868731	1147	1148.7	0.0	3867580.8	3867584	1148.3	1147	0.2	0
61	6457427	7604277	7599937	4340	4434.6	0.0	7595409.6	7595599	4431.1	4336	1.7	2
62	12808557	9510731	9503821	6910	6936.2	0.0	9496861.9	9496913	6929.5	6906	3.4	2

is not uniform anymore, and the χ^2 test rejects the null hypothesis of uniformity with overwhelming probability. We remark, as usual in statistical hypothesis testing, that all the previous experimental analysis and statistical tests only show that we have no reason to reject the assumption implied by Heuristic 4.2, but are not meant to be proofs. The justification lies in the heuristic arguments given in Section 4.1.

Additionally, for $p = 33 \cdot 2^{503} - 1$, we verified that the sampling algorithm returns generators of the class group with overwhelming probability, corroborating the heuristic assumption that the sampled ideals are not contained in any structured subgroup of the class group, as argued in Section 4.3. For this, we used the data reported in [Oud25]. This is consistent with the expectation that the class group is cyclic. Moreover, if we could sample ideals in a proper subgroup constructively, we could simplify the class number problem for $\mathbb{Z}[\pi]$, which seems unlikely given the current state of the art in class number computation.

5 Implementation results

We implemented CORAL in SageMath and in C[12]. The latter is based on the NIST round 2 implementation of SQIsign [AAA+24] and relies on the GMP library for the ideal sampling implemention (the multi-word integers functionality of GMP is used to solve the norm equation), but our implementation of the group action evaluation is in pure C. In the following, we evaluate the performance of the scheme and explain details of the implementations which helped to achieve it. We then discuss in Section 5.2 the performance of a CORAL-based NIKE protocol.

[12] Our implementations are available at `https://github.com/CORAL-nike/CORAL`.

Table 2: Uniformity tests for $\mathcal{D}_e^{\mathrm{Cl}}$ for different values of p, e

p	e	# keys	# colls	χ^2	p-value
$2^{32} \cdot 5 - 1$	33	1629	8	1659.4	0.288
	34	3097	56	3572.2	0.000
	35	5960	198	7632.7	0.000
$2^{43} \cdot 3 - 1$	39	4012	3	3964.1	0.698
	40	7776	5	7678.3	0.780
	42	29581	116	31116.6	0.000
$2^{45} \cdot 7 - 1$	42	5564	1	5562.6	0.499
	43	10665	9	10876.9	0.073
	44	20862	33	21198.8	0.050
$2^{64} \cdot 3 - 1$	54	67734	2	67821.4	0.404
	55	133411	1	133240.5	0.628

5.1 The CORAL group action

Given the uncertainty around the quantum security of CSIDH and related protocols with different parameter sets [Pei20,BLMP19,BS20], we consider different possible choices for the prime p bit size. We provide in Table 3 benchmarks for a wide range of parameters, from the most optimistic 512 bits to the more conservative 4096 bits. More precisely, we consider the same primes as in [DEIV25]. It is outside of the scope of this work to write assembly-optimized implementations for arithmetic for each of these primes. However, to prove the compatibility of this potential improvement and as a reference point, we also provide benchmarks for the 505-bit prime used as level-5 parameter of the round 2 NIST submission of SQIsign [AAA+24], using their optimized arithmetic.

Table 3: Performance of CORAL subroutines

$\log p$	e	Ideal Sampling (ms)	Action Evaluation (ms)
505†	392	8.57	4.43
509	394	8.01	6.99
1008	642	45.7	36.5
2032	1155	355	240
4092	2185	3701	1780

Language/Library: C/GMP 6.3.0. Compiler: GCC 15.2.1. Hardware: AMD Ryzen 7 PRO 7840U @3.3GHz, boost disabled. Average over 1000 samples, data rounded to three significant figures. †Using optimized arithmetic for the level 5 parameters of the round 2 NIST submission of SQIsign [AAA+24].

Table 4: Comparison of selected isogeny-based group-action evaluations

			Time (ms)			
			$\approx \log(\mathrm{Disc}(\mathcal{O}))$			
	Name	Lang.	512	1024	2048	4096
Unrestricted	KLaPoTi [PPS25]	Rust	2 360	17 500	138 000	
	PEGASIS [DEF+25]	Sage	1 900	5 290	21 300	121 000
	qt-PEGASIS [DEIV25]	Sage	1 200	3 250	11 800	58 000
	qt-PEGASIS [DD26]	C	73.6	419	2 940	22 000
Restricted	CSIDH [CLM+18]	C/`asm`	26.5	529		
	SQALE [CSCJR22]	C/`asm`	66.7	240	1 300	7 710
	dCTIDH [CHMR25]	C/`asm`			400	
	CSIDH-LDO [ZLL+26]	C/AVX			408	696
	OSIDH-LD [WLLZ26]	C/AVX				754
	CORAL	C	6.99	36.5	240	1 780
	CORAL	C/`asm`	4.43			

Benchmarks exclusively measure the isogeny computation; excluding norm equation solving, public key compression, or public key validation where applicable. The dCTIDH variant used is $m = 6, \ell = 194$. The SQALE variant used is "MCR". `asm` denotes use of assembly-optimized arithmetic. AVX denotes use of platform-specific *Advanced Vector Extensions*. Data for CORAL and CSIDH collected over 1000 samples. Data for other algorithms collected over 100 samples. All benchmarks measured with AMD Ryzen 7 PRO 7840U@3.3GHz, boost and hyperthreading disabled. Times rounded to three significant figures.

5.2 An efficient NIKE from CORAL

CORAL serves as a drop-in replacement for the well-known CSIDH non-interactive key exchange [CLM+18]: thus, the improved efficiency of CORAL leads to a significant improvement for compact, actively secure post-quantum NIKEs, an area in which the community has been looking for quantum-resistant alternatives to classical Diffie-Hellman. We refer to [DHK+22, DHK+23] for details on the precise design and security properties of the group-action-based NIKEs in the QROM [BDF+11]. To estimate the concrete performance of a CORAL-based NIKE, we recall that:

- The public key consists of a supersingular elliptic curve defined over $\mathbb{F}_p$, which can be represented by the j-invariant of the curve, thus taking exactly $\log(p)$ bits (e.g. 256 bytes for a 2048-bit base prime p).
- Key generation requires sampling one ideal and one group action evaluation.

– The secret derivation procedure requires only one group action evaluation, since the ideal has already been sampled during key generation.

We summarise the resulting performance in Table 5, and compare to other post-quantum NIKEs in Table 6.

Table 5: CORAL NIKE performance

$\log p$	e	$\lvert\mathsf{pk}\rvert$ (bytes)	KeyGen (ms)	SharedKey (ms)
505†	392	64	13.0	4.43
509	394	64	15.0	6.99
1008	642	128	81.1	36.5
2032	1155	256	594	240
4092	2185	512	5 490	1 780

Written in unoptimised C. Use of GMP 6.3.0 in KeyGen. Compiler: GCC 15.2.1. Hardware: AMD Ryzen 7 PRO 7840U @3.3GHz, boost and multithreading disabled. Average over 1000 samples, data rounded to three significant figures. †Using optimized arithmetic for the level 5 parameters of the round 2 NIST submission of SQIsign [AAA+24].

Remark 5.1. As shown by the benchmarks in Table 3, the norm equation solving step (`SampleIdeal`) is the bottleneck of CORAL's key generation, and timings are on the order of seconds for larger parameter choices. However, we remark that the widely-deployed RSA cryptosystem suffers from a comparably costly key generation: concretely, we observed key generation times of 60/616/6560 ms for RSA keys of size 2048/4096/8192 bits; this data was averaged over 25 runs, rounded to three significant figures, and generated with OpenSSH's `ssh-keygen` version 3.6.1 on the same hardware as Table 3.

5.3 Future optimisations

While our implementation is sufficient to evaluate the practical performance of CORAL and to support a fair comparison with prior work, it was not intended to represent a fully optimized software realization. Several improvements and optimizations are possible.

Assembly-optimized arithmetic. The use of assembly-optimized finite field arithmetic can significantly improve performance of isogeny-based schemes [AAA+24, JAC+22]. The concrete speedup for CORAL using the broadwell optimized arithmetic developed for SQIsign's modulus $p = 5 \cdot 2^{248} - 1$ was roughly 1.6× (Table 3).

Table 6: Comparison of selected post-quantum NIKEs

Scheme	Lang.	Actively secure	|pk| (B)	KeyGen (ms)	SharedKey (ms)
dCTIDH2047 [CHMR25]	C/`asm`	Yes	256	3840	479†
CSIDH-LDO2000 [ZLL+26]	C/AVX	No*	320§	439	427
CSIDH-LDO4000 [ZLL+26]	C/AVX	No*	576§	784	761
OSIDH-LD4000 [WLLZ26]	C/AVX	No*	576§	844	754
Swoosh [GdKQ+24]	Rust Jasmin `asm`	No‡	221184	38.2	2.53
CORAL2032-1155	C	Yes	256	594	240
CORAL4092-2185	C	Yes	512	5480	1 780

Using variants `AVX-512IFMA-8w_r7` and `AVX-512IFMA-8w8w_r13` for CSIDH-LDO2000 and CSIDH-LDO4000 respectively. CSIDH-LDO variants include compression (KeyGen) and decompression (SharedKey). †With improved public key validation as described in [PRR+25, Tab. 4]. *Does not include public-key validation necessary to make the protocol actively secure. At time of writing, practically efficient public-key validation is still in progress [Hou25, Sec. 7]. §Using best possible theoretical approximation $2\log(p) + \log(\mathrm{Disc}(\mathcal{O}))$ bits for public-key sizes, not implemented sizes (here $r = 7$ (resp. 13) for 2048-bit (resp. 4096-bit) discriminant). ‡Whilst the full Swoosh protocol can be made actively secure though applying proofs-of-knowledge, the current (benchmarked) implementation does not include these; nor does the reported public-key size include the additional 89 KB space required for these proofs. `asm` denotes an implementation with assembly-optimized arithmetic. Data collected from 500 key exchanges (i.e. 1000 KeyGen and SharedKey calls). All benchmarks measured with AMD Ryzen 7 PRO 7840U@3.3GHz, boost and hyperthreading disabled. Times rounded to three significant figures.

We note that the final performance increase heavily depends on the concrete implementation and compiler used; however, as [JAC+22, Tab. 2.1] has shown, these gains are likely to stay constant as the modulus grows (or even marginally increase).

Hardware acceleration. Recent work [AALP26] has shown that one can accelerate computing the two-dimensional isogeny formula by parallelising the doubling and evaluation of kernel points in the 2-isogeny chain using AVX512 extensions. While their concrete implementation is specific to SQIsign Lvl3 parameters, the technique itself is field-agnostic. The reported overall speedup lies at a factor of 2.9× in their SQIsign Lvl3 implementation. We note that also the ARM architectures oriented optimization to the two-dimensional isogeny formula from [DJWY26] could be adapted to CORAL.

Better isogeny formulas. As detailed in Section B, there are a few avenues of optimisation in the gluing and diagonal isogenies steps. However, in practice, these steps together account for less than 5% of the total runtime. Four additional square roots (in total, two per E_1, E_2) are necessary to recover the Montgomery model after computing the diagonal isogenies; these square roots become more expensive as the prime grows, and they account for around 2% of the total runtime at 2000 bits. Better formulas could thus save a few computations. Moreover, adapting the gluing formulas from [Dup25, DD26] may lead to an implementation that only relies on $\mathbb{F}_p$ arithmetic, without requiring $\mathbb{F}_{p^2}$ operations for the gluing step.

5.4 Towards a constant-time implementation

The implementation of CSIDH is notoriously hard to achieve in constant time, and several techniques have been proposed to protect against timing attacks [JAKJ19, MCR19, OAYT19, CCC+19, LH20, BKL+23, LeG23, CHMR25]. CORAL has almost constant-time evaluation by design, while the ideal sampling routine may be harder to implement in constant-time, similarly to (qt-)PEGASIS.[13]

Ideal sampling. Sampling a CORAL ideal requires solving a norm equation. Compared to most norm-equation algorithms in isogeny-based cryptography, ours should be relatively simple to make constant-time, but it is still a non-trivial task. Important simplifications are that the integer sizes are bounded by 2^{2e}, and rejecting an integer does not leak any information since integers are assumed to be sampled independently. However, the trial division and `SumOfTwoSquares` steps of the algorithm are not constant-time, as they depend on the number of prime factors of the sampled integer $2^{2e-2} - y^2 p$ and on the size of these factors. A straightforward solution would be to only allow for prime sampled integers, i.e. set $\mathcal{P}_{\text{good}} = \emptyset$ and directly use `PrimeSumOfTwoSquares`. The latter procedure can be implemented in constant time, as shown in [BHJ+25, JMKR23]. We leave it to future work to determine the efficiency of such an approach or whether it is possible to develop a more efficient solution.

Action evaluation. Evaluating the CORAL action comprises elliptic curve point multiplication, one- and two-dimensional isogenies. Constant-time point-multiplication is a cornerstone of elliptic-curve cryptography; while both one- and two-dimensional isogenies are amenable to constant-time implementation, as demonstrated, for example, by the SIKE [JAC+22] and SQIsign [AAA+24] NIST submissions. Indeed, our current implementation is constant-time and deterministic[14] for each of these subroutines.

[13] In (qt-)PEGASIS, evaluation consists of two steps: solving a norm equation and evaluating a four-dimensional isogeny. The norm equation solving is a one-time cost and can be moved to the sampling phase, leaving the evaluation constant-time as long as evaluating 4D isogenies is constant-time.

[14] Understood to mean that no high-quality source of randomness is required.

However, since the gluing isogeny occurs after δ (Equation (5)) diagonal isogeny steps, δ is leaked during computation. Recall that $\delta + 1$ equals the maximum 2-adic valuation of $x - y \pm (2^{e-1} - a)$ over the two sign choices, where (y, x, a) is the secret (i.e. a solution to Eq. (4)). Crucially, δ is fixed per secret, so that evaluating the group action for the same secret with different curves (public keys) does not reveal additional information.

This suggests a simple path to constant-time group action evaluation: modify the ideal sampling algorithm to accept only norm-equation solutions with exactly one diagonal isogeny (recall that all secret keys require at least one diagonal isogeny). Heuristically (supported by over 250,000 experiments at all parameters), this rejects half of all solutions, making the ideal sampling step roughly twice as expensive (thus slowing down CORAL's KeyGen by a factor about 1.5 for 2000 bit parameters), and reducing the key-space by 1 bit. In this case, the group action evaluation becomes fully constant-time at no additional cost; and in particular makes performance comparisons to other fully constant-time implementations meaningful (e.g. Table 4).

Furthermore, even without this modification, knowledge of δ does not appear to immediately help an attacker: conditioned on $\delta = n$, there remain 2^n possible diagonal isogenies and so does not reduce the search space. This suggests that the rigorous $\delta > 1$ rejection technique outlined above could potentially be relaxed, even if this would require a more refined security analysis and is left for future work.

Finally, we remark that our basis-sampling algorithm could be made deterministic (at the expense of the public key size) by attaching a 2^{e+2}-torsion basis to the public key (for example using an Elligator seed, like dCTIDH [CHMR25]). Note that basis-sampling is done on public data, and so does not need to be constant-time.

6 Conclusion

In this work, we introduced CORAL, a restricted group action based on the ideal class group of $\mathbb{Z}[\pi]$. CORAL leverages two-dimensional isogenies to achieve significant performance improvements over CSIDH and (qt-)PEGASIS for group-action evaluation. As a direct application, we obtained an actively secure post-quantum NIKE with very small keys and practical performance.

Several directions for future work remain open. On the implementation side, further optimizations to both the ideal sampling and evaluation subroutines are possible, and we hope our implementation will serve as a starting point. On the construction side, CORAL can substitute random group action evaluations in schemes such as the signature scheme SeaSign [DG19] and oblivious transfer protocols [LGD21]. More ambitiously, combining CORAL for commitment generation with Qlapoti [BCE+25] for the remaining isogeny evaluations would yield a variant of the linkable ring signature Calamari [BKP20] computed entirely via two-dimensional isogenies.

Acknowledgements. All authors were supported by SNSF Consolidator Grant CryptonIs 213766.

References

AAA+24. Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

AALP26. Marius A. Aardal, Diego F. Aranha, Patrick Longa, and Giacomo Pope. Vectorized (2,2)-isogenies in the theta model using avx512 and applications to sqisign, 2026. Unpublished article.

ABE+25. Bill Allombert, Jean-François Biasse, Jonathan Komada Eriksen, Péter Kutas, Chris Leonardi, Aurel Page, Renate Scheidler, and Márton Tot Bagi. Faster SCALLOP from non-prime conductor suborders in medium sized quadratic fields. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part III*, volume 15676 of *LNCS*, pages 333–363. Springer, Cham, May 2025. doi:10.1007/978-3-031-91826-1_11.

ACC+19. Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In Carlos Cid and Michael J.: Jacobson, Jr., editors, *SAC 2018*, volume 11349 of *LNCS*, pages 322–343. Springer, Cham, August 2019. doi:10.1007/978-3-030-10970-7_15.

ADMP20. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Cham, December 2020. doi:10.1007/978-3-030-64834-3_14.

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 TCHES*, 2021(4):351–387, 2021. URL: https://tches.iacr.org/index.php/TCHES/article/view/9069, doi:10.46586/tches.v2021.i4.351-387.

BCE+25. Giacomo Borin, Maria Corte-Real Santos, Jonathan Komada Eriksen, Riccardo Invernizzi, Marzio Mula, Sina Schaeffler, and Frederik Vercauteren. Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 174–205. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5113-2_6.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In

Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_3.

BG06. Valentin Blomer and Andrew Granville. Estimates for representation numbers of quadratic forms, 2006. URL: https://dms.umontreal.ca/~andrew/PDF/quadraticforms.pdf.

BHJ+25. Andrea Basso, Chenfeng He, David Jacquemin, Fatna Kouider, Péter Kutas, Anisha Mukherjee, Sina Schaeffler, and Sujoy Sinha Roy. Constant-time quaternion algorithms for SQIsign. Cryptology ePrint Archive, Report 2025/2192, 2025. URL: https://eprint.iacr.org/2025/2192.

BKL+23. Gustavo Banegas, Juliane Krämer, Tanja Lange, Michael Meyer, Lorenz Panny, Krijn Reijnders, Jana Sotáková, and Monika Trimoska. Disorientation faults in CSIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 310–342. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_11.

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Cham, December 2020. doi:10.1007/978-3-030-64834-3_16.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Cham, December 2019. doi:10.1007/978-3-030-34578-5_9.

BKW20. Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 520–550. Springer, Cham, December 2020. doi:10.1007/978-3-030-64834-3_18.

BLMP19. Daniel J. Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. Quantum circuits for the CSIDH: Optimizing quantum evaluation of isogenies. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 409–441. Springer, Cham, May 2019. doi:10.1007/978-3-030-17656-3_15.

BS20. Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 493–522. Springer, Cham, May 2020. doi:10.1007/978-3-030-45724-2_17.

BY91. Gilles Brassard and Moti Yung. One-way group actions. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 94–107. Springer, Berlin, Heidelberg, August 1991. doi:10.1007/3-540-38424-3_7.

Cas21. Wouter Castryk. CSIDH on the surface (CSURF): Isogeny-based cryptography school, 2021. URL: https://homes.esat.kuleuven.be/~wcastryc/summer_school_csurf.pdf.

CCC+19. Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. Stronger and faster side-channel protections for CSIDH. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Secu-*

rity in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings, Lecture Notes in Computer Science, pages 173–193. Springer, 2019. doi:10.1007/978-3-030-30530-7_9.

CD20. Wouter Castryck and Thomas Decru. CSIDH on the surface. In Jintai Ding and Jean-Pierre Tillich, editors, *PQCrypto 2020*, pages 111–129. Springer, Cham, 2020. doi:10.1007/978-3-030-44223-1_7.

CD23. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_15.

CDS20. Wouter Castryck, Thomas Decru, and Benjamin Smith. Hash functions from superspecial genus-2 curves using richelot isogenies. *Journal of Mathematical Cryptology*, 14(1):268–292, 2020.

CHMR25. Fabio Campos, Andreas Hellenbrand, Michael Meyer, and Krijn Reijnders. dCTIDH: Fast & deterministic CTIDH. *IACR TCHES*, 2025(3):516–541, 2025. doi:10.46586/tches.v2025.i3.516-541.

CK20. Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. Cryptology ePrint Archive, Report 2020/985, 2020. URL: https://eprint.iacr.org/2020/985.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Cham, December 2018. doi:10.1007/978-3-030-03332-3_15.

CLP24. Mingjie Chen, Antonin Leroux, and Lorenz Panny. SCALLOP-HD: Group action from 2-dimensional isogenies. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14603 of *LNCS*, pages 190–216. Springer, Cham, April 2024. doi:10.1007/978-3-031-57725-3_7.

Coh93. Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag Berlin Heidelberg, 1993. URL: https://link.springer.com/book/10.1007/978-3-662-02945-9, doi:10.1007/978-3-662-02945-9.

Cor08. Giuseppe Cornacchia. Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. *Giornale di Matematiche di Battaglini*, 46:33–90, 1908.

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. URL: https://eprint.iacr.org/2006/291.

Cox13. David A. Cox. *Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication*, volume 116 of *Pure and Applied Mathematics*. Wiley, 2nd edition, 2013.

CS18. Craig Costello and Benjamin Smith. Montgomery curves and their arithmetic - the case of large characteristic fields. *Journal of Cryptographic Engineering*, 8(3):227–240, September 2018. doi:10.1007/s13389-017-0157-6.

CSCJR22. Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, 12(3):349–368, September 2022. doi:10.1007/s13389-021-00271-w.

CSD+23. Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

DD26. Pierrick Dartois and Max Duparc. Chasing rabbits through hypercubes: better algorithms for higher dimensional 2-isogeny computations. Cryptology ePrint Archive, Paper 2026/114, 2026. URL: https://eprint.iacr.org/2026/114.

De 17. Luca De Feo. Mathematics of isogeny based cryptography. arXiv preprint, Paper 1711.04062, 2017. See updated version https://github.com/defeo/MathematicsOfIBC. URL: https://arxiv.org/abs/1711.04062.

DEF+25. Pierrick Dartois, Jonathan Komada Eriksen, Tako Boris Fouotsa, Arthur Herlédan Le Merdy, Riccardo Invernizzi, Damien Robert, Ryan Rueger, Frederik Vercauteren, and Benjamin Wesolowski. PEGASIS: Practical effective class group action using 4-dimensional isogenies. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part I*, volume 16000 of *LNCS*, pages 67–99. Springer, Cham, August 2025. doi:10.1007/978-3-032-01855-7_3.

DEIV25. Pierrick Dartois, Jonathan Komada Eriksen, Riccardo Invernizzi, and Frederik Vercauteren. qt-pegasis: Simpler and faster effective class group actions. Cryptology ePrint Archive, Report 2025/1859, 2025. URL: https://eprint.iacr.org/2025/1859.

DFK+23. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: Scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 345–375. Springer, Cham, May 2023. doi:10.1007/978-3-031-31368-4_13.

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

DG19. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Cham, May 2019. doi:10.1007/978-3-030-17659-4_26.

DHK+22. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Group action key encapsulation and non-interactive key exchange in the QROM. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 36–66. Springer, Cham, December 2022. doi:10.1007/978-3-031-22966-4_2.

DHK+23. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Cham, May 2023. doi:10.1007/978-3-031-31368-4_15.

DJP11. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Cryptology

ePrint Archive, Report 2011/506, 2011. URL: https://eprint.iacr.org/2011/506.

DJWY26. Luca De Feo, Li-Jie Jian, Ting-Yuan Wang, and Bo-Yin Yang. SQISign on ARM. Cryptology ePrint Archive, Paper 2026/394, 2026. URL: https://eprint.iacr.org/2026/394.

DKL+20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 64–93. Springer, Cham, December 2020. doi:10.1007/978-3-030-64837-4_3.

DM20. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Cham, May 2020. doi:10.1007/978-3-030-45388-6_7.

Dup25. Max Duparc. Superglue: Fast formulae for (2,2)-gluing isogenies. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 372–400. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5113-2_12.

FHL+26. Tako Boris Fouotsa, Marc Houben, Gioella Lorenzon, Ryan Rueger, and Parsa Tasbihgou. On the active security of the pearl-scallop group action. In Magali Bardet and Ruben Niederhagen, editors, *Post-Quantum Cryptography*, pages 74–104, Cham, 2026. Springer Nature Switzerland.

FM02. Mireille Fouquet and François Morain. Isogeny volcanoes and the sea algorithm. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory*, pages 276–291, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

Gal18. Steven D. Galbraith. Mathematics of public key cryptography. version 2.0, 2018. URL: https://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html.

GdKQ+24. Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. SWOOSH: Efficient lattice-based non-interactive key exchange. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024*. USENIX Association, August 2024. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/gajland.

HHM+24. Lena Heimberger, Tobias Hennerbichler, Fredrik Meisingseth, Sebastian Ramacher, and Christian Rechberger. OPRFs from isogenies: Designs and analysis. In Jianying Zhou, Tony Q. S. Quek, Debin Gao, and Alvaro A. Cárdenas, editors, *ASIACCS 24*. ACM Press, July 2024. doi:10.1145/3634737.3645010.

HLM+25. Lucjan Hanzlik, Yi-Fu Lai, Marzio Mula, Eugenio Paracucchi, Daniel Slamanig, and Gang Tang. Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum group actions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 35–69. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5113-2_2.

HM89. James L Hafner and Kevin S McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American mathematical society*, 2(4):837–850, 1989.

Hou25. Marc Houben. Efficient post-quantum commutative group actions from orientations of large discriminant. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 141–173.

Springer, Singapore, December 2025. doi:10.1007/978-981-95-5113-2_5.

JAC+22. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.

JAKJ19. Amir Jalali, Reza Azarderakhsh, Mehran Mozaffari Kermani, and David Jao. Towards optimized and constant-time CSIDH on embedded devices. In Ilia Polian and Marc Stöttinger, editors, *COSADE 2019*, volume 11421 of *LNCS*, pages 215–231. Springer, Cham, April 2019. doi:10.1007/978-3-030-16350-1_12.

JMKR23. David Jacquemin, Anisha Mukherjee, Péter Kutas, and Sujoy SINHA ROY. Ready to SQI? Safety first! Towards a constant-time implementation of isogeny-based signature, SQIsign. Cryptology ePrint Archive, Report 2023/807, 2023. URL: https://eprint.iacr.org/2023/807.

Kan97. Ernst Kani. The number of curves of genus two with elliptic differentials. *Journal für die reine und angewandte Mathematik*, 485:93–122, 1997. URL: http://eudml.org/doc/183547.

KLLQ23. Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 729–761. Springer, Cham, August 2023. doi:10.1007/978-3-031-38548-3_24.

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

Kup05. Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. doi:10.1137/S0097539703436345.

Kup13. Greg Kuperberg. Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. In Simone Severini and Fernando Brandao, editors, *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20–34, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2013.20, doi:10.4230/LIPIcs.TQC.2013.20.

Lai24. Yi-Fu Lai. Capybara and tsubaki: Verifiable random functions from group actions and isogenies. *CiC*, 1(3):1, 2024. doi:10.62056/avr-11zn4.

Lan87. Serge Lang. *Elliptic Functions*, volume 112 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, NY, 2nd edition, 1987. doi:10.1007/978-1-4612-4752-4.

LB20. Jonathan Love and Dan Boneh. Supersingular curves with small non-integer endomorphisms, 2020. URL: https://arxiv.org/abs/1910.03180, arXiv:1910.03180.

LeG23. Jason T. LeGrow. A faster method for fault attack resistance in static/ephemeral CSIDH. *Journal of Cryptographic Engineering*, 13(3):283–294, September 2023. doi:10.1007/s13389-023-00318-0.

LGD21. Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and UC-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 213–241. Springer, Cham, October 2021. doi:10.1007/978-3-030-77870-5_8.

LH20. Jason LeGrow and Aaron Hutchinson. An analysis of fault attacks on CSIDH. Cryptology ePrint Archive, Report 2020/1006, 2020. URL: https://eprint.iacr.org/2020/1006.

Lit28. J. E. Littlewood. On the class-number of the corpus $p(\sqrt{-k})$. *Proceedings of the London Mathematical Society*, s2-27(1):358–372, 1928. URL: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s2-27.1.358, doi:10.1112/plms/s2-27.1.358.

MCR19. Michael Meyer, Fabio Campos, and Steffen Reith. On lions and elligators: An efficient constant-time implementation of CSIDH. In Jintai Ding and Rainer Steinwandt, editors, *PQCrypto 2019*, pages 307–325. Springer, Cham, 2019. doi:10.1007/978-3-030-25510-7_17.

MMP+23. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_16.

MN90. François Morain and Jean-Louis Nicolas. On Cornacchia's algorithm for solving the diophantine equation $u^2 + dv^2 = m$, 1990. URL: http://www.lix.polytechnique.fr/Labo/Francois.Morain/Articles/cornac.pdf.

NO24. Kohei Nakagawa and Hiroshi Onuki. QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part V*, volume 14924 of *LNCS*, pages 75–106. Springer, Cham, August 2024. doi:10.1007/978-3-031-68388-6_4.

OAYT19. Hiroshi Onuki, Yusuke Aikawa, Tsutomu Yamazaki, and Tsuyoshi Takagi. (Short paper) A faster constant-time algorithm of CSIDH keeping two points. In Nuttapong Attrapadung and Takeshi Yagi, editors, *IWSEC 19*, volume 11689 of *LNCS*, pages 23–33. Springer, Cham, August 2019. doi:10.1007/978-3-030-26834-3_2.

Onu20. Hiroshi Onuki. On oriented supersingular elliptic curves, 2020. URL: https://arxiv.org/abs/2002.09894, arXiv:2002.09894.

Oud25. Remy Oudompheng. pyroclastic. https://github.com/remyoudompheng/pyroclastic, 2025.

OZ24. Emmanuela Orsini and Riccardo Zanotto. Simple two-message OT in the explicit isogeny model. *CiC*, 1(1):15, 2024. doi:10.62056/a39qgy4e-.

Pan23. Lorenz Panny. Csi-fish really isn't polynomial-time, 2023. URL: https://yx7.cc/blah/2023-04-14.html.

Pei20. Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 463–492. Springer, Cham, May 2020. doi:10.1007/978-3-030-45724-2_16.

PPS25. Lorenz Panny, Christophe Petit, and Miha Stopar. KLaPoTi: An asymptotically efficient isogeny group action from 2-dimensional isogenies. *CiC*, 2(3):5, 2025. doi:10.62056/ahp2wakrz.

PR23. Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766, 2023. URL: https://eprint.iacr.org/2023/1766.

PRR+25. Giacomo Pope, Krijn Reijnders, Damien Robert, Alessandro Sferlazza, and Benjamin Smith. Simpler and faster pairings from the montgomery ladder. *CiC*, 2(2):29, 2025. doi:10.62056/ah2i893y6.

Ren18. Joost Renes. Computing isogenies between Montgomery curves using the action of (0, 0). In Tanja Lange and Rainer Steinwandt, editors, *PQCrypto 2018*, pages 229–247. Springer, Cham, 2018. doi:10.1007/978-3-319-79063-3_11.

Rob23. Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_17.

Rob25. Damien Robert. On the efficient representation of isogenies. In Andrzej Dąbrowski, Josef Pieprzyk, and Jacek Pomykała, editors, *Number-Theoretic Methods in Cryptology*, pages 3–84, Cham, 2025. Springer Nature Switzerland.

RS06. Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. URL: https://eprint.iacr.org/2006/145.

RS24. Damien Robert and Nicolas Sarkis. Computing 2-isogenies between Kummer lines. *CiC*, 1(1):26, 2024. doi:10.62056/abvua69p1.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990. doi:10.1007/0-387-34805-0_22.

Sie35. Carl Siegel. Über die classenzahl quadratischer zahlkörper. *Acta Arithmetica*, 1(1):83–86, 1935. URL: http://eudml.org/doc/205054.

Sil94. Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer Science & Business Media, New York, 1994. doi:10.1007/978-1-4612-0851-8.

Sil09. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.

Sut13. Andrew V. Sutherland. Isogeny volcanoes. In *Proceedings of the Tenth Algorithmic Number Theory Symposium*, volume 1 of *Open Book Series*, pages 507–530. Mathematical Sciences Publishers, 2013. URL: https://doi.org, doi:10.2140/obs.2013.1.507.

Vél71. Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, July 1971. URL: https://gallica.bnf.fr/ark:/12148/bpt6k56191248/f52.item.

vOW99. Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, January 1999. doi:10.1007/PL00003816.

Wat69. William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 2(4):521–560, 1969. URL: https://www.numdam.org/articles/10.24033/asens.1183/, doi:10.24033/asens.1183.

WLLZ26. Weize Wang, Yi-Fu Lai, Kaizhan Lin, and Yunlei Zhao. Efficient and parallel implementation of isogeny-based deterministic group actions. Cryptology ePrint Archive, Paper 2026/627, 2026. URL: https://eprint.iacr.org/2026/627.

Zar74. Ju G Zarhin. A remark on endomorphisms of abelian varieties over function fields of finite characteristic. *Mathematics of the USSR-Izvestiya*, 8(3):477–480, 1974.

ZLL+26. Yuhao Zheng, Jianming Lin, Yutong Liang, Yanzhen Ren, Huixin Zhang, and Chang-An Zhao. Compressed key exchange protocol from orientations of large discriminant using AVX-512. Cryptology ePrint Archive, Paper 2026/679, 2026. URL: https://eprint.iacr.org/2026/679.

A Sampling in $\mathcal{P}_e$: decomposing into sums of two squares

Ideal sampling in CORAL is described in `SampleIdeal`. It consists of sampling uniformly random odd integers y until $L = 2^{2e-2} - y^2 p$ is decomposable as a sum of two squares (and then computing such a decomposition). The performance of `SampleIdeal` therefore crucially depends on an algorithm to efficiently decompose random odd integers into a sum of two squares.

We recall that an integer L is the sum of two squares $x^2 + a^2$ if and only if primes $q \equiv 3 \pmod 4$ dividing L, divide with even multiplicity. Primes that are 1 modulo 4 can be efficiently decomposed into the sum of two squares with `PrimeSumOfTwoSquares` (Algorithm 2). However, decomposing *composite* integers generally requires knowing the prime factorisation of L or knowing a square root of -1 modulo L.

Choices and optimizations. In order to efficiently solve our norm equation, we need to consider a tradeoff. More successful (or less partial) factorizations are more costly, but also allow to decompose a larger proportion of odd integers L into sums of two squares, thus allowing a lower iteration number for `SampleIdeal`. The extreme case of full factorization is not efficient for our parameter choices. The opposite approach (only attempting to only decompose primes 1 mod 4) is chosen by [AAA+24], but also not optimal for our larger problem inputs. We therefore use trial division by small primes to obtain a (partial) factorization of inputs.

In the following, we describe a technique to reduce the cost of primality tests on the remainder of the trial divisions, before detailing our trial division approach and finally combining them to obtain our `SumOfTwoSquares` algorithm in Algorithm 5.

Primality testing. Once a suitable number is found and partially factored by trial division, the remaining quotient can be tested for primality, if it is 1 (mod 4), and rejected otherwise. However, as remarked by [BHJ+25], the primality test is unnecessary, since directly attempting to compute a square root of -1 is not more costly than a single Miller-Rabin iteration, and allows one to immediately know whether `PrimeSumOfTwoSquares` has a chance to succeed. Our version of this, given in Algorithm 3, slightly differs from [BHJ+25], because we do not aim for a constant-time execution, but for a low failure probability and efficiency.

Trial division. Similarly to the first NIST submission of SQIsign [CSD+23], we use trial division to decompose at least some non-prime odd positive integers L as sums of two squares by applying `PrimeSumOfTwoSquares` to each factor with odd exponent, and combining the results as explained in `SumOfTwoSquares`. Unlike `PrimeSumOfTwoSquares`, we only compute Euclidean divisions, without any GCDs. We did not compare the efficiency of the two approaches. An additional difference to [CSD+23] is that we reject a number only if it has a prime factor 3 (mod 4) with odd exponent. This increases the probability of finding solvable instances, which is important since our instances are larger than those in SQIsign and primes therefore more rare.

We give pseudocode for the trial division algorithm in Algorithm 4 and for our final sum-of-two-squares solution in Algorithm 5.

Algorithm 2 `PrimeSumOfTwoSquares` (following [MN90])

Input: q prime such that $q \equiv 1 \pmod 4$, u such that $u^2 \equiv -1 \pmod q$
Output: x, y such that $x^2 + y^2 = q$

Compute r_0, q_0 such that $u = qq_0 + r_0$ ▷ *By Euclidean division*
Compute r_1, q_1 such that $q = r_0 q_1 + r_1$ ▷ *By Euclidean division*
while not $r_1^2 < q \leq r_0^2$ **do**
  Compute r_2, q_2 such that $r_0 = r_1 q_2 + r_2$ ▷ *By Euclidean division*
  $r_1, r_0 \leftarrow r_2, r_1$
$x \leftarrow r_1$
$y \leftarrow \sqrt{q - r_1^2}$
if $x^2 + y^2 = q$ **and** y integer **then**
  return x, y
else return $\perp$

Algorithm 3 `sqrtMinusOne`

Input: A positive odd integer n, a list $\mathcal{P}_{\text{good}}$ of μ primes which are $1 \pmod 4$
Output: u such that $u^2 \equiv -1 \pmod n$ otherwise $\perp$

if $n \pmod 3 = 2$ **then**
  $g \leftarrow -3$
else
  found $\leftarrow$ False
  while found = False **do**
    $g \leftarrow$ next element of $\mathcal{P}_{\text{good}}$
    if n is not a square modulo g **then**
      found $\leftarrow$ True
  if found = False **then**
    return $\perp$ ▷ *If n is prime this happen with probability $\leq 1/2^{\mu}$*
$u \leftarrow g^{n-1/4} \pmod n$
if $u^2 \equiv -1 \pmod n$ **then return** u
else return $\perp$

Algorithm 4 `TrialDivision`

Input: A non-negative odd integer n, a set $\mathcal{P}_{\text{good}}$ of primes which are $1 \pmod 4$, a set of primes $\mathcal{P}_{\text{bad}}$ which are $3 \pmod 4$
Output: $\perp$ or N and $\mathcal{F}$ factorization such that N times all elements in the factorization is n
Output: N is coprime to all elements in $\mathcal{P}_{\text{good}}$ and $\mathcal{P}_{\text{bad}}$
Output: the exponent of any prime in $\mathcal{P}_{\text{bad}}$ is even in factorization

$\mathcal{F} \leftarrow []$ ▷ *Used to store factorisation of n by the given primes*
for $\ell \in \mathcal{P}_{\text{bad}}$ **do** ▷ *Iterating over ℓ's in ascending order*
  $m \leftarrow 0$
  while $\ell \mid n$ **do**
    $n \leftarrow n/\ell$; $m \leftarrow m + 1$
  if $m \equiv 1 \pmod 2$ **then**
    return $\perp$
  $\mathcal{F} \leftarrow \mathcal{F} \cup (\ell, m)$
for $\ell \in \mathcal{P}_{\text{good}}$ **do**
  $m \leftarrow 0$
  while $\ell \mid n$ **do**
    $n \leftarrow n/\ell$
    $m \leftarrow m + 1$
  $\mathcal{F} \leftarrow \mathcal{F} \cup (\ell, m)$
return $(n, \mathcal{F})$

Algorithm 5 `SumOfTwoSquares`

Input: A non-negative odd integer n, a set $\mathcal{P}_{\text{good}}$ of primes which are $1 \pmod 4$, a set of primes $\mathcal{P}_{\text{bad}}$ which are $3 \pmod 4$
Output: x, y such that $x^2 + y^2 = n$ or $\perp$ if n cannot be decomposed in sum of two squares using $\mathcal{P}_{\text{bad}}$ and $\mathcal{P}_{\text{good}}$

$\mathcal{F}, N \leftarrow$`TrialDivision` $(n, \mathcal{P}_{\text{good}}, \mathcal{P}_{\text{bad}})$
if $\mathcal{F} = \perp$ **then return** $\perp$
$S \leftarrow 1$, $C \leftarrow 1$
$i \leftarrow$ a square root of -1 ▷ *i.e. $\mathbb{Z}[i]$ are the Gaussian integers*
$u \leftarrow$ `sqrtMinusOne`(N)
if $u = \perp$ **then return** $\perp$
$x, y \leftarrow$ `PrimeSumOfTwoSquares`(N, u)
if $(x, y) = \perp$ **then return** $\perp$
$C \leftarrow x + iy$
for q, e in $\mathcal{F}$ **do**
  $S \leftarrow Sq^{\lfloor e/2 \rfloor}$
  if $e \equiv 1 \pmod 2$ **then**
    $x, y \leftarrow$ `PrimeSumOfTwoSquares`(q)
    $C \leftarrow C(x + iy)$
$C \leftarrow CS$
return $(\Re(C), \Im(C))$ ▷ *Real and imaginary part of C*

B Two-dimensional isogeny computation

We break down the isogeny computation into several stages: basis sampling, constructing the kernel, applying an endomorphism, diagonal isogenies, gluing into a two-dimensional surface, general two-dimensional steps, and finally a splitting isogeny (See Fig. 1).

Because the overall chain $\Phi\colon E^2 \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$ embeds $\mathbb{F}_p$-rational isogenies between $\mathbb{F}_p$-rational elliptic curves on the surface, we are able to perform (the majority of) the computations over $\mathbb{F}_p$, as in CSIDH and (qt-)PEGASIS.

More precisely, our C implementation only the gluing isogeny and splitting isogenies are computed over $\mathbb{F}_{p^2}$. The Sage implementation computes the splitting over $\mathbb{F}_p$ too. Notably, these are only two steps in an isogeny chain of length roughly $\log(p)/2+\lambda$, and so account for relatively little computation. This holds especially for the splitting, because no kernel points need to be mapped through it.

Basis sampling. Recall that we are acting on $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$, i.e. the set of supersingular elliptic curves defined over $\mathbb{F}_p$ (up to $\mathbb{F}_p$-isomorphism) with $\mathbb{F}_p$-rational 2-torsion. Curves in $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$ have the following group structure

$$E(\mathbb{F}_p)[2^f] \cong \mathbb{Z}/2^{f-1}\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z},$$

and they may always be put into Montgomery form $y^2 = x^3 + Ax^2 + x$ for some $A \in \mathbb{F}_p$ [CD20]. The *quadratic twist* $E^t : y^2 = x^3 - Ax^2 + x$ of E is isomorphic over $\mathbb{F}_{p^2} = \mathbb{F}_p(\mathrm{i})$ via the twisting map $(x, y) \mapsto (-x, \mathrm{i}y)$ and also represents an element of $\mathcal{E}(\mathbb{Z}[(\pi+1)/2])$. By sampling points P, Q of order $2^e \leq 2^{f-1}$ on $E(\mathbb{F}_p), E^t(\mathbb{F}_p)$ respectively, we obtain a basis $P, Q^t = \tau(Q)$ of $E(\mathbb{F}_{p^2})[2^e]$ whose x-coordinates are defined over $\mathbb{F}_p$. By performing most computations using exclusively the x-coordinates, a basis of this form would allow us to perform most computation over $\mathbb{F}_p$.

To sample P, Q, we rely on the algorithm suggested by [DEF+25, Lem. D.2]. This lemma states that a point (x, y) lying on $E(\mathbb{F}_p)$ has order divisible by 2^{f-1} if and only if $x - x(T_{-1})$ is a non-square in $\mathbb{F}_p$; likewise, (x, y) lying on $E^t(\mathbb{F}_p)$ has order divisible by 2^{f-1} if $x + x(T_1)$ is a non-square in $\mathbb{F}_p$. Here, T_1, T_{-1} are the unique 2-torsion points on $E(\mathbb{F}_p)$ that respectively do and do not have a ($\mathbb{F}_p$-rational) 4-torsion point above them. Notice that we must only compute T_1, T_{-1} for E, not also E^t.

As usual, we find points lying on E, E^t by guessing an x-coordinate of the form $x = n + x(T_{-1})$, $x = n - x(T_1)$, where n is from a pre-computed set of non-squares in $\mathbb{F}_p$, and checking whether $x^3 + Ax^2 + x$ is a square. To finally obtain a basis of the desired order 2^e, we scale the already sampled points by $c2^{f-1-e}$.

Remark B.1. To compute the gluing isogeny without additional square root computations, one needs the 4-torsion above the kernel. In practice, our implementation samples a basis of order 2^{e+2} to account for this. We have omitted this detail for clarity of exposition.

Non-gluing isogenies. Our analysis follows [BCE+25, Sec. 5.2], adapted to our case. Recall that a 2^n-isogeny $\varphi\colon A \to B$ between principally polarized abelian varieties can be decomposed into a sequence of n 2-isogenies $\varphi = \varphi_n \circ \cdots \circ \varphi_1$, and the kernel of $\varphi_i \circ \cdots \circ \varphi_1$ is generated by $2^{n-i}K$. Consequently, the kernel of φ_i is generated by $2^{n-i}(\varphi_{i-1} \circ \cdots \circ \varphi_1)(K)$ for $i > 1$ and by $2^{n-1}K$ for $i = 1$.

Using a basis P, Q^t of $E(\mathbb{F}_{p^2})[2^e]$ sampled as described in the previous passage, the kernel of the two-dimensional 2^e-isogeny $\Phi\colon E^2 \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$ is generated by the points

$$\begin{aligned}\mathcal{P}_e &= (P_{e,1}, P_{e,2}) = (qP, (x+y\pi)(P)) = (qP, (x+y)P), \quad \text{and}\\ \mathcal{Q}_e^t &= (Q_{e,1}^t, Q_{e,2}^t) = (qQ^t, (x+y\pi)(Q^t)) = (qQ^t, (x-y)Q^t)\end{aligned}$$

on E^2. Indeed $\pi(P) = P$ by construction and $\pi(Q^t) = ((-x)^p, (\mathrm{i}y)^p) = (x, -\mathrm{i}y) = -Q^t$ because $p \equiv 3 \pmod 4$. By construction, q, y are odd and x is even; and so $P_{e,1}, P_{e,2}, Q_{e,1}^t, Q_{e,2}^t$ individually have order 2^e, hence so do $\mathcal{P}_e, \mathcal{Q}_e^t$.

Initial Endomorphisms. The kernel of the first step Φ_1 in the 2-isogeny chain $\Phi = \Phi_e \circ \Phi_{e-1} \circ \cdots \circ \Phi_1$ is given by $2^{e-1}\mathcal{P}_e = 2^{e-1}(P, P), 2^{e-1}\mathcal{Q}_e = 2^{e-1}(Q^t, Q^t)$, precisely because q, y are odd and x even. Consequently Φ_1 is given by the matrix $\left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$, because these maps share the same kernel.

Diagonal Isogenies. By mapping $\mathcal{P}_e, \mathcal{Q}_e^t$ through $\Phi_1 = \left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$, we see that the kernel of the remaining steps is generated by

$$\begin{aligned}\mathcal{P}_{e-1} &= (P_{(e-1),1}, P_{(e-1),2}) = (s_1 P, s_2 P)\\ \mathcal{Q}_{e-1}^t &= (Q_{(e-1),1}^t, Q_{(e-1),2}^t) = (t_1 Q^t, t_2 Q^t)\end{aligned}$$

where

$$s_1 = q + x + y, \quad s_2 = x + y - q, \quad t_1 = q + x - y, \quad t_2 = x - y - q.$$

Let $v_2(n)$ be the largest positive integer such that $2^{v_2(n)}$ divides n. We have that $1 = \min(v_2(s_1), v_2(s_2)) \lneq \max(v_2(s_1), v_2(s_2))$ (likewise for the t_i). Indeed, $s_1 = a + b, s_2 = a - b$ with $a = x + y, b = q$ odd, and so 4 divides exactly one of s_1, s_2. Moreover, by further algebraic manipulation on the equality $2^{2(e-1)} - y^2 p = q(2^e - q)$, one sees that $\max(v_2(s_1), v_2(s_2)) = \max(v_2(t_1), v_2(t_2)) =: \mu$.

Let us assume in the following that

$$v_2(s_1) = v_2(t_2) = 1 \quad \text{and} \quad v_2(s_2) = v_2(t_1) = \mu$$

so that

$$\begin{aligned}\mathrm{ord}(P_{(e-1),1}) = \mathrm{ord}(Q_{(e-1),2}) &= 2^{e-1}\\ \text{and} \quad \mathrm{ord}(P_{(e-1),2}) = \mathrm{ord}(Q_{(e-1),1}) &= 2^{e-\mu};\end{aligned}$$

confirming that $\mathcal{P}_{e-1}, \mathcal{Q}_{e-1}^t$ generate a *maximally isotropic kernel*.

We know that the kernel of next $\mu - 1$ steps $\Phi_\mu \circ \cdots \circ \Phi_2$ is generated by

$$\begin{aligned} 2^{(e-1)-(\mu-1)}\mathcal{P}_{e-1} &= \left(2^{e-\mu}s_1P, 2^{e-\mu+v_2(s_2)}(s_2/2^{v_2(s_2)})P\right) \\ &= \left(2^{e-\mu}s_1P, 0\right) \\ 2^{(e-1)-(\mu-1)}\mathcal{Q}^t_{e-1} &= \left(2^{e-\mu+v_2(t_1)}(t_1/2^{v_2(t_1)})Q^t, 2^{e-\mu}t_2Q^t\right) \\ &= \left(0, 2^{e-\mu}t_2Q^t\right); \end{aligned}$$

point of order $2^{e-(e-\mu+1)} = 2^{\mu-1}$. That is to say, the next $\mu - 1$ steps are diagonal isogenies. In particular, because $1 \lneq \mu$, we always have at least one diagonal isogeny, unlike [BCE+25], who could work around this by discarding some solutions of the norm equation.

Writing $\varphi_P \colon E \to E_1$ for the isogeny with kernel $2^{e-\mu}s_1P$ and $\varphi_{Q^t} \colon E \to E_2$ for the isogeny with kernel $2^{e-\mu}t_2Q^t$, we obtain generators for the kernel of the remaining $e - \mu$ steps

$$\begin{aligned} \mathcal{P}_{e-\mu} &= (P_{e-\mu,1}, P_{e-\mu,2}) = (\varphi_P(s_1P), \varphi_{Q^t}(s_2P)) \\ \mathcal{Q}^t_{e-\mu} &= (Q^t_{e-\mu,1}, Q^t_{e-\mu,2}) = \left(\varphi_P(t_1Q^t), \varphi_{Q^t}(t_2Q^t)\right). \end{aligned}$$

We note that the points $P_{e-\mu,*}$, $Q^t_{e-\mu,*}$ now all have the same order $2^{e-\mu}$. As such, the next step cannot be a diagonal isogeny.

Final endomorphisms. Suppose the next step were a (non-diagonal) endomorphism $\Phi_{\mu+1} \colon E_1 \times E_2 \to E_1 \times E_2$. Then it must be a matrix of isomorphisms $\Phi_{\mu+1} \colon \left(\begin{smallmatrix} \sigma_{11} & \sigma_{21} \\ \sigma_{12} & \sigma_{22} \end{smallmatrix}\right)$ with $\sigma_{ij} \colon E_i \xrightarrow{\sim} E_j$. However, then E_1 and E_2 must be isomorphic, which can only happen if $\langle 2^{e-\mu}s_1P\rangle = \langle 2^{e-\mu}t_2Q^t\rangle$, or E_1, E_2 have an $\mathbb{F}_p$-rational degree-$2^{2\mu}$ non-scalar endomorphism. The first case cannot happen because P, Q^t are independent. The second case is exponentially rare [LB20].

Working over the base field. It is clear that our basis sampling algorithm is entirely over $\mathbb{F}_p$ until we map Q through the twisting map to obtain $Q^t = \tau(Q)$. We show that we can defer the twisting map to after we have computed the diagonal isogenies, to perform more computation over $\mathbb{F}_p$ and greatly improve the performance of the two-dimensional isogeny computation step.

Diagonal isogenies. We note that to compute the rational maps of the diagonal isogeny φ_{Q^t} we only need the x-coordinate Q^t_x which lies in $\mathbb{F}_p$. Indeed, Vélu's formulae [Vél71] tell us that the 2-isogeny $\varphi \colon E' \to E''$ with kernel (x_0, y_0) is given by

$$\varphi(x, y) = \left(\frac{x^2 - x_0x + t}{x - x_0}, \frac{x - x_0^2}{(x - x_0)^2}y\right).$$

To then evaluate φ_{Q^t} on points (x, y) on $E^t(\mathbb{F}_p)$ we can first "x-twist" the point $(-x, y)$, evaluate $\varphi_{Q^t}((-x, y)) = (x', y')$, and finally "$y$-twist" the point to obtain $(x', \mathrm{i}y') = \varphi_{Q^t}(\tau((x, y)))$ without having done any computation over $\mathbb{F}_{p^2}$.

When implementing higher-dimensional isogenies in the theta model (as is done in the SQIsign [AAA+24] library we adapted), it is advantageous to work with elliptic curves in the Montgomery model. Switching to the Weierstrass model to apply Vélu's formulae is relatively cheap (a few multiplications), however returning back from Weierstrass to the Montgomery model before passing the kernel to the two-dimensional library is quite expensive, requiring a square-root computation.

In general, given a Weierstrass curve $y^2 = x^3 + ax + b$, one can find an isomorphic Montgomery model $\beta^{-1}y^2 = x^3 + 3\alpha\beta^{-1}x^2 + x$ by finding $\alpha^3 + a\alpha + b = 0$ and $\beta^2 = 3\alpha^2 + a$ [CS18, Sec. 2.6]. We note that α must be the x-coordinate of a 2-torsion point, and to recover a Montgomery model $y^2 = x^3 + Ax^2 + x$, we must map through the isomorphism $(x, y) \mapsto (x, y\beta^{-1/2})$. Our implementation uses the kernel of the gluing isogeny (which is generated by order-2 points) to recover α, but must still compute the expensive square roots $\beta = (3\alpha^2 + a)^{1/2}$ and $\beta^{1/2}$. However, by computing $\beta^{1/4}$ directly as an exponentiation, instead of successive square roots, we save roughly half the computation.

We remark that the 2-isogeny formulae in the Montgomery model [Ren18] explicitly avoid the case in which the kernel contains the point $(0, 0)$ — which our use-case requires. Our implementation currently utilises Vélu's formulae and previously described methods. Recent work [RS24] has developed a method for computing generic 2-isogenies between Kummer lines in the Montgomery model by studying the action of the theta group. By using the results of their paper, we have extended their accompanying implementation from working for cyclic 2^n-isogenies whose kernel lies above a specific 2-torsion point R_0, to any 2^n-isogeny. However, our extended implementation is still x-only, and we would have to extend it to a xy-isogeny to map the kernel points of the gluing isogeny (whose y-coordinates we need). We suspect that this can be done using the result [Gal18, Th. 9.7.5], and manipulating the explicit formulae.

The gluing isogeny. To lift sign ambiguities when using the level-2 theta model to glue from the product of Kummer lines into the Kummer surface $E_1/\pm \times E_2/\pm \to A/\pm$, SQIsign's two-dimensional library [AAA+24] requires both x and y coordinates (i.e. "sign information"). Moreover, to compute the action of the theta group, it computes translates (i.e. sums of points), and so requires full $\mathbb{F}_{p^2}$ arithmetic.

Using more symmetric gluing formulae [Dup25, DD26], one can compute gluing isogenies with points on the twist to avoid going into higher field extensions (i.e. staying over $\mathbb{F}_p$). We leave the implementation of these formulae to future work and note that the performance loss through using $\mathbb{F}_{p^2}$ arithmetic in the gluing is quite small ($\approx 3\%$). This is especially true in comparison to SQIsign, because the isogeny chains we compute are much longer.

General 2d-isogeny steps. After computing the gluing isogeny, and mapping the previously doubled points through the gluing, the intermediate abelian surfaces permit an $\mathbb{F}_p$-rational theta model. As such, all general steps can be performed over $\mathbb{F}_p$. Both our Sage and C implementation do this.

Final splitting isogeny. Mapping from the final (non-split) abelian surface $A_n \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$ is done by a change of basis computation, and so encompasses only a few multiplications. A priori, this can also be computed over $\mathbb{F}_p$, but for more direct compatibility with the SQIsign library, our C implementation also performs this over $\mathbb{F}_{p^2}$.

Integrating diagonal isogenies with strategies. The SQIsign [AAA+24] implementation of two-dimensional isogenies employs *strategies* [DJP11]. They are a fundamental tool for efficient evaluation of smooth 2^n-isogenies $\varphi = \varphi_n \circ \cdots \circ \varphi_1$ from a kernel $K = \ker(\varphi)$ description; they turn the naïve evaluation algorithm that is quadratic (in n) to one that is quasilinear by strategically storing intermediate multiples of the kernel K.

Indeed, consider that to compute the kernel K_1 of φ_1 one must compute $2^{n-1}K$, where $K = \ker(\varphi)$; and that storing the intermediate value $2^{n-2}K$ one obtains the kernel K_2 of φ_2 by a single isogeny evaluation $K_2 = \varphi_1(2^{n-2}K)$, instead of $n-2$ doublings of $\varphi_1(K)$. This idea can be extended to devising a *strategy* for the entire isogeny and obtaining a significantly faster evaluation algorithm.

While computing $2^{e-\mu+1}\mathcal{P}_{e-1}$, $2^{e-\mu+1}\mathcal{Q}_{e-1}$ to determine the kernel of the diagonal isogenies, our implementation also stores the intermediate doubles $2^i\mathcal{P}_{e-1}$, $2^i\mathcal{Q}_{e-1}$ that the balanced strategy requires. We then mapped these doubles through the diagonal isogenies and passed them to the two-dimensional library (modified to accept such an input).

Ultimately, this means that the overhead obtained from the diagonal isogenies is minimal, with the only significant concrete cost coming from the two square roots needed to return to the Montgomery model.

Final splitting. Recall that we compute the two-dimensional isogeny $\Phi\colon E \times E \to E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$ embedding $\varphi_{\mathfrak{a}}$ from a description of the kernel of Φ. As such, the isogeny we actually compute Ψ only equals Φ up to post-composition with a (polarised) $\mathbb{F}_p$-isomorphism. In particular, this permits switching the output curves $E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}} \cong E_{\overline{\mathfrak{a}}} \times E_{\mathfrak{a}}$. As described by [PR23, Cor. 2.2], this ambiguity can be lifted mathematically using pairings; however, as described in PEGASIS [DEF+25, App. B], the underlying isogeny evaluation algorithm in the Theta-model implicitly orders the output in a principled way, allowing us to consistently choose the first output curve.