Efficient Post-quantum Commutative Group Actions from Orientations of Large Discriminant

Marc Houben(✉)

Inria Bordeaux, Institut de Mathématiques de Bordeaux, Talence, France
marc.houben@math.u-bordeaux.fr

Abstract. We describe an algorithm to efficiently evaluate class group actions on supersingular elliptic curves that are oriented by an imaginary quadratic order of arbitrarily large discriminant. Contrary to CSIDH, this allows to increase the post-quantum security of the group action without increasing the size of the base field. In particular, we describe instances where Kuperberg's algorithm loses to generic supersingular isogeny path finding. Our algorithm is fully deterministic, strictly constant time, dummy free, and can be implemented without conditional branches. We show that the (restricted effective) group action can be employed in a non-interactive key exchange protocol, that we argue is asymptotically more efficient than CSIDH.

Keywords: Isogeny-based cryptography · orientations · class group actions

1 Introduction

Isogeny-based cryptography is an area of post-quantum cryptography. At the origin of the field lies CRS [39,75], a key exchange scheme based on class group actions on ordinary elliptic curves. CSIDH [22] replaces ordinary curves with supersingular ones, giving rise to the first efficient variant of the protocol. Compared to other post-quantum candidates, CSIDH's main selling points are its small public keys and that it is *non-interactive*. The latter feature is especially compelling; the only practical lattice-based scheme providing a non-interactive key exchange is Swoosh [48], but it uses public keys that are several orders of magnitude larger. CSIDH's main flaw is that it admits subexponential quantum attacks, due to Kuperberg's algorithm [54,55,72]. Although the precise post-quantum security of CSIDH is still subject of debate [8,10,70], recent analyses [26] are rather unfavorable. This has led to present-day work adopting large parameter choices [12,13], severely impacting estimates of the practical performance. The analysis [12, Sec. 7] concludes that—unless significant practical improvements are made—CSIDH is unlikely to be practical in real-world applications, except from niche cases where a non-interactive protocol is required.

G. Hanaoka and B.-Y. Yang (Eds.): ASIACRYPT 2025, LNCS 16248, pp. 141–173, 2026.
https://doi.org/10.1007/978-981-95-5113-2_5

This despite an already large body of existing research dedicated to practical optimizations of the protocol [4,8,12,13,17,18,25,28–30,44,50,63,64,68].

The main problem is that the subexponential quantum attacks force us to significantly increase the size of the base field $\mathbf{F}_p$. In CSIDH, we act on supersingular elliptic curves over $\mathbf{F}_p$ by the class group $\mathrm{Cl}(\mathcal{O})$ of the imaginary order $\mathcal{O} = \mathbf{Z}[\pi]$, where π denotes the p-Frobenius endomorphism. The size of $\mathrm{Cl}(\mathcal{O})$ is roughly proportional to $|\operatorname{Disc}(\mathcal{O})|^{1/2}$, where $\operatorname{Disc}(\mathcal{O}) = \operatorname{Disc}(\pi) = -4p$. Since Kuperberg's algorithm is subexponential in $\#\,\mathrm{Cl}(\mathcal{O})$, this forces the value of p to be large. Recent estimates [12,26] of the required size to attain NIST level 1 range from 2048 to 4096 bits. A crucial factor as to CSIDH's inefficiency is that arithmetic over fields of such large size comes at a significant cost, cf. [12, Sec. 5.2].

In the more general framework of class group actions on oriented elliptic curves, there are methods employing imaginary quadratic orders $\mathcal{O}$ different from the one generated by Frobenius, most notably (PEARL)-SCALLOP [3,27,41]. These schemes work with supersingular elliptic curves over $\mathbf{F}_{p^2}$ instead of over $\mathbf{F}_p$, but—by the method through which the orientation is represented—they are restricted by the similar upper bound $|\operatorname{Disc}(\mathcal{O})| \leq 4p^2$. A recent approach [21] manages to improve on this constraint, essentially by combining the CSIDH and (a generalization of) SCALLOP orientations. Concretely, they consider an orientation by $\mathcal{O} = \mathbf{Z}[\sqrt{-dp}]$, where $d \leq p^2$. Although there are practical limitations, this allows for discriminants asymptotically of size p^3 (still over a base field of size p^2).

Our Contributions

(i) We describe a method to efficiently represent orientations that allows for the discriminant to be arbitrarily large, without increasing the size of the base field. We give a practical method to generate such representations on a supersingular base curve.

(ii) We describe an efficient algorithm for computing the associated restricted effective class group action. It is based on an extension of a recent technique [49] for the CSIDH group action. In particular, the algorithm is fully deterministic, strictly constant time, dummy free, does not have conditional branches, and allows for the evaluation of arbitrary exponent vectors. Moreover, increasing the bitsize of the discriminant by a factor r comes at a cost factor r that parallelizes perfectly across r processors.

(iii) Employing the class group action, we instantiate a non-interactive key exchange protocol which is asymptotically more efficient than CSIDH. We provide a practical example for a 4096-bit discriminant. Our unoptimized proof-of-concept SageMath [81] implementation outperforms an unoptimized SageMath implementation of dCSIDH-4096 [12] by a factor 7 (both running on a single core of a laptop CPU).

Main Idea. We will now describe the main ideas behind our approach. Let p be a prime number for which $p+1$ has many small prime divisors. For example,

the CSIDH-512 prime

$$p = 4 \cdot \underbrace{(3 \cdot 5 \cdot \ldots \cdot 373)}_{73 \text{ consecutive primes}} \cdot 587 - 1 \approx 2^{511}.$$

Let $\mathbf{F}_q = \mathbf{F}_{p^2}$ and let $E/\mathbf{F}_q$ be a supersingular elliptic curve for which $E(\mathbf{F}_q) \cong (\mathbf{Z}/(p+1)\mathbf{Z})^2$. Say that we want to represent a (primitive) orientation on E by an imaginary quadratic order $\mathcal{O} = \mathbf{Z}[\sigma]$ of *large* discriminant. That is, we want to represent a (cyclic) endomorphism σ of E for which

$$|\operatorname{Disc}(\sigma)| = 4\deg(\sigma) - \operatorname{tr}(\sigma)^2$$

is large. It should be stressed that, in what follows, we will only be concerned with how to *represent* such an orientation. How to *find* one (on a given elliptic curve E) is a different question (and in practice requires knowledge of the endomorphism ring of E); this will be discussed in Sect. 4. The main goal for now is to describe a representation that can be efficiently carried through the class group action.

Assuming that the trace of σ can be chosen small, we are concerned with representing endomorphisms of large degree. A natural first approach would be to represent such an endomorphism by its kernel K. However, if we want that $\langle K \rangle \subseteq E(\mathbf{F}_q)$ (for example, if we want to use Vélu isogeny formulae to evaluate the endomorphism using $\mathbf{F}_q$-arithmetic), then this restricts the degree by $\deg(\sigma) \leq p+1$. A trick appearing in (PEARL)-SCALLOP [3,41] is to represent σ as a composition of two isogenies $\widehat{\varphi_2} \circ \varphi_1$, where we are given the kernels of φ_1 and φ_2; this allows for a degree up to $(p+1)^2$.

The first crucial observation is that nothing stops us from decomposing the endomorphism into even more isogenies. Write $M = \prod_{i=1}^{n} \ell_i = (p+1)/4$ for the product of the odd prime divisors of $p+1$, so that $E[M] \subseteq E(\mathbf{F}_q)$. Let σ be an endomorphism of degree M^r, represented as a chain of r isogenies

$$E = E_1 \xrightarrow{\varphi_1} E_2 \xrightarrow{\varphi_2} \ldots \xrightarrow{\varphi_{r-1}} E_r \xrightarrow{\varphi_r} E_{r+1} = E,$$

each of degree M, where $\ker \varphi_j = \langle P_j \rangle$ for $P_j \in E_j(\mathbf{F}_q)$. By choosing r large enough, this allows for a representation of an endomorphism of arbitrarily large degree. Note that such endomorphisms can be efficiently evaluated using only $\mathbf{F}_q$-arithmetic, by computing them as chains of ℓ_i-isogenies whose kernels are generated by $\mathbf{F}_q$-rational points.

We are now going to describe how to evaluate the class group action coming from $\mathcal{O}$-ideals, in such a way that we can always keep track of the orientation. The main idea is based on [49, Algorithm 2], which handles the case $r = 2$. Let us denote by $Q_j \in E_j(\mathbf{F}_q)$ a point generating the kernel of the dual isogeny $\widehat{\varphi_{j-1}}$; see Fig. 1. Let $\ell_i \mid M$ be one of the small primes. The principal $\mathcal{O}$-ideal generated by ℓ_i factors as

$$(\ell_i) = (\ell_i, \sigma)(\ell_i, \hat{\sigma}) = \mathfrak{l}_i \overline{\mathfrak{l}_i}.$$

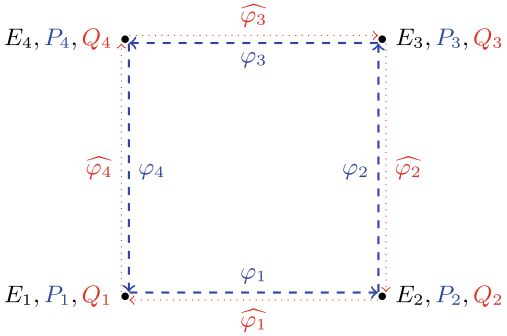

Fig. 1. Graphical representation of a *full kernel representation* of an endomorphism of length $r = 4$.

Note that the isogeny $\varphi_1 : E_1 \to E_2$ corresponds to an action by the $\mathcal{O}$-ideal $\prod_{i=1}^n \mathfrak{l}_i = (M, \sigma)$, because the kernel $\langle P_1 \rangle$ of φ_1 is $\ker \sigma \cap E_1[M]$. In CSIDH's *exponent vector* notation, where the vector $(s_1, \ldots, s_n)$ corresponds to the ideal $\prod_{i=1}^n \mathfrak{l}_i^{s_i}$, this corresponds to $(1, \ldots, 1)$. Similarly, the isogeny $\widehat{\varphi_1}$, whose kernel is generated by Q_2, corresponds to the action by the ideal $\prod_{i=1}^n \overline{\mathfrak{l}_i}$ on E_2, with exponent vector $(-1, \ldots, -1)$.

Suppose now that we want to evaluate the action on E_1 by an ideal of the form $\mathfrak{a} = \prod_{i=1}^n \mathfrak{l}_i^{s_i}$, where the exponent vector $(s_1, \ldots, s_n) \in \{0, 1\}^n$ is *binary*. The ideal corresponds to an isogeny $\varphi_1^+ : E_1 \to E_1'$ of degree $d = \prod_{i=1}^n \ell_i^{s_i}$, and its kernel is generated by the point $[M/d]\, P_1 \in E_1$. Note that the codomain can also be expressed as $E_1' = \prod_{i=1}^n \overline{\mathfrak{l}_i}^{1-s_i} E_2$, corresponding to the action by the ideal with exponent vector $(s_1 - 1, \ldots, s_n - 1)$ on E_2. The latter corresponds to an isogeny $\varphi_1^- : E_2 \to E_1'$ with kernel generated by $[d]Q_2$.

Now, under the crucial assumption that $\langle P_j \rangle \cap \langle Q_j \rangle = \{0\}$,[1] we find that $Q_1' := \varphi_1^+(Q_1) \in E_1'$ has order M, and similarly for $P_1' := \varphi_1^-(P_2) \in E_1'$. In fact, as we will see, we have that $\langle P_1' \rangle = E_1'[M, \sigma]$ and $\langle Q_1' \rangle = E_1'[M, \hat{\sigma}]$, for the induced $\mathcal{O}$-orientation on E_1' (Proposition 2.1).

We have essentially "factored" the isogeny $\varphi_1 : E_1 \to E_2$ as $E_1 \xrightarrow{\varphi_1^+} E_1' \xleftarrow{\varphi_1^-} E_2$. Applying the same trick to every isogeny $\varphi_j : E_j \to E_{j+1}$ in the chain, we recover a representation of an $\mathcal{O}$-orientation on $E' = \mathfrak{a} * E$; see Fig. 2. Iterating the procedure we can compute the action by any ideal class $(s_1, \ldots, s_n) \in \mathbf{Z}^n_{\geq 0}$.

Assuming every isogeny $\varphi_j^\pm$ is computed as a chain of ℓ_i-isogenies, the cost of acting by a binary ideal class is essentially one evaluation of the endomorphism σ. Since σ has degree M^r, it is r times the cost of a CSIDH-512 group action.[2] This method of increasing the discriminant is asymptotically cheaper than increasing

[1] This turns out to be equivalent to the statement that none of the ℓ_i ramify in the class group; this holds in particular when $\deg(\sigma)$ and $\mathrm{tr}(\sigma)$ are coprime.

[2] If we use dCSIDH [12] or [49, Algorithm 3]. This does not (yet) take into account the crucial caveat that we are required to work over $\mathbf{F}_{p^2}$ instead of over $\mathbf{F}_p$.

the size of the base field, because finite field arithmetic scales at best quasi-linearly in the size of the field (in practice, the cost increases significantly, see [12, Sec. 5.2]). Theoretically, we can choose r large enough for Kuperberg's algorithm to lose to generic supersingular isogeny path finding algorithms; a precise trade-off will be discussed in Sects. 5.2 and 6.

Outline. In Sect. 2 we recall the theory of orientations and their associated class group actions. In Sect. 3 we present our algorithm for computing the restricted effective class group action. Section 4 describes how to practically instantiate an orientation on a base curve. In Sect. 5 we discuss a non-interactive key exchange protocol based on the class group action, and perform a security analysis. We also describe how to compress (and decompress) public keys to bitsize $\approx (r + 2)\log_2(p)$. In Sect. 6 we describe practical parameters for an efficient class group action. Finally, in Sect. 7, we present opportunities for future work.

2 Preliminaries

Let k be a perfect field. Our main references concerning the theory of orientations are [34,67].

2.1 Isogenies

An isogeny is a non-constant morphism of elliptic curves. Given an elliptic curve E/k, isogenies φ with domain E are uniquely determined up to post-composition by an isomorphism by their scheme-theoretic kernel, denoted $E[\varphi]$. For separable isogenies $\varphi : E \to E'$, this is the usual (group-theoretic) kernel, consisting of all points in $E(\overline{k})$ that map to the identity element $0 \in E'(k)$. In this case, the cardinality of the kernel equals the degree $\deg \varphi$. We often identify separable isogenies with their kernel, even though this only well-defines the underlying map up to post-composition by an isomorphism. An endomorphism of an elliptic curve E is an isogeny from E to itself, or the zero map. We denote by $\mathrm{End}(E) = \mathrm{End}_{\overline{k}}(E)$ the full endomorphism ring of E, consisting of all endomorphisms defined over $\overline{k}$, where the ring structure is given by point-wise addition and composition.

2.2 Orientations

Let E/k be an elliptic curve and let K be an imaginary quadratic number field. A *K-orientation* is a (necessarily injective) ring homomorphism

$$\iota : K \to \mathrm{End}^0(E) := \mathrm{End}(E) \otimes_{\mathbf{Z}} \mathbf{Q}.$$

The pair (E, ι) is called a *K-oriented elliptic curve*. If (E, ι) is a K-oriented elliptic curve, and $\varphi : E \to E'$ is an isogeny, then the K-orientation induced by φ on E', denoted $\varphi_*(\iota)$, is given by

$$\varphi_*(\iota)(\alpha) := \varphi \circ \iota(\alpha) \circ \hat{\varphi} \otimes \frac{1}{\deg(\varphi)}. \tag{1}$$

Given two K-oriented elliptic curves (E, ι) and (E', ι'), a *K-oriented isogeny* is an isogeny $\varphi : E \to E'$ for which $\varphi_*(\iota) = \iota'$.

If (E, ι) is a K-oriented elliptic curve and $\mathcal{O} \subseteq K$ is an imaginary quadratic order, we call ι an *$\mathcal{O}$-orientation* if $\iota(\mathcal{O}) \subseteq \mathrm{End}(E)$. If (E, ι) is an $\mathcal{O}$-orientation, then for $\sigma \in \mathcal{O} \setminus \{0\}$ we have $\deg \iota(\sigma) = N(\sigma)$, where $N = N_{K/\mathbf{Q}} : K \to \mathbf{Q}$ denotes the algebraic norm. Moreover, we have $\iota(\overline{\sigma}) = \widehat{\iota(\sigma)}$, where $\bar{\cdot}$ denotes the (unique) conjugate (in $K/\mathbf{Q}$) and $\widehat{\cdot}$ denotes the dual isogeny. We will also write $\hat{\sigma}$ for $\overline{\sigma}$. An $\mathcal{O}$-orientation is called *primitive* if it does not extend to a strictly larger imaginary quadratic order. That is, for any strict superorder $\mathcal{O}' \supsetneq \mathcal{O}$ in K, we have that $\iota(\mathcal{O}') \not\subseteq \mathrm{End}(E)$. A K-orientation is primitive for a unique order, called the *primitive order*, given by $\mathcal{O}_{\mathrm{pr}} := \iota^{-1}(\mathrm{End}(E))$. If $\varphi : (E, \iota) \to (E', \iota')$ is a K-oriented isogeny of prime degree ℓ, then exactly one of the following is true:

(i) $\mathcal{O}_{\mathrm{pr}} = \mathcal{O}'_{\mathrm{pr}}$, in which case φ is called *horizontal*;
(ii) $\mathcal{O}_{\mathrm{pr}} \subsetneq \mathcal{O}'_{\mathrm{pr}}$ and $[\mathcal{O}'_{\mathrm{pr}} : \mathcal{O}_{\mathrm{pr}}] = \ell$, in which case φ is called *ascending*;
(iii) $\mathcal{O}'_{\mathrm{pr}} \subsetneq \mathcal{O}_{\mathrm{pr}}$ and $[\mathcal{O}_{\mathrm{pr}} : \mathcal{O}'_{\mathrm{pr}}] = \ell$, in which case φ is called *descending*.

In general, a K-oriented isogeny of composite degree may be any composition of the three above types of isogenies. In case the primitive orders of the domain and codomain are comparable, in the sense that one is contained in the other, we will use the same terminologies (horizontal, ascending, descending) to describe the corresponding isogeny.

2.3 Class Group Actions

Let $\mathcal{O}$ be an order in an imaginary quadratic number field K, and let (E, ι) be an $\mathcal{O}$-oriented elliptic curve. If $\mathfrak{a} \subseteq \mathcal{O}$ is an ideal of norm coprime to $\mathrm{char}\, k$, we define its *kernel* by

$$E[\mathfrak{a}] := \bigcap_{\alpha \in \mathfrak{a}} E[\iota(\alpha)],$$

and denote by $\varphi_{\mathfrak{a}} : E \to E'$ an isogeny with kernel $E[\mathfrak{a}]$. The codomain curve $E'/\overline{k}$, unique up to isomorphism, is also denoted $\mathfrak{a} * E$. The induced K-oriented isogeny

$$(E, \iota) \to (\mathfrak{a} * E, (\varphi_{\mathfrak{a}})_*(\iota))$$

is horizontal if and only if $\mathfrak{a}$ is invertible (if $\mathfrak{a}$ is not invertible, then it is ascending) [67, Prop. 3.5]. Moreover, if $\mathfrak{b}$ is an equivalent $\mathcal{O}$-ideal, then

$$(\mathfrak{a} * E, (\varphi_{\mathfrak{a}})_*(\iota)) \cong (\mathfrak{b} * E, (\varphi_{\mathfrak{b}})_*(\iota))$$

as K-oriented elliptic curves. This induces an action

$$\mathrm{Cl}(\mathcal{O}) \circlearrowright \{(E, \iota) \mid E/\overline{k} \text{ ell. curve, } \iota \text{ an } \mathcal{O}\text{-orientation}\}/\cong$$

by the ideal class group $\mathrm{Cl}(\mathcal{O})$ of $\mathcal{O}$ on the (possibly empty) set of $\mathcal{O}$-oriented elliptic curves over $\overline{k}$ up to K-oriented isomorphism. If we restrict to primitive $\mathcal{O}$-orientations, then this action is free. We will frequently make use of the following proposition.

Proposition 2.1. *Let $\mathcal{O} = \mathbf{Z}[\sigma]$ be an imaginary quadratic order and let (E, ι) be a primitively $\mathcal{O}$-oriented elliptic curve over k. Assume that $N(\sigma)$ is coprime to both* $\operatorname{char} k$ *and* $\operatorname{Disc}(\mathcal{O})$ *(or, equivalently, to* $\operatorname{char} k$ *and* $\operatorname{tr}(\sigma)$*).*

(a) We have $E[\iota(\sigma)] \cap E[\iota(\hat{\sigma})] = \{0\}$.
(b) If $m \in \mathbf{Z}_{>0}$ such that $m \mid N(\sigma)$, then the $\mathcal{O}$-ideal $\mathfrak{a} := (m, \sigma)$ is invertible. Moreover, denoting by $\varphi_{\mathfrak{a}} : E \to \mathfrak{a} * E$ the corresponding isogeny, we have $(\mathfrak{a} * E)[\iota'(\hat{\sigma})] = \varphi_{\mathfrak{a}}(E[\iota(\hat{\sigma})])$, where ι' denotes the orientation induced by $\varphi_{\mathfrak{a}}$.

Proof. (a) This follows from [49, Cor. 2.2].
(b) The statement that $\mathfrak{a}$ is invertible follows from the fact that $N(\sigma)$ is coprime to the conductor f of $\mathcal{O}$ (indeed, $f \mid \operatorname{Disc}(\mathcal{O})$). The second statement follows from [49, Prop. 3.4]. □

Given an $\mathcal{O}$-oriented curve (E, ι) and an element $\sigma \in \mathcal{O}$, we will also write $E[\sigma] := E[\iota(\sigma)] = E[(\sigma)]$ for the kernel of $\iota(\sigma)$. Similarly, if (m, σ) is an $\mathcal{O}$-ideal, where $m \in \mathbf{Z}_{>0}$, we denote its kernel by $E[m, \sigma] := E[(m, \sigma)]$.

2.4 Representing Orientations

Let $E/\mathbf{F}_q$ be an elliptic curve and let $\mathcal{O}$ be an imaginary quadratic order. We say an *effective representation* of an orientation $\iota : \mathcal{O} \hookrightarrow \operatorname{End}(E)$ consists of the following data:

(i) The minimal polynomial $f \in \mathbf{Z}[X]$ of a generator σ of $\mathcal{O}$ (or, equivalently, the norm and trace of σ);
(ii) An algorithm to evaluate $\iota(\sigma)$ on any point $P \in E(\mathbf{F}_{q^r})$ for any $r \in \mathbf{Z}_{>0}$.

A related notion that appears in the literature is that of an *efficient* representation, which additionally requires that the evaluation algorithm (ii) executes in polynomial time, see [83, p. 7-8] for a precise definition. In this paper, our main focus will be on a specific type of effective (in fact, efficient) representation, where the endomorphism $\iota(\sigma)$ is given as a composition of isogenies of smooth degree, each given by an $\mathbf{F}_q$-rational point P generating the kernel. We call this a *kernel representation*.

Kernel Representations. For the purpose of the exposition, let us momentarily return to the general case where the base field is k. Let E/k be an elliptic curve and let $\mathcal{O} = \mathbf{Z}[\sigma]$ be an imaginary quadratic order. An $\mathcal{O}$-orientation ι on E is uniquely determined by the image of σ as an element of $\operatorname{End}(E)$. Suppose that the endomorphism $\iota(\sigma)$ of E decomposes as a chain of isogenies

$$E = E_1 \xrightarrow{\varphi_1} E_2 \xrightarrow{\varphi_2} \dots \xrightarrow{\varphi_{r-1}} E_r \xrightarrow{\varphi_r} E_{r+1} = E,$$

for some $r \in \mathbf{Z}_{>0}$. Assuming that the orientation is primitive, each φ_j is necessarily cyclic. In case points $P_j \in E_j(k)$ are given such that $\ker \varphi_j = \langle P_j \rangle$, we call the sequence $\{(E_j, P_j) \mid 1 \le j \le r\}$ a *kernel representation* of the

$\mathcal{O}$-orientation on E. The integer $r \in \mathbf{Z}_{>0}$ is called the *length* of the representation. If we are also given points $Q_j \in E_j(k)$ such that $\ker(\widehat{\varphi_{j-1}}) = \langle Q_j \rangle$,[3] then $\{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$ is called a *full* kernel representation. Although a full kernel representation will in practice be (computationally) recoverable from the kernel representation, we record this notion because we will require the points Q_j in our algorithm to compute the class group action. In SCALLOP [41, Definition 6] and PEARL-SCALLOP [3], the orientation admits a kernel representation of length $r = 2$. CSIDH [22] and CSURF [15] can be viewed as the case $r = 1$, if we choose the generators $\sigma = \pi - 1$ and $\sigma = \frac{\pi - 1}{2}$ respectively, where π denotes the p-Frobenius endomorphism, cf. [49, Sec. 2.4].

3 Restricted Effective Group Actions

Let k be a perfect field. The goal of this section is to describe an algorithm to evaluate the (restricted effective) class group action on primitively oriented elliptic curves over k, assuming that the orientation is given by a kernel representation as described in Sect. 2.4. This can be seen as a generalization of [49], which considers the case $r \in \{1, 2\}$. Though in practice we will always work with elliptic curves over finite fields, we will describe the algorithms over the general base field k; all we require is that we can (efficiently) perform elliptic curve arithmetic over k and compute isogenies from their kernel (e.g. using Vélu's formulae [82]). More specifically, we assume that we have the following two black-box algorithms at our disposal.

(i) $\mathsf{Multiply}(E, P, N)$. **Input:** an elliptic curve E/k, a point $P \in E(k)$, and an integer $N \in \mathbf{Z}_{>0}$. **Output:** $[N]P$.
(ii) $\mathsf{Isogeny}(E, \mathcal{L}, K, N)$. **Input:** an elliptic curve E/k, a tuple of points $\mathcal{L} = (P_1, \ldots, P_n)$, where $P_i \in E(k)$, and a kernel point $K \in E(k)$ of order $N \in \mathbf{Z}_{>0}$. **Output:** $E', (\varphi(P_1), \ldots, \varphi(P_n))$, where $E' \cong E/\langle K \rangle$ is the codomain of a cyclic isogeny $\varphi : E \to E'$ with kernel $\langle K \rangle$.

Note that the algorithm $\mathsf{Isogeny}$ depends intrinsically on n, the length of the evaluation tuple. We will mainly consider the case $n = 2$. In particular, if we denote by $R = (E, P, Q)$ is a triple where E/k is an elliptic curve an $P, Q \in E(k)$, we will also write $\mathsf{Isogeny}(R, K, N)$ for the function that calls $\mathsf{Isogeny}(E, (P, Q), K, N)$ and returns the triple $R' = (E', \varphi(P), \varphi(Q))$.

3.1 The Imaginary Quadratic Order

Let $M, r \in \mathbf{Z}_{>0}$ and let $\mathcal{O} = \mathbf{Z}[\sigma]$ be an imaginary quadratic order generated by an element σ of norm

$$N(\sigma) = M^r.$$

[3] Here we adopt the notational convention that $\varphi_0 = \varphi_r$.

In practical instantiations, the integer M will always be smooth and r will always be small. We will assume throughout that

$$\gcd(M, \mathrm{Disc}(\mathcal{O})) = 1. \tag{2}$$

This is equivalent to $\gcd(N(\sigma), \mathrm{tr}(\sigma)) = 1$. Write

$$M = \prod_{i=1}^{n} \ell_i^{e_i}$$

for the prime factorization of M. Then (2) implies that the principal $\mathcal{O}$-ideal generated by σ factors as

$$(\sigma) = \prod_{i=1}^{n} (\ell_i, \sigma)^{re_i} = \prod_{i=1}^{n} \mathfrak{l}_i^{re_i} = \mathfrak{e}^r,$$

where

$$\mathfrak{e} := \prod_{i=1}^{n} \mathfrak{l}_i^{e_i} \tag{3}$$

and each $\mathfrak{l}_i := (\ell_i, \sigma)$ is an invertible $\mathcal{O}$-ideal. We will also write $\overline{\mathfrak{l}_i} = (\ell_i, \overline{\sigma})$ for the conjugate ideals.

3.2 The Kernel Representation

Let $\mathcal{O} = \mathbf{Z}[\sigma]$ as above. Suppose that the characteristic of k is coprime to M and let E/k primitively oriented by $\mathcal{O}$. The cyclic endomorphism $\iota(\sigma)$ can be decomposed into a sequence of r isogenies of degree M. Explicitly, if $P \in E(\overline{k})$ is a generator for $E[\sigma]$, then $\iota(\sigma) = \varphi_r \circ \ldots \circ \varphi_1$, where $\ker \varphi_1 = \langle M^{r-1} P \rangle$ and

$$\ker \varphi_j = \langle M^{r-j} \varphi_{j-1} \circ \ldots \circ \varphi_1(P) \rangle = E[M] \cap \langle \varphi_{j-1} \circ \ldots \circ \varphi_1(P) \rangle.$$

Defining $P_j := M^{r-j} \varphi_{j-1} \circ \ldots \circ \varphi_1(P)$, we have that $\{(E_j, P_j) \mid 1 \leq j \leq r\}$ is a kernel representation of the $\mathcal{O}$-orientation on E. Each φ_j is necessarily horizontal; in fact, it corresponds to the action by the ideal $\mathfrak{e}$ as in (3), that is,

$$\ker \varphi_j = \langle P_j \rangle = E_j[M, \sigma] = E_j[\mathfrak{e}].$$

3.3 Evaluating the Class Group Action

Adopting the notation of the previous section, let $Q_j \in E_j(\overline{k})$ such that $\langle Q_j \rangle = \ker \widehat{\varphi_{j-1}}$, so that $\{(E_j, P_j, Q_j)\}$ is a full kernel representation of the $\mathcal{O}$-orientation on E. Assume moreover that $P_j, Q_j \in E_j(k)$ for all j. Let $\mathfrak{a}$ be an ideal of the form

$$\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{s_i},$$

where $0 \leq s_i \leq e_i$ for all $1 \leq i \leq n$. We are going to describe an algorithm to compute $E' := \mathfrak{a} * E$ together with a full kernel representation $\{(E'_j, P'_j, Q'_j)\}$ of the induced $\mathcal{O}$-orientation on E' (where $E'_1 = E'$). Denote by $d = \prod_{i=1}^{n} \ell_i^{s_i}$ the norm of $\mathfrak{a}$. Since $E_1[\mathfrak{a}] \subseteq E_1[\mathfrak{e}] = E_1[M, \sigma]$, the latter being cyclic and generated by P_1, we have

$$E_1[\mathfrak{a}] = \langle [M/d]P_1 \rangle.$$

Denote by $\varphi_1^+ : E_1 \to E'_1$ the corresponding isogeny. Now consider the $\mathcal{O}$-ideal $\mathfrak{b} := \prod_{i=1}^{n} \mathfrak{l}_i^{e_i - s_i}$. Since $\mathfrak{e} = \mathfrak{a}\mathfrak{b}$, we have $[\mathfrak{b}] * E'_1 \cong [\mathfrak{e}] * E_1 \cong E_2$, hence $\overline{\mathfrak{b}} * E_2 \cong E'_1$. The isogeny $\varphi_1^- : E_2 \to E'_1$ corresponding to the action by $\overline{\mathfrak{b}}$ on E_2 has kernel

$$E_2[\overline{\mathfrak{b}}] = \langle [d]Q_2 \rangle.$$

Similarly, for any $1 \leq j \leq r$, we define isogenies

$$E_j \xrightarrow{\varphi_j^+} E'_j \xleftarrow{\varphi_j^-} E_{j+1}, \tag{4}$$

with kernels $\langle [M/d]P_j \rangle$ and $\langle dQ_{j+1} \rangle$ respectively, corresponding to the action by $\mathfrak{a}$ on E_j and $\overline{\mathfrak{b}}$ on E_{j+1} respectively. We set $P'_j := \varphi_j^-(P_{j+1})$ and $Q'_j := \varphi_j^+(Q_j)$.[4]

Proposition 3.1. *The data $\{(E'_j, P'_j, Q'_j)\}$ as defined above is a full kernel representation of the $\mathcal{O}$-orientation on E' induced by the isogeny $\varphi_{\mathfrak{a}} : E \to E'$.*

Proof. It follows from Proposition 2.1 that

$$\langle P'_j \rangle = E_j[M, \sigma] = E_j[\mathfrak{e}] \qquad \text{and} \qquad \langle Q'_j \rangle = E_j[M, \hat{\sigma}] = E_j[\overline{\mathfrak{e}}]. \tag{5}$$

Now note that

$$E'_{j+1} \cong [\mathfrak{a}] * E_{j+1} \cong [\mathfrak{a}] * [\mathfrak{e}] * E_j \cong [\mathfrak{e}] * [\mathfrak{a}] * E_j \cong [\mathfrak{e}] * E'_j,$$

hence $E'_{j+1} \cong E'_j / \langle P'_j \rangle$. Finally, by (5), the isogenies corresponding to $\langle P_j \rangle$ and $\langle Q_{j+1} \rangle$ are necessarily dual to each other. □

Since $M = \prod_{i=1}^{n} \ell_i^{e_i}$, any isogeny of degree dividing M is a composition of some amount of ℓ_i-isogenies for various i. This applies in particular to the isogenies φ_j^+ and φ_j^-, whose degrees are d and M/d respectively. From a computational perspective, the total number of ℓ_i-isogenies required to compute both φ_j^+ and φ_j^- is independent of d (it is equal to e_i). If we call the factors of φ_j^+ and φ_j^- *positive* and *negative* isogenies respectively, then the decomposition (4) is completely determined by the number of positive ℓ_i-isogenies (this is s_i). Algorithmically, for each value of j, we will always perform e_i isogenies of degree ℓ_i, and switch between the sign (positive or negative) of the isogeny using a conditional swap (cswap) function. This gives rise to Algorithm 1 for computing the class group action on $\mathcal{O}$-oriented elliptic curves. Here R_0 and R_1 represent abstract memory registers that store elliptic curves together with torsion points, and we understand $R[j]$ to mean the j-th element of the register, starting at zero. The function $\mathsf{cswap}(R_0, R_1, b)$ swaps the contents of the registers if the boolean b is 1.

[4] By notational convention, we say $E_{r+1} = E_1$, $P_{r+1} = P_1$, and $Q_{r+1} = Q_1$.

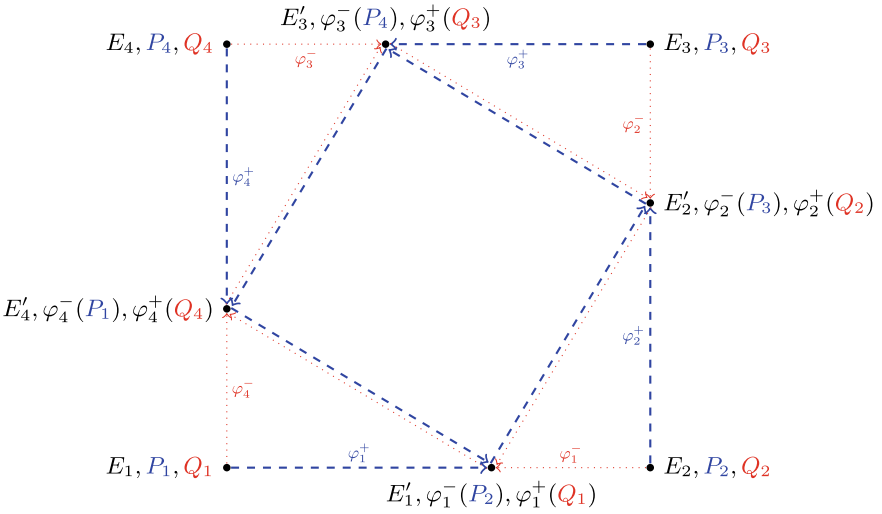

Fig. 2. Computing $E_1' = [\mathfrak{a}] * E_1$ while keeping track of a full kernel representation of the orientation.

Algorithm 1. Evaluating the class group action on full kernel representations

Input: A full kernel representation $\{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$ of a $\mathbf{Z}[\sigma]$-orientation on E, together with a vector of integers $(s_1, \ldots, s_n) \in [0, e_i]^n$.
Output: A full kernel representation $\{(E_j', P_j', Q_j')\}$ of a $\mathbf{Z}[\sigma]$-orientation on $E' = \left[\prod_{i=1}^n \mathfrak{l}_i^{s_i}\right] * E$.

for $j = 1, \ldots, r$ **do**
  $R_0 \leftarrow (E_j, P_j, Q_j)$;
  $R_1 \leftarrow (E_{j+1}, Q_{j+1}, P_{j+1})$;
  $m \leftarrow \prod_i \ell_i^{e_i}$;
  for $i = 1, \ldots, n$ **do**
    for $t = 1, \ldots, e_i$ **do**
      $b \leftarrow t > s_i$;
      $\mathsf{cswap}(R_0, R_1, b)$;
      $m \leftarrow m/\ell_i$; $K \leftarrow \mathsf{Multiply}(R_0[1], m)$;
      $R_0 \leftarrow \mathsf{Isogeny}(R_0, K, \ell_i)$;
      $R_1[1] \leftarrow \mathsf{Multiply}(R_1[1], \ell_i)$;
      $\mathsf{cswap}(R_0, R_1, b)$;
    end for
  end for
  assert $R_0[0] = R_1[0]$;
  $R_0[1] \leftarrow R_1[2]$;
  $R_j' \leftarrow R_0$;
end for
return $R_1', \ldots, R_r'$;

Algorithm 1 for Humans. Consider the algorithm as described in Algorithm 1 and graphically depicted in Fig. 2. The outer loop of the algorithm iterates over the (outer) edges of the square, each corresponding to an isogeny $E_j \xrightarrow{\varphi_j} E_{j+1}$. Let us consider what happens for $j = 1$. The goal of the algorithm is to factor the isogeny φ_1 into an isogeny φ_1^+ of degree d and (the dual of) an isogeny φ_1^- of degree M/d.

The Setup. The two memory registers R_0 and R_1 store elliptic curves together with torsion points. Initially, they correspond to the curves E_1 and E_2 respectively, together with a full M-torsion basis on each curve. The point P_1, which is at index 1 in R_0, generates the kernel of the M-isogeny $\varphi_1 : E_1 \to E_2$. The point Q_2, which is at index 1 in R_1, generates the kernel of the M-isogeny $\widehat{\varphi_1} : E_2 \to E_1$. Informally, we say that E_1 and E_2 are at *distance* M. This *distance* will be represented by the cofactor m (which is indeed initialized at value M).

Walking Along the Edge. Each value of i corresponds to one *isogeny degree* ℓ_i. The algorithm always computes exactly e_i isogenies of degree ℓ_i. Depending on the value of b, the isogeny to be computed will be either a factor of φ_1 (i.e. walking to the right in Fig. 2) or of $\hat{\varphi}_1$ (i.e. walking to the left in Fig. 2). The cswap function ensures that we are computing the correct isogeny. That is, it makes sure that the domain of the isogeny to be computed is (temporarily) stored in register R_0.

The Kernel Point. The kernel of the isogeny is generated by the point K of order ℓ_i, which is always obtained by multiplying the point at index 1 in R_0 by the cofactor; this is achieved by the first execution of the function Multiply in of the inner-most loop.

Isogeny and Multiply. The ℓ_i-isogeny step, which consists of the execution of the function Isogeny, consists of computing the codomain curve, and evaluating the isogeny at both torsion points stored in R_0. This decreases the order of the point at index 1 by a factor ℓ_i, while the point at index 2 always remains of order M. We then make sure that the point at index 1 in R_1 (which lives on the other curve) also decreases by a factor of ℓ_i; this is ensured by the (second) execution of Multiply.

After One Iteration. At the end of the inner-most loop, the orders of the points at index 1 are equal to m; this is new *distance* between the curves stored in R_0 and R_1.

Completing the Algorithm. After performing all iterations, the curves the two registers are equal (i.e. they are at *distance* 1) to E_1'. The points at index 2 form an M-torsion basis, which (partially) represents the $\mathcal{O}$-orientation on E_1'. We store E_1', together with the M-torsion basis, in the output register R_1'.

3.4 Complexity Analysis

From a high-level point of view, we factor the isogeny $\iota(\sigma)$ into $2r$ isogenies $\varphi_j^{\pm}$ of smaller degree, and compute each as a composition of several ℓ_i-isogenies,

where every ℓ_i-isogeny is evaluated at two points; one generating the kernel of the isogeny chain and one additional point. Since the product of the degrees of the $\varphi_j^{\pm}$ is M^r, this is essentially equivalent to computing the endomorphism $\iota(\sigma)$ as a chain of ℓ_i-isogenies, and evaluating it at one point.

Isogeny and Multiply. Apart from the comparison $t > s_i$, the cost of the algorithm is dominated by calls to the functions Multiply and Isogeny. Although it seems we still require some integer divisions to determine the cofactors m, the sequence of cofactors is actually fixed and can thus be precomputed. Assuming that the running time of $\mathsf{Multiply}(-, N)$ depends only on N and $\mathsf{Isogeny}(-, -, \ell_i)$ depends only on ℓ_i,[5] every iteration of the outer loop (indexed by j) runs in strictly constant time. In fact, these iterations are independent of each other, so theoretically parallelize perfectly across r processors.

Let us analyze the complexity of one such iteration. By construction, we necessarily compute e_i isogenies of degree ℓ_i and perform e_i multiplications by a factor ℓ_i for every $1 \leq i \leq n$. In addition to this, there are cofactor multiplications by a predetermined list of (larger) integers m. In practical instantiations, we can partially mitigate the cost of these cofactor multiplications by a trade-off known as a *strategy*; in a nutshell, by splitting up the M-torsion into multiple components, we can reduce the size of the cofactors at the expense of evaluating the isogenies at more points. This is a trick originally introduced in SIDH for the computation of isogenies whose degree is a power of a prime, and "optimal" strategies are known in this setting [51, Sec. 4.2.2]. In case of a general isogeny of smooth degree, determining the optimal trade-off is a more complicated problem, and a significant amount of research has been done for the special case of CSIDH [28,30,50,63,68]. We remark that the best strategy depends on the value of M and the practical complexities of Isogeny and Multiply.

Comparison to the CSIDH Group Action. Back to a high-level point of view, let us consider what happens in the special case where the base field is a finite field $\mathbf{F}_q = \mathbf{F}_{p^2}$, where $M = \prod_{i=1}^n \ell_i$ (that is, $e_i = 1$ for all i) and $M \mid (p+1)$. Assume that we are working with supersingular elliptic curves with $(p+1)^2$ points, so that the kernel representation of the orientation can be defined over the base field. As we saw above, the cost of evaluating a *binary* ideal class $\prod_{i=1}^n \ell_i^{s_i}$, corresponding to an exponent vector $(s_1, \ldots, s_n) \in \{0,1\}^n$, is roughly equivalent to the evaluation of the endomorphism representing the orientation as a chain of ℓ_i-isogenies. This endomorphism has degree M^r.

In dCSIDH [12]—the current state of the art for dummy-free constant-time CSIDH—a binary ideal class has exponent vector in $\{-1,1\}^n$ (as opposed to $\{0,1\}^n$). To evaluate it requires the computation of an isogeny of degree M as a chain of ℓ_i-isogenies, which is thus computationally comparable to the case

[5] These assumptions can be satisfied in practice if we use (optimal) differential addition chains for Multiply and Vélu [82] or $\sqrt{\text{élu}}$ [7] isogeny formulae for Isogeny, provided that we can perform field arithmetic in constant time.

$r = 1$. As we will see below, Algorithm 1 allows in practice to use orientations with discriminants that are larger (in bitsize) by a factor of r. By the above discussion, this comes at a computational cost factor of exactly r (which can moreover be mitigated by parallelizing across r processor cores). To increase the discriminant when restricting to CSIDH is asymptotically more expensive, since it requires to increase the size of the base field, which increases the cost of finite field arithmetic; the latter scales at best quasi-linearly with r.[6] Here we remark that, if we are allowed to use dummy operations,[7] there is an additional trick for constant-time CSIDH which is incompatible with Algorithm 1, using *Matryoshka isogenies* [4,8]. These are employed in dCTIDH [13], the current state of the art for constant-time CSIDH. We leave it to future work to see how an optimized implementation of Algorithm 1 fares in comparison to both dCSIDH and dCTIDH, although we do give a practical example in Sect. 6.3. Lastly, we remark that a version of Algorithm 1 can also be applied to CSIDH directly, cf. [49, Algorithm 3], where it enjoys several advantages compared to dCSIDH and dCTIDH; see [49, Sec. 4.2] for a discussion.

4 Generating Public Parameters

To obtain practical instantiations of the class group action described in Sect. 3, we require at least one primitively oriented elliptic curve, together with a kernel representation of the orientation. The goal of this section is to describe a method to find such instantiations. Let $p \equiv 3 \pmod 4$ be a prime number and let $M \mid (p+1)$ be a positive integer. Write

$$p + 1 = \prod_{i=1}^{n} \ell_i^{f_i} \qquad \text{and} \qquad M = \prod_{i=1}^{n} \ell_i^{e_i}, \tag{6}$$

for the respective prime factorizations, where $0 \leq e_i \leq f_i$ for all $1 \leq i \leq n$. For all practical purposes, the number M will be assumed smooth.

4.1 Orienting a Base Curve

Denote by $\mathbf{F}_q = \mathbf{F}_p(\sqrt{-1})$ a field with p^2 elements and let $E/\mathbf{F}_q$ be the supersingular elliptic curve $E : y^2 = x^3 + x$. Denote by $\pi : E \to E, (x, y) \mapsto (x^p, y^p)$ the p-Frobenius endomorphism and by $i : E \to E$ the automorphism $(x, y) \mapsto (-x, \sqrt{-1}y)$. Then the endomorphism ring of E is given by $\mathrm{End}(E) = \mathbf{Z}[1, i, \frac{i+\pi}{2}, \frac{1+i\pi}{2}]$. We have $\#E(\mathbf{F}_q) = (p+1)^2$ and $E[M] \subseteq E(\mathbf{F}_q)$.

[6] An important remark is that the cost of a non-Frobenius orientation is already larger to begin with, since it requires to work over $\mathbf{F}_{p^2}$. This overhead can be partially mitigated by using points on the *twist*; see Sect. 6.2.

[7] That is, operations whose results may be discarded but whose effect is to mask a variable running time. Such operations are potentially susceptible to *fault injections* [5,14,57–59].

The Norm Equation. Let $r \in \mathbf{Z}_{>0}$. Suppose that we want to find an orientation on E by an imaginary quadratic order $\mathcal{O} = \mathbf{Z}[\sigma]$, where σ has norm M^r. Given the explicit description of the endomorphism ring, this amounts to solving the norm equation

$$N(\sigma) = N\left(a + bi + c\frac{i+\pi}{2} + d\frac{1+i\pi}{2}\right) = M^r,$$

where $a, b, c, d \in \mathbf{Z}$. After the linear change of variables $x := 2a + d$, $y := 2b + c$, we can rewrite as

$$x^2 + y^2 + p(c^2 + d^2) = 4M^r.$$

where $\mathrm{tr}(\sigma) = x$.

The Discriminant. For our purposes, we will additionally require that $|\operatorname{Disc}(\sigma)| = 4N(\sigma) - \mathrm{tr}(\sigma)^2 = 4M^r - x^2$ be either

(i) prime; or
(ii) the product of two large (distinct) primes.

Note that the second case is best we can hope for when $r = 2s$ is even, by the factorization

$$4M^{2s} - x^2 = (2M^s + x)(2M^s - x).$$

Although this requirement on the discriminant is not strictly necessary, we will argue why such a discriminant is desirable in Sect. 5.2. Note that it implies in particular that any $\mathbf{Z}[\sigma]$-orientation is automatically primitive.

Finding the Orientation. We obtain the following pseudo-algorithm for finding an element of $\mathrm{End}(E)$ with the required properties.

(i) Set x to be the smallest (odd) prime number different from the ℓ_i.
(ii) Verify that
  (a) if r is odd: $4M^r - x^2$ is prime;
  (b) if r is even: $2M^{r/2} + x$ and $2M^{r/2} - x$ are prime.
  If this fails, try again with next smallest value for x.
(iii) Solve $x^2 + y^2 \equiv 0 \pmod{p}$ for y, and lift to $y \in 2\mathbf{Z}_{>0}$ minimal. If no solution exists, start over with the next smallest value of x.
(iv) Solve $\left(4M^r - (x^2 + y^2)\right)/p = c^2 + d^2$ using Cornacchia's algorithm, where c is even and d is odd. If no solution exists, start over with the next value of x.

Note that, since the output to (iii) is expected to be of size $\approx p$, we only expect step (iv) to succeed if $4M^r$ is at least of size $\approx p^2$. This will not be a restriction for our purposes, since we are most interested in the case where $M^r > p^2$. In practice, we will only attempt (iv) after verifying that $\left(4M^r - (x^2 + y^2)\right)/p$ is a (positive) prime $\equiv 1 \pmod 4$, since Cornacchia's algorithm requires a factorization of the integer that is to be represented by the quadratic form. Note that in this situation we can always guarantee a solution where c is even and d is odd. The latter condition is necessary to ensure that $a = (x - d)/2$ and $b = (y - c)/2$ are both integral. Finally, we remark that since x is coprime to M by construction, the order $\mathcal{O} = \mathbf{Z}[\sigma]$ automatically satisfies (2).

Example 4.1. Consider the prime number

$$p = 2^{12} \cdot 3^6 \cdot 5^4 \cdot \underbrace{(7 \cdot 11 \cdot \ldots \cdot 281)}_{57 \text{ consecutive primes}} - 1 \approx 2^{409.2}.$$

Then $E/\mathbf{F}_{p^2} : y^2 = x^3 + x$ is orientable by an order $\mathcal{O} = \mathbf{Z}[\sigma]$ for which

$$N(\sigma) = M^5 \qquad \text{and} \qquad \mathrm{tr}(\sigma) = 1800301,$$

where $M = p + 1$. Moreover, the integer

$$|\operatorname{Disc}(\mathcal{O})| = |\operatorname{Disc}(\sigma)| = 4M^5 - 1800301^2 \approx 2^{2048}$$

is prime.

4.2 Obtaining a Kernel Representation

Now that we have obtained an element $\sigma \in \mathrm{End}(E)$ of norm M^r given as an explicit $\mathbf{Z}$-linear combination

$$\sigma = a + bi + c\frac{i+\pi}{2} + d\frac{1+i\pi}{2} \tag{7}$$

of the endomorphisms $1, i, \frac{i+\pi}{2}, \frac{1+i\pi}{2}$, we would like obtain a kernel representation of the orientation. That is, we want to represent σ as a composition of r cyclic isogenies of degree M. Since $E[M] \subseteq E(\mathbf{F}_q)$ (and similarly for any curve isogenous to E), the kernel of every such isogeny is generated by an $\mathbf{F}_q$-rational point, i.e. there exists a kernel representation $\{(E_j, P_j)\}$ of the $\mathcal{O}$-orientation, where $P_j \in E_j(\mathbf{F}_q)$ of order M for all $1 \leq j \leq r$.

The Base Case. Consider first the orientation on $E_1 = E : y^2 = x^3 + x$. The desired point $P_1 \in E_1(\mathbf{F}_q)$ is a generator for $E[M, \sigma]$. If $R \in E[M]$ is any point, then $S = \hat{\sigma}(R) \in E[M, \sigma]$, where $\hat{\sigma}$ denotes the dual endomorphism. Given σ as in (7), the dual isogeny corresponds to the quaternion conjugate; knowing $(a, b, c, d) \in \mathbf{Z}^4$, it can be explicitly evaluated on points of $E(\mathbf{F}_q)$. We can thus sample random points $R_i \in E(\mathbf{F}_q)$, and list their images $S_i = \hat{\sigma}(R_i)$. The latter will span $E[M, \sigma]$ once the least common multiple of their orders is M; in this case we compute P_1 as a suitable linear combination of the S_i. A generator $Q_1 \in E_1(\mathbf{F}_q)$ for $E[M, \hat{\sigma}]$ can be obtained in an analogous manner, by evaluating random M-torsion points under σ.

Pushing Forward the Orientation. Given the point P_1, we can compute the curve $E_2 \cong E_1/\langle P_1 \rangle \cong [\mathfrak{e}] * E_1$, where we recall that $\mathfrak{e}$ is the $\mathcal{O}$-ideal (M, σ) (cf. (3)). Denote by $\varphi_1 : E_1 \to E_2$ the corresponding cyclic isogeny with kernel $E_1[M, \sigma]$. Since $E[\sigma] \cap E[\hat{\sigma}] = \{0\}$, the image $Q_2 := \varphi_1(Q_1)$ generates $E_2[M, \hat{\sigma}]$. In what follows, we will describe how to obtain a generator P_2 for $E_2[M, \sigma]$.

Consider the $\mathcal{O}$-ideal $I := (M^{r-1}, \hat{\sigma}) = \overline{\mathfrak{e}}^{r-1}$. Since $[\mathfrak{e}] = [\overline{\mathfrak{e}}]^{-1} = [\overline{\mathfrak{e}}]^{r-1}$ in $\mathrm{Cl}(\mathcal{O})$, it corresponds to a (horizontal) isogeny of degree M^{r-1}

$$\psi = \varphi_I : E_1 \to E_2, \quad \text{where} \quad \ker \psi = E_1[M^{r-1}, \hat{\sigma}].$$

Note that, by the same reasoning as before, we have that $\psi(P_1)$ generates $E_2[M, \sigma]$. Unfortunately, it is not immediately clear how to efficiently evaluate ψ if $r > 2$, since a generator for $\ker \psi$ may only be defined over a possibly large extension field of $\mathbf{F}_q$. Luckily, we are saved by the Deuring correspondence.

Denote by $\mathfrak{O}_0 \cong \mathbf{Z}\left[1, i, \frac{i+\pi}{2}, \frac{1+i\pi}{2}\right]$ the full endomorphism ring of E_1, viewed as a maximal order of the quaternion algebra $B_{p,\infty}$. We can extend I to a left $\mathfrak{O}_0$-ideal $I_0 := \mathfrak{O}_0 I$, so that $\psi = \varphi_{I_0}$ through the Deuring correspondence. Using the KLPT algorithm [53], we can compute a left $\mathfrak{O}_0$-ideal J_0 equivalent to I_0. Moreover, the ideal J_0 can be chosen of pleasant norm, in the sense that

(i) we may require that $N(J_0)$ is coprime to M;
(ii) translating J_0 to an isogeny φ_{J_0} via the Deuring correspondence is practical; that is, its kernel is generated by torsion points of (relatively) small order that are defined over (relatively) small field extensions of the base field $\mathbf{F}_q$.

To achieve both requirements in practice we employ the implementation[8] of the KLPT algorithm in [47]. Given J_0, let $\beta \in \mathfrak{O}_0$ such that $\overline{J_0} I_0 = \mathfrak{O}_0 \beta$, i.e.

$$\beta = \widehat{\varphi_{J_0}} \circ \varphi_{I_0} \in \mathrm{End}(E_1).$$

Applying φ_{J_0} to both sides, we obtain

$$[N(J_0)] \circ \varphi_{I_0} = \varphi_{J_0} \circ \beta.$$

Since we assumed $N(J_0)$ to be coprime to M, we find that $P_2 := (\varphi_{J_0} \circ \beta)(P_1)$ generates $E_2[M, \sigma]$, as desired. Once P_2 is found, we can compute $E_3 \cong E_2/\langle P_2 \rangle$ and repeat the process with $I = (M^{r-2}, \hat{\sigma})$ to obtain a generator P_3 for $E_3[M, \sigma]$, and so on. Finally, we can push the point $Q_1 \in E_1[M, \hat{\sigma}]$ through the resulting chain of isogenies to obtain generators $Q_j \in E_j[M, \hat{\sigma}]$ for the dual isogenies. We have obtained a full kernel representation $\{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$ of the $\mathcal{O}$-orientation on E.

All algorithms described in this section are implemented in the SageMath [81] package `orientation_tools.py`, which can be found in the GitHub repository [1].

5 The Key Exchange Protocol

Now that we have a method to instantiate full kernel representations of primitive orientations on a base curve, we can employ Algorithm 1 to obtain a non-interactive key exchange protocol, as follows. Let us abbreviate the full kernel representation of an $\mathcal{O}$-orientation on $E : y^2 = x^3 + x$ as found in Sect. 4 by

[8] https://github.com/friends-of-quaternions/deuring.

$\mathcal{E} := \{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$; it is part of the public parameters of the scheme. We can view Algorithm 1 as defining an action by $\mathcal{O}$-ideals of the form $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{s_i}$, where $0 \leq s_i \leq e_i$, on the set of full kernel representations of $\mathcal{O}$-orientations. Iterating this action B times, we can evaluate the action by any ideal class with exponent vector $(s_1, \ldots, s_n)$ for which $0 \leq s_i \leq Be_i$. This gives $\prod_{i=1}^{n}(1 + Be_i)$ possible options for the s_i, which will be the size of the key space (the bound B is to be determined by the security parameter). A high-level overview of the protocol is now as follows.

(i) Alice's secret key is an exponent vector $\mathsf{sk}_A = (a_1, \ldots, a_n) \in \mathbf{Z}_{\geq 0}^n$, where $0 \leq a_i \leq Be_i$. She computes $\mathcal{E}_A = \prod_{i=1}^{n} \mathfrak{l}_i^{a_i} * \mathcal{E}$ using B applications of Algorithm 1, and sends it to Bob.
(ii) Bob's secret key is an exponent vector $\mathsf{sk}_B = (b_1, \ldots, b_n) \in \mathbf{Z}_{\geq 0}^n$, where $0 \leq b_i \leq Be_i$. He computes $\mathcal{E}_B = \prod_{i=1}^{n} \mathfrak{l}_i^{b_i} * \mathcal{E}$ using B applications of Algorithm 1, and sends it to Alice.
(iii) Alice computes $\mathcal{E}_{AB} = \prod_{i=1}^{n} \mathfrak{l}_i^{a_i} * \mathcal{E}_B$.
(iv) Bob computes $\mathcal{E}_{BA} = \prod_{i=1}^{n} \mathfrak{l}_i^{b_i} * \mathcal{E}_A$.

It is not obvious that the full kernel representations $\mathcal{E}_{AB}$ and $\mathcal{E}_{BA}$ are equal, and this is in fact not necessarily true. We did observe that this holds in practice when Alice and Bob both use Vélu's formulae for the isogeny computations; a partial explanation for this phenomenon can be found in [60, Thm. 3.1]. Regardless, it is not difficult for Alice and Bob to deduce a shared secret from the protocol, a most obvious choice being the j-invariant of, say, the first curve in the kernel representation. We will denote this by $j(\mathcal{E}_{AB}) = j(\mathcal{E}_{BA})$.

5.1 Public Keys

The key exchange protocol as described still has relatively large public keys. Indeed, if a public key is given as a full kernel representation, it consists of r elliptic curves (say, Montgomery invariants) and $2r$ elliptic curve points. This overdetermines the necessary information of an $\mathcal{O}$-orientation; indeed, it is not difficult to see that we can discard at least half of the points without losing the information of the orientation.

Public Key Compression. We are going to describe how to compress a full kernel representation $\mathcal{E} = \{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$ of an orientation. Recall that the endomorphism $\iota(\sigma)$ it encodes corresponds to the chain of isogenies

$$E_1 \to E_1/\langle P_1 \rangle \cong E_2 \to \ldots \to E_r/\langle P_r \rangle \cong E_1.$$

We first discard the points Q_j, as they can be recomputed as generators of the kernels of the duals of the isogenies. Note also that we do not require the exact points P_j, but only the (cyclic) subgroups of $E_j[M]$ that they generate. To express an order-M cyclic subgroup of the M-torsion requires at least $\tilde{O}(\log M)$

bits of information. Indeed, given the prime factorization (6) of M, the number of such subgroups is $M \cdot \prod_{i=1}^{n} \left(1 + \frac{1}{\ell_i}\right)$. We can express (a multiple of) P_j in terms of a deterministically sampled basis S_j, T_j for $E_j[M]$ by an element $\lambda_j \in \mathbf{P}^1(\mathbf{Z}/M\mathbf{Z})$. Lastly, we remark that we can discard the curves E_j for $j > 1$, since they can be inductively recomputed from the pair $(E_{j-1}, \langle P_{j-1} \rangle)$. The compressed public key consists of the data

$$\mathsf{pk} = j(E_1), (\lambda_1, \ldots, \lambda_r),$$

where $j(E_1)$ is of size $2\log_2(p)$ and λ_i is of size $\approx \log_2(M)$. In case the M-torsion spans $E(\mathbf{F}_q)$, i.e. $M = p + 1$, we thus obtain a public key of (bit)size $\approx (r+2)\log_2(p)$. In practice, there is a small overhead arising from the factor $\prod_{i=1}^{n} \left(1 + \frac{1}{\ell_i}\right)$, e.g. the actual (bit)size of public keys in the situation of Example 4.1 is $2878 > 2864.5 \approx (r+2)\log_2(p)$.

Public Key Decompression. To decompress public keys, start by sampling a deterministic basis for $E_1[M]$ and computing P_1 from λ_1. Since we have sampled a torsion basis for $E_1[M]$, we obtain for free a point $R_1 \in E_1[M]$ which is linearly independent of P_1. Next, we compute $E_1 \xrightarrow{\varphi_1} E_2 \cong E_1/\langle P_1 \rangle$, and evaluate φ_1 at R_1; the resulting point $Q_2 = \varphi_1(R_2)$ generates the kernel of the dual isogeny $\widehat{\varphi_1}$. Given E_2, we recover P_2 from λ_2, and repeat the procedure. In each step, we also compute $Q_{j+1} := \varphi_j(Q_j) \in E_{j+1}[M, \hat{\sigma}]$. Finally, we obtain the isogeny $E_r \xrightarrow{\varphi_r} E_1 \cong E_r/\langle P_r \rangle$, and evaluating it at Q_r gives $Q_1 \in E[M, \hat{\sigma}]$. We have recovered a full kernel representation $\{(E_j, P_j, Q_j) \mid 1 \leq j \leq r\}$ of the $\mathcal{O}$-orientation. The cost of this procedure is essentially (dominated by): r computations of a deterministic M-torsion basis, and one evaluation of the endomorphism $\iota(\sigma)$.

5.2 Security Analysis

In a Nutshell. For an accurate security analysis we require to consider at least the following parameters.

(i) The class number $\#\,\mathrm{Cl}(\mathcal{O})$, which is roughly proportional to $|\,\mathrm{Disc}(\mathcal{O})|^{1/2}$.
(ii) The number of distinct supersingular elliptic curves over the base field $\mathbf{F}_q = \mathbf{F}_{p^2}$, which depends on the size of p.
(iii) The size of the public key space, which (under some heuristic assumptions; see below) is roughly equal to the size of the private key space $\mathcal{K}$.

The most important difference with other protocols based on class group actions on oriented elliptic curves, such as CRS [39,75], CSIDH [22], and SCALLOP [3,27,41], is that we are allowed to choose the size of the discriminant and the size of the base field independently. In the schemes mentioned above, the quantum security is essentially dictated by Kuberberg's algorithm, whose complexity is subexponential in the class number. However, when the characteristic of the base field is small, we also need to take generic path finding algorithms between supersingular elliptic curves into account. The resulting parameter estimates for NIST level 1 are summarized in Table 1.

Table 1. Parameter estimates required to achieve NIST level 1 based on the best known generic attacks, cf. [12, Table 1].

Parameter	Attack	Quantum	NIST level 1 bitsize
$\lvert \operatorname{Disc} \mathcal{O} \rvert$	Kuperberg [54,55,72]	Yes	2048 (aggressive) 4096 (conservative)
p	BJS [9]	Yes	230 (aggressive) 256 (conservative)
$\#\mathcal{K}$	vOW [69] MITM [26]	No	221 (aggressive) 256 (conservative)

Private vs Public Keys. The private key space is the set of exponent vectors

$$\mathcal{K} := \mathcal{K}_{\text{priv}} = \{(s_1, \ldots, s_n) \mid 0 \leq s_i \leq Be_i\}.$$

One such exponent vector corresponds to the ideal class $[\prod_{i=1}^{n} \mathfrak{l}_i^{s_i}] \in \operatorname{Cl}(\mathcal{O})$. Similar to CSIDH, we require the heuristic assumption that there are few collisions in this correspondence; i.e. that the private key space maps roughly injectively into the class group. This can be motivated by combining the Cohen-Lenstra heuristics [32] with the Gaussian heuristic for the relationship lattice spanned by ideal classes $[\mathfrak{l}_i]$, see [22, Sec. 7.1] for a more in-depth discussion on this issue. We furthermore remark that the correspondence is (under a mild condition) provably injective, apart from the trivial collision $(0, \ldots, 0) \equiv (e_1, \ldots, e_n)$, if $B = 1$; see [49, Prop. 3.1]. Denoting by $\mathcal{S} \subseteq \operatorname{Cl}(\mathcal{O})$ the image of the private key space in $\operatorname{Cl}(\mathcal{O})$, the public key space is the set

$$\mathcal{K}_{\text{pub}} = \{g * \mathcal{E} \mid g \in \mathcal{S}\},$$

where $\mathcal{E}$ represents a (primitively $\mathcal{O}$-oriented) base curve. Since the class group action is free, the association $g \mapsto g * \mathcal{E}$ is injective. However, it is not necessarily injective on the level of j-invariants. Indeed, if $\#\operatorname{Cl}(\mathcal{O}) > p$, the full orbit of a primitively $\mathcal{O}$-oriented supersingular elliptic curve necessarily contains curves with the same j-invariant (but endowed with a different orientation). In particular, when p is small, we must take into account generic supersingular path-finding algorithms. Indeed, it was shown in [83] that connecting two $\mathcal{O}$-oriented elliptic curves by an ideal class is equivalent to finding any (not necessarily oriented) isogeny between them.

Remark 5.1. For the purpose of security analysis, a (full) kernel representation of an $\mathcal{O}$-orientation is equivalent to an *efficient* representation of an $\mathcal{O}$-orientation in the sense of [83, p. 7-8]. Indeed, a kernel representation is a specific instance of an efficient representation, and given an efficient representation of an $\mathcal{O}$-orientation, a kernel representation can be computed in polynomial time.[9]

[9] Assuming that M is smooth.

Heuristically, we expect that the class group action covers the supersingular j-invariants roughly uniformly.

Conjecture 5.2. Let p be a prime number and let $j \in \mathbf{F}_{p^2}$ be a supersingular j-invariant different from $0, 1728$. Let D denote a fundamental discriminant such that D does not split modulo p, and denote by $\mathcal{O}$ the associated maximal order. Then

$$\frac{p}{12} \lim_{D \to \infty} \frac{\#\{(E, \iota) \mid j(E) = j, \iota \text{ is an } \mathcal{O}\text{-orientation}\}/\cong}{\#\operatorname{Cl}(\mathcal{O})} = 1.$$

This is roughly motivated by equidistribution results for CM points [45,65]. A weaker, but provable, result is that if $|\operatorname{Disc}(\mathcal{O})|$ is sufficiently large then every supersingular elliptic curve is $\mathcal{O}$-orientable; see [46, Thm. 1.2] for a precise (but ineffective) statement, and [61] for a weaker (but effective) statement.

We summarize the above discussion into the following (inexact) formulation of our main heuristic assumption.

Heuristic 5.3. *If $\#\mathcal{K}_{\text{priv}}$ and $\operatorname{Disc}(\mathcal{O})$ are sufficiently large, then acting by (the image in $\operatorname{Cl}(\mathcal{O})$ of) uniformly random elements of $\mathcal{K}_{\text{priv}}$ on a fixed primitively $\mathcal{O}$-oriented supersingular base curve yields ($\mathcal{O}$-oriented curves with) statistically uniform supersingular j-invariants.*

We now provide some experimental evidence toward this hypothesis. Note that for parameters of cryptographic size, we expect collisions between random private key pairs to be unlikely, even when the heuristic is false. In particular, we expect to be able to falsify the heuristic only for smaller values of p. We will restrict our experimental parameters to the case where $\#\mathcal{K}_{\text{priv}} \approx p$, which is in line with what is dictated by generic attacks according to our security analysis below.

In the first set of experiments, we use $p = 67339 \approx 2^{16}$, for which $p + 1 = 2^2 \cdot 5 \cdot 7 \cdot 13 \cdot 37$ and $p - 1 = 2 \cdot 3^3 \cdot 29 \cdot 43$; see Sect. 6.2 for why such a choice of prime is natural. We take $M = 3^3 \cdot 5 \cdot 7 \cdot 13 \cdot 29 \cdot 37 \cdot 43$ to be the odd part of $(p+1)(p-1)$. Next, we choose the smallest possible value of B for which the private key space is of size $\geq 2^{16}$; this is $B = 4$, for which $\#\mathcal{K}_{\text{priv}} = 203125$. For various values of $r \in \mathbf{Z}_{>0}$, we compute orientations on the base curve $(E_0 : y^2 = x^3 + x)$ corresponding to an endomorphism of degree M^r (and negligibly small trace). We then act by the first 5612 elements (this is the total number of supersingular j-invariants in $\mathbf{F}_{p^2}$) of $\#\mathcal{K}_{\text{priv}}$, ordered lexicographically, and record the total number of distinct j-invariants encountered. Comparing the results to the uniform distribution, we report on the one-sided p-value, i.e. the probability that we obtain at most the observed amount of distinct j-invariants assuming that the distribution is uniform, in Table 2. The difference with the uniform distribution is statistically significant for $r = 1$, which corresponds to a discrimimant of size $|\operatorname{Disc}(\mathcal{O})| \approx 2^{31}$; this is as expected, since this is likely too small to cover all supersingular j-invariants, let alone uniformly. Interestingly, for $r = 3$ we obtain significantly *more* distinct j-invariants than what would be expected from a uniform distribution. A possible explanation for this is that curves that are connected by small degree isogenies are typically not isomorphic

(since few curves have small degree endomorphisms), hence there is a bias for elements of $\mathcal{K}_{\text{priv}}$ that are close, say in ℓ^1-norm, to correspond to distinct j-invariants.

Table 2. Experimental results for $p \approx 2^{16}$.

r	1	2	3	4	8	16
# distinct j	3475	3554	3630	3547	3553	3576
p-value	0.0010	0.6154	0.9998	0.4975	0.5990	0.8917

In the second set of experiments, we use $p = 8{\cdot}3{\cdot}5{\cdot}7{\cdot}17{\cdot}19{\cdot}23{\cdot}29{\cdot}31-1 \approx 2^{32}$. We take M to be the odd part of $p+1$. Using $B = 16$, we obtain a key space of size $\approx 2^{32.7}$. This time, we act by 2^{16} uniformly random elements of $\mathcal{K}_{\text{priv}}$, and store the 8 most significant bits of the MD5 hash of the resulting j-invariant. We then compare the resulting distribution with the uniform one using a χ^2 distribution test. None of the obtained p-values indicates a statistically significant deviation (Table 3).

Table 3. Experimental results for $p \approx 2^{32}$.

r	2	3	4	8	16
χ^2 statistic	231.93	231.61	233.84	259.84	246.75
p-value	0.847	0.851	0.825	0.404	0.633

Classical Security. The most important classical attacks are (depth-first) meet-in-the-middle (MITM) [26] and van Oorschot-Wiener golden collision search (vOW) [69]. Given access to $O(\#\mathcal{K}^{1/4})$ memory, the first has an asymptotic time complexity of $O(\#\mathcal{K}^{1/4})$. vOW has a worse time complexity, but performs better when memory is limited. It was argued in [26, Sec. 5.2] and [2] that given realistic constraints on the amount of available memory, vOW is the superior algorithm for a classical attack. Using the estimates of [26], we require bitsizes of $\log_2(\#\mathcal{K}) = 221, 234, 332$ for NIST levels $1, 2, 3$ respectively. In [12], the more *conservative* choices $\log_2(\#\mathcal{K}) = 256, 256, 348$ (for the same respective NIST levels) are also considered. Note that classical generic supersingular isogeny path finding has complexity $O(p^{1/2})$, which loses to vOW in case $\#\mathcal{K} \leq p$.

Quantum Security. Assuming the heuristic that the ideal classes $[\mathfrak{l}_i]$ span $\mathrm{Cl}(\mathcal{O})$, Kuberberg's algorithm [54,55,72] has time complexity subexponential in the class number $\#\,\mathrm{Cl}(\mathcal{O})$, where $\#\,\mathrm{Cl}(\mathcal{O})/|\,\mathrm{Disc}(\mathcal{O})|^{1/2}$ is, asymptotically on

average, equal to ≈ 0.46 [33, p. 290]. Although estimates of the actual complexity of Kuperberg's algorithm for concrete parameters choices—most prominently for CSIDH—are still subject of debate [8,10,26,70], the most recent papers on the topic employ the relatively pessimistic estimates [12,13] of 2048- and 4096-bit discriminants for "aggressive" and "conservative" NIST level 1 respectively.

Like CSIDH and B-SIDH [37], the degrees of the secret isogenies are large compared to p, so Tani's algorithm [52,80] does not give an advantage compared to the best generic path finding algorithm BJS [9]. The latter has a time complexity of $O(p^{1/4})$, which suggests primes of bitsize ≥ 256 for NIST level 1. However, the more aggressive value $\log_2(p) = 230$ has also been proposed [37, Sec. 5.2].

Non-generic Attacks. In case the imaginary quadratic order $\mathcal{O}$ is not maximal, the class group can be reduced to that of a larger order if an ascending isogeny can be computed. This applies most notably to SCALLOP (and variants), where it is argued that computing such an ascending isogeny is infeasible, see e.g. [41, Sec. 7], because it has large prime degree. In OSIDH [34], the ascending isogeny is of smooth degree, which means that access to an efficient representation of the orientation renders the protocol insecure [67]; it was shown that this leads to a practical (though exponential time) attack [40]. In practice, we choose to orient only by maximal orders (we use discriminants that are squarefree), so that this attack does not apply.

In case the exponent of the class group is exceptionally small, there are quantum attacks that outperform Kuperberg's [23,31,56]. Heuristically [32], however, we expect the class group to be close to cyclic, so that these attacks do not apply.

In some cases, primes that ramify in the class group lead to attacks based on self-pairings [19], most notably for the decisional Diffie–Hellman problem [20, 24]. The complexity of these attacks is polynomial in the size of the ramifying prime, hence infeasible for discriminants that contain no small prime divisors. In practical instances, we avoid these attacks by using a discriminant that is either prime or the product of two large primes, cf. Sect. 4.1.

By construction, the kernel representation of the orientation on the public key curve contains images of smooth torsion points under a secret isogeny. In public key compression, this information is automatically masked, because it is reduced to the subgroup that the point generates (in fact, this subgroup is an eigenspace for the action of $\iota(\sigma)$ on the M-torsion, which could be recovered from any efficient representation of the orientation). However, torsion point attacks based on higher-dimensional isogeny interpolation [16,62,73] do not seem to apply, even if the kernel representation is sent in uncompressed form, because the degree of the secret isogeny is unknown (in fact the degree *is* the secret, because it is in one-to-one correspondence with the exponent vector).

6 Practical Instances

6.1 A First Approach

Since the main idea of representing an orientation as a chain of isogenies is that we can increase the size of the discriminant without increasing the size of the base field (and so to mitigate subexponential quantum attacks), it is tempting to consider a base field of minimal characteristic p, say e.g. 256 bits for "conservative" NIST level 1. For example

$$p = 2^5 \cdot 3^3 \cdot \underbrace{(5 \cdot 7 \cdot 11 \cdot \ldots \cdot 193)}_{\text{skip } 173} - 1 \approx 2^{256.5}. \tag{8}$$

We then take $M = p+1$, so that $E(\mathbf{F}_q) = E[M]$ for the base curve $E : y^2 = x^3+x$. Next, we can use the method of Sect. 4 to generate a primitive orientation by an order $\mathcal{O} = \mathbf{Z}[\sigma]$ where $N(\sigma) = M^r$ and $\mathrm{tr}(\sigma)$ is small, where we require $r = 16$ to attain a 4096-bit discriminant. Finally, we determine the size of the exponent vector required to attain a key space of 256 bits. Note that the number of distinct prime divisors $\ell_i \mid M$ is $n = 43$. The smallest positive integer $B \in \mathbf{Z}_{>0}$ for which

$$2^{256} < \#\mathcal{K} = \prod_{i=1}^{n}(1 + e_i B) = (1 + 5B)(1 + 3B)(1 + B)^{41}$$

is $B = 58$. Recall that B is the number of iterations of Algorithm 1 in the evaluation of the class group action to attain the desired key space. Although all isogeny computations are over a small base field and of small degree ≤ 193, this is likely suboptimal. The issue here is that the size of the key space is exponential in n but only polynomial in B; this suggests a trade-off between minimizing the size of p and maximizing the number of prime divisors $\ell_i \mid (p + 1)$.

6.2 The Twist Trick

A well-known trick [6,36,43,76,77] when lacking torsion, is to consider points on the quadratic twist E^t of E. It has opposite trace of Frobenius, so that $\#E^t(\mathbf{F}_q) = (p - 1)^2$. The $\mathbf{F}_q$-rational points on the twist correspond to points on $E(\mathbf{F}_q)$ with x-coordinate in $\mathbf{F}_q$ (but y-coordinate in $\mathbf{F}_{q^2} = \mathbf{F}_{p^4}$). When using x-only arithmetic, such points can be handled purely using $\mathbf{F}_q$-arithmetic. In our situation, such points may thus be part of the kernel representation of the orientation with virtually no loss of efficiency.[10] In effect, this means that we can add prime divisors of $p - 1$ to the set of ℓ_i used in the class group action computation. We remark that the analogue of this trick in the case that the base field is $\mathbf{F}_p$ is also used in CSIDH, the difference being that in the CSIDH setting

[10] One technicality is that points on the twist must be individually pushed through every isogeny, increasing the minimum number of isogeny evaluations in the group action computation. In practice, this restricts the space of possible *strategies*; cf. Sect. 3.4.

the group of $\mathbf{F}_p$-rational points on the twist has the same order as that of the curve itself.

Finding *twin-smooth* primes, i.e. prime numbers p for which $p+1$ and $p-1$ are both smooth, is a non-trivial number-theoretic problem [35,79], which has recently gained attention in isogeny-based cryptography [11,38,77,78] for its applications in B-SIDH [36], (one-dimensional) SQISign [42,43,76], and POKE [6]. For the primes that we require, the number $p^2-1 = (p+1)(p-1)$ need not necessarily be fully smooth (i.e. a possibly large prime divisor is allowed), since we are free to use any $M \mid (p^2-1)$. However, contrary to the applications mentioned above, we require *many* distinct prime divisors $\ell_i \mid (p^2-1)$. An example, which was communicated to us by Bruno Sterner, is the 256-bit prime

$$p = \texttt{0xaee61f0e7eb0b904142a14b1ba552ca011c6104b007e448fe6aed1f1af734f1f},$$

where the prime factorizations of $p \pm 1$ are

$$\begin{aligned} p+1 &= 2^5 \cdot 7^2 \cdot 11 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 53 \cdot 61 \cdot 67 \cdot 73 \cdot 79 \cdot 83 \\ &\quad \cdot 103 \cdot 107 \cdot 131 \cdot 137 \cdot 149 \cdot 173 \cdot 199 \cdot 211 \cdot 277 \cdot 307 \\ &\quad \cdot 5370594787 \cdot 10398664516670979076559; \\ p-1 &= 2 \cdot 3^2 \cdot 5 \cdot 13 \cdot 17 \cdot 19 \cdot 31 \cdot 47 \cdot 59 \cdot 71 \cdot 89 \cdot 97 \cdot 101 \cdot 109 \cdot 113 \cdot 127 \\ &\quad \cdot 139 \cdot 151 \cdot 157 \cdot 163 \cdot 167 \cdot 179 \cdot 181 \cdot 191 \cdot 193 \cdot 223 \cdot 233 \cdot 269 \\ &\quad \cdot 167936514812729522270554 81. \end{aligned}$$

Finding primes that are "optimal" for evaluating the class group action depends on an optimized implementation of Algorithm 1. Both require a significant amount of non-trivial effort and are left for future work.

6.3 An Example

Using the prime number p above, we define M to be the product of all small (i.e. ≤ 307) *odd*[11] prime power divisors $\ell_i^{e_i}$ of $p \pm 1$, i.e.

$$M = 3^2 \cdot 5 \cdot 7^2 \cdot \ldots \cdot 233 \cdot 277 \cdot 307 = \prod_{i=1}^{n} \ell_i^{e_i}.$$

Using the method of Sect. 4.1, we found an orientation on $E : y^2 = x^3 + x$ by an element $\sigma \in \mathrm{End}(E)$ for which

$$N(\sigma) = M^{13} \qquad \text{and} \qquad \mathrm{tr}(\sigma) = 29171033,$$

[11] Although there is no theoretical limitation preventing us from including the power 2^5, this leads to some practical technicalities when using Kummer arithmetic for Montgomery curves; cf. the section "Future Work" in [71, `README.md`]. Since we are anyway far from an optimized implementation, we chose to exclude even-degree isogenies in the example (at the cost of a minor loss in efficiency).

such that $|\mathrm{Disc}(\sigma)| = 4M^{13} - 29171033 \approx 2^{4105}$ is prime. Next, using a minor modification of the method of Sect. 4.2 to include points on the quadratic twist, we generated a kernel representation of the orientation on the base curve. This (pre)computation was relatively expensive, and took about 200 core hours on a desktop CPU[12], with the vast majority taken up by the effective Deuring correspondence [47]. We obtain a key space of size 256 bits for $B = 33$. Using these parameters, our (unoptimized) proof-of-concept SageMath [81] implementation executes a class group action corresponding to the *conservative* NIST level 1 category (cf. Table 1) using Algorithm 1 on a single core of a laptop CPU[13] in a median 53.4 s (over 100 ideal classes, standard deviation ≈ 0.1 s). We relied on a package by Giacomo Pope for Kummer arithmetic [71]. Using the same algorithms for isogeny computations, our SageMath implementation of the class group action from dCSIDH [12] for the 4096-bit prime given in [12, Table 1] runs in a median 370.6 s (over 40 ideal classes, standard deviation ≈ 2.2 s), excluding random point sampling. It should be stressed that, since both benchmarks are based on heavily suboptimal implementations, future work needs to be done to accurately assess the practical performance difference. The SageMath code associated to our implementations can be found in the GitHub repository [1].

7 Future Work

Implementation. There are various tricks, mostly from the CSIDH literature, to speed up the computation of class group actions on oriented elliptic curves. For an optimized implementation of Algorithm 1 in a low-level programming language we should include (at least) the following methods

(i) Optimal differential addition chains [25];
(ii) Different elliptic curve models [64];
(iii) Optimal strategies [28,30,50,63,68];
(iv) Fast constant-time finite field arithmetic [12,13].

Secondly, we would like to find good choices for the parameters, most notably the characteristic of the base field p. This involves a further study of twin smooth primes and—due to the unique shape of prime—likely requires new techniques. In practice, these two goals are heavily interdependent, in the sense that the optimal parameters depend on an optimal implementation and vice versa. It would be most interesting to compare to the optimized implementations of dCSIDH [12] and dCTIDH [13] in practice. Another interesting contender is the non-interactive isogeny-based key exchange protocol $\otimes$-MIKE [74], which is based on a generalization of the class group action known as the *module action*, though, at the time of writing, no implementation is publicly available.

[12] Intel i9-9900KS @ 4.00 GHz.
[13] AMD Ryzen 7 PRO 8840U @ 3.30 GHz.

Public Key Validation. In a non-interactive key exchange with static keys, one is required to *validate* public keys to prevent active attacks. Indeed, a malicious Bob could publish a representation of a "weak" orientation (e.g. with several ℓ_i ramifying in the class group) an infer information about Alice's secret from (the success or failure of) subsequent encryptions. Note that in the public key decompression algorithm, Sect. 5.1, we already verify that the full kernel representation corresponds to an endomorphism $\tau \in \mathrm{End}(E_1)$ of degree $M^r = N(\sigma)$. Indeed, every P_j has order M by construction, and we checked that the last isogeny φ_r maps to E_1. What remains to verify is that

$$\mathrm{tr}(\tau) = \mathrm{tr}(\sigma). \tag{9}$$

Indeed, if this holds, then $\mathrm{Disc}(\tau) = \mathrm{Disc}(\sigma)$, and we can conclude that our kernel representation corresponds to a $\mathbf{Z}[\sigma]$-orientation. Moreover, since we chose $\mathrm{Disc}(\sigma)$ to be a fundamental discriminant, we are guaranteed that the orientation is primitive. There exist generic polynomial time algorithms that compute the trace of an endomorphism that is given as a sequence of isogenies of smooth degree [66], but these are likely not the best we can do for our particular situation, for at least two reasons.

(i) We assume that we have a large amount of smooth rational torsion. This gives a strong advantage compared to generic algorithms [66, Sec. 3.3].
(ii) We do not require to compute the trace, but merely have to verify that it is correct. In particular, this means that we may not require discrete logarithm computations [66, Sec. 3.3(3)] (cf. the discussion below).

As a partial result, it is not too difficult to see that the trace can be easily verified modulo M, as follows. To check the validity of (9) (mod M), we will use the point $R_1 \in E_1[M]$ that we computed during public key decompression. Let us write $R_1 = \lambda P_1 + \mu Q_1$, where $\lambda, \mu \in (\mathbf{Z}/M\mathbf{Z})$ are unknown to us (for now). By construction, we have

$$Q_1 = \tau(R_1) = [\mu] \circ \tau(Q_1) = [\mu] \circ (\tau + \hat{\tau})(Q_1) = [\mu \cdot \mathrm{tr}(\tau)]Q_1.$$

Denoting by $e_M : E[M] \times E[M] \to \mu_M$ the M-Weil pairing, it follows that

$$e_M(P_1, Q_1) = e_M(P_1, R_1)^{\mathrm{tr}(\tau)}.$$

Using two Weil-pairing computations and one exponentiation in $\mathbf{F}_q$, we can thus verify that $\mathrm{tr}(\tau) \equiv \mathrm{tr}(\sigma) \pmod{M}$.[14] This is not sufficient, because we need to determine the trace modulo at least $4M^{r/2}$ due to the Hasse–Weil bound. We remark that knowing $\mathrm{tr}(\tau) \pmod{M}$ does give a head start when using the generic trace computation [66], since the latter works by computing the trace modulo sufficiently many small integers and lifting via the Chinese Remainder Theorem (CRT). We leave determining the best algorithm to validate public keys as a topic for future research, the main question being whether we can verify the trace, say modulo $M^{r/2}$, without resorting to generic techniques.

[14] Note that, if we were to compute the trace from scratch, we would require a discrete logarithm computation to obtain $\mathrm{tr}(\tau) \pmod{M}$.

Acknowledgements. We thank Damien Robert for helpful discussions, and Krijn Reijnders for feedback on an earlier version of the paper. Special thanks to Bruno Sterner for finding the prime number used in the example of Sect. 6.3. This work was financially supported by the France 2030 program, managed by the French National Research Agency under grant agreement No. ANR-22-PETQ-0008 PQ-TLS.

References

1. GitHub repository associated to this paper. https://github.com/houbenmr/LargeDiscriminantOrientations
2. Adj, G., Cervantes-Vázquez, D., Chi-Domínguez, J.J., Menezes, A., Rodríguez-Henríquez, F.: On the cost of computing isogenies between supersingular elliptic curves. In: Cid, C., Jacobson, M.J., Jr. (eds.) SAC 2018. LNCS, pp. 322–343. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10970-7_15
3. Allombert, B., et al.: PEARL-SCALLOP: parameter extension applicable in real-life SCALLOP. Cryptology ePrint Archive, Report 2024/1744 (2024). https://eprint.iacr.org/2024/1744
4. Banegas, G., et al.: CTIDH: faster constant-time CSIDH. IACR TCHES **2021**(4), 351–387 (2021). https://doi.org/10.46586/tches.v2021.i4.351-387. https://tches.iacr.org/index.php/TCHES/article/view/9069
5. Banegas, G., et al.: Disorientation faults in CSIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 310–342. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_11
6. Basso, A.: POKE: a framework for efficient PKEs, split KEMs, and OPRFs from higher-dimensional isogenies. Cryptology ePrint Archive, Report 2024/624 (2024). https://eprint.iacr.org/2024/624
7. Bernstein, D.J., De Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. Cryptology ePrint Archive, Report 2020/341 (2020). https://eprint.iacr.org/2020/341
8. Bernstein, D.J., Lange, T., Martindale, C., Panny, L.: Quantum Circuits for the CSIDH: optimizing quantum evaluation of isogenies. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 409–441. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_15
9. Biasse, J.-F., Jao, D., Sankar, A.: A quantum algorithm for computing isogenies between supersingular elliptic curves. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014. LNCS, vol. 8885, pp. 428–442. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13039-2_25
10. Bonnetain, X., Schrottenloher, A.: Quantum security analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 493–522. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_17
11. Bruno, G., et al.: Cryptographic smooth neighbors. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 190–221. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8739-9_7
12. Campos, F., et al.: Optimizations and practicality of high-security CSIDH. CiC **1**(1), 5 (2024). https://doi.org/10.62056/anjbksdja
13. Campos, F., Hellenbrand, A., Meyer, M., Reijnders, K.: dCTIDH: fast & deterministic CTIDH. Cryptology ePrint Archive, Report 2025/107 (2025). https://eprint.iacr.org/2025/107

14. Campos, F., Kannwischer, M.J., Meyer, M., Onuki, H., Stöttinger, M.: Trouble at the CSIDH: protecting CSIDH with dummy-operations against fault injection attacks. Cryptology ePrint Archive, Report 2020/1005 (2020). https://eprint.iacr.org/2020/1005
15. Castryck, W., Decru, T.: CSIDH on the surface. In: Ding, J., Tillich, J.-P. (eds.) PQCrypto 2020. LNCS, vol. 12100, pp. 111–129. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44223-1_7
16. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_15
17. Castryck, W., Decru, T., Houben, M., Vercauteren, F.: Horizontal racewalking using radical isogenies. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part II. LNCS, vol. 13792, pp. 67–96. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22966-4_3
18. Castryck, W., Decru, T., Vercauteren, F.: Radical isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 493–519. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_17
19. Castryck, W., Houben, M., Merz, S.P., Mula, M., van Buuren, S., Vercauteren, F.: Weak instances of class group action based cryptography via self-pairings. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part III. LNCS, vol. 14083, pp. 762–792. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38548-3_25
20. Castryck, W., Houben, M., Vercauteren, F., Wesolowski, B.: On the decisional Diffie-Hellman problem for class group actions on oriented elliptic curves. Res. Number Theory **8**(4), 99 (2022)
21. Castryck, W., Invernizzi, R., Lorenzon, G., Meers, J., Vercauteren, F.: Orient express: using frobenius to express oriented isogenies. Cryptology ePrint Archive, Paper 2025/1047 (2025). https://eprint.iacr.org/2025/1047
22. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15
23. Castryck, W., Meeren, N.V.: Two remarks on the vectorization problem. Cryptology ePrint Archive, Report 2022/1366 (2022). https://eprint.iacr.org/2022/1366
24. Castryck, W., Sotáková, J., Vercauteren, F.: Breaking the decisional Diffie-Hellman problem for class group actions using genus theory: extended version. J. Cryptol. **35**(4), 24 (2022). https://doi.org/10.1007/s00145-022-09435-1
25. Cervantes-Vázquez, D., Chenu, M., Chi-Domínguez, J.-J., De Feo, L., Rodríguez-Henríquez, F., Smith, B.: Stronger and faster side-channel protections for CSIDH. In: Schwabe, P., Thériault, N. (eds.) LATINCRYPT 2019. LNCS, vol. 11774, pp. 173–193. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30530-7_9
26. Chávez-Saab, J., Chi-Domínguez, J.J., Jaques, S., Rodríguez-Henríquez, F.: The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. J. Cryptogr. Eng. **12**(3), 349–368 (2022). https://doi.org/10.1007/s13389-021-00271-w
27. Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: group action from 2-dimensional isogenies. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part II. LNCS, vol. 14603, pp. 190–216. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-57725-3_7
28. Cheng, H., Fotiadis, G., Großschädl, J., Ryan, P.Y.A., Rønne, P.B.: Batching CSIDH group actions using AVX-512. IACR TCHES **2021**(4), 618–649

(2021). https://doi.org/10.46586/tches.v2021.i4.618-649. https://tches.iacr.org/index.php/TCHES/article/view/9077
29. Chi-Domínguez, J.-J., Reijnders, K.: Fully projective radical isogenies in constant-time. In: Galbraith, S.D. (ed.) CT-RSA 2022. LNCS, vol. 13161, pp. 73–95. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95312-6_4
30. Chi-Domínguez, J.J., Rodríguez-Henríquez, F.: Optimal strategies for CSIDH. Cryptology ePrint Archive, Report 2020/417 (2020). https://eprint.iacr.org/2020/417
31. Childs, A.M., Van Dam, W.: Quantum algorithms for algebraic problems. Rev. Mod. Phys. **82**(1), 1–52 (2010)
32. Cohen, H., Lenstra, H.W.: Heuristics on class groups of number fields. In: Jager, H. (ed.) Number Theory Noordwijkerhout 1983, pp. 33–62. Springer, Heidelberg (1984). https://doi.org/10.1007/BFb0099440
33. Cohen, H.: A Course in Computational Algebraic Number Theory. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-662-02945-9
34. Colò, L., Kohel, D.: Orienting supersingular isogeny graphs. J. Math. Cryptol. **14**, 414–437 (2020). https://doi.org/10.1515/jmc-2019-0034
35. Conrey, J.B., Holmstrom, M.A., McLaughlin, T.L.: Smooth neighbors. Exp. Math. **22**(2), 195–202 (2013)
36. Costello, C.: Computing supersingular isogenies on Kummer surfaces. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 428–456. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_16
37. Costello, C.: B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 440–463. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_15
38. Costello, C., Meyer, M., Naehrig, M.: Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 272–301. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_10
39. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006). https://eprint.iacr.org/2006/291
40. Dartois, P., De Feo, L.: On the security of OSIDH. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part I. LNCS, vol. 13177, pp. 52–81. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-97121-2_3
41. De Feo, L., et al.: SCALLOP: scaling the CSI-FiSh. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 345–375. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-31368-4_13
42. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_3
43. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence: towards practical and secure sqisign signatures. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 659–690. Springer (2023)
44. Decru, T.: Radical $\sqrt[N]{\text{élu}}$ isogeny formulae. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 107–128. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68388-6_5
45. Duke, W.: Hyperbolic distribution problems and half-integral weight Maass forms. Invent. Math. **92**(1), 73–90 (1988). https://doi.org/10.1007/BF01393993

46. Elkies, N., Ono, K., Yang, T.: Reduction of CM elliptic curves and modular function congruences. Int. Math. Res. Not. **44**, 2695–2707 (2005). https://doi.org/10.1155/IMRN.2005.2695
47. Eriksen, J.K., Panny, L., Sotáková, J., Veroni, M.: Deuring for the people: supersingular elliptic curves with prescribed endomorphism ring in general characteristic. In: LuCaNT: LMFDB, Computation, and Number Theory, Contemp. Math., vol. 796, pp. 339–373 (2024). https://doi.org/10.1090/conm/796/16008
48. Gajland, P., de Kock, B., Quaresma, M., Malavolta, G., Schwabe, P.: SWOOSH: efficient lattice-based non-interactive key exchange. In: Balzarotti, D., Xu, W. (eds.) USENIX Security 2024. USENIX Association (2024)
49. Houben, M.: Deterministic algorithms for class group actions. Cryptology ePrint Archive, Report 2025/847 (2025). https://eprint.iacr.org/2025/847
50. Hutchinson, A., LeGrow, J., Koziel, B., Azarderakhsh, R.: Further optimizations of CSIDH: a systematic approach to efficient strategies, permutations, and bound vectors. In: Conti, M., Zhou, J., Casalicchio, E., Spognardi, A. (eds.) ACNS 2020. LNCS, vol. 12146, pp. 481–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57808-4_24
51. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
52. Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the RAM model: claw-finding attacks on SIKE. Cryptology ePrint Archive, Report 2019/103 (2019). https://eprint.iacr.org/2019/103
53. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ-isogeny path problem. LMS J. Comput. Math. **17**, 418–432 (2014). https://doi.org/10.1112/S1461157014000151
54. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM J. Comput. **35**(1), 170–188 (2005)
55. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In: 8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013). Leibniz International Proceedings in Informatics (LIPIcs), vol. 22, pp. 20–34. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2013). https://doi.org/10.4230/LIPIcs.TQC.2013.20. http://drops.dagstuhl.de/opus/volltexte/2013/4321
56. Leem, S., Jacobson, M.J., Jr., Scheidler, R.: Solving norm equations in global function fields. Res. Number Theory **11**(1), 17 (2025)
57. LeGrow, J., Hutchinson, A.: An analysis of fault attacks on CSIDH. Cryptology ePrint Archive, Report 2020/1006 (2020). https://eprint.iacr.org/2020/1006
58. LeGrow, J.T.: A faster method for fault attack resistance in static/ephemeral CSIDH. J. Cryptogr. Eng. **13**(3), 283–294 (2023). https://doi.org/10.1007/s13389-023-00318-0
59. LeGrow, J.T., Hutchinson, A.: (Short Paper) analysis of a strong fault attack on static/ephemeral CSIDH. In: Nakanishi, T., Nojima, R. (eds.) IWSEC 2021. LNCS, vol. 12835, pp. 216–226. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85987-9_12
60. Leonardi, C.: A note on the ending elliptic curve in SIDH. Cryptology ePrint Archive, Report 2020/262 (2020). https://eprint.iacr.org/2020/262
61. Leroux, A.: An effective lower bound on the number of orientable supersingular elliptic curves. In: Smith, B., Wu, H. (eds.) SAC 2022. LNCS, vol. 13742, pp. 263–281. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58411-4_12

62. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 448–471. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_16
63. Meyer, M., Campos, F., Reith, S.: On lions and elligators: an efficient constant-time implementation of CSIDH. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, pp. 307–325. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_17
64. Meyer, M., Reith, S.: A faster way to the CSIDH. In: Chakraborty, D., Iwata, T. (eds.) INDOCRYPT 2018. LNCS, vol. 11356, pp. 137–152. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05378-9_8
65. Michel, P.: The subconvexity problem for Rankin-Selberg L-functions and equidistribution of Heegner points. Ann. of Math. (2) **160**(1), 185–236 (2004). https://doi.org/10.4007/annals.2004.160.185
66. Morrison, T., Panny, L., Sotáková, J., Wills, M.: The sea algorithm for endomorphisms of supersingular elliptic curves (2025). https://arxiv.org/abs/2501.16321
67. Onuki, H.: On oriented supersingular elliptic curves. Finite Fields and Their Applications (2021). https://doi.org/10.1016/j.ffa.2020.101777
68. Onuki, H., Aikawa, Y., Yamazaki, T., Takagi, T.: (Short Paper) a faster constant-time algorithm of CSIDH keeping two points. In: Attrapadung, N., Yagi, T. (eds.) IWSEC 2019. LNCS, vol. 11689, pp. 23–33. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26834-3_2
69. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. J. Cryptol. **12**(1), 1–28 (1999). https://doi.org/10.1007/PL00003816
70. Peikert, C.: He Gives C-Sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 463–492. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_16
71. Pope, G.: An implementation of isogenies between Kummer Lines of Montgomery curves using x-only arithmetic. https://github.com/GiacomoPope/KummerIsogeny
72. Regev, O.: A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space (2004). https://arxiv.org/pdf/quant-ph/0406151
73. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_17
74. Robert, D.: The module action for isogeny based cryptography. Cryptology ePrint Archive, Paper 2024/1556 (2024). https://eprint.iacr.org/2024/1556
75. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145 (2006). https://eprint.iacr.org/2006/145
76. Santos, M.C.R., Eriksen, J.K., Meyer, M., Reijnders, K.: AprèsSQI: extra fast verification for SQIsign using extension-field signing. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 63–93. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58716-0_3
77. Santos, M.C.R., Eriksen, J.K., Meyer, M., Rodríguez-Henríquez, F.: Finding practical parameters for isogeny-based cryptography. CiC **1**(3), 39 (2024). https://doi.org/10.62056/ayojbhey6b
78. Sterner, B.: Towards optimally small smoothness bounds for cryptographic-sized twin smooth integers and their isogeny-based applications. Cryptology ePrint Archive, Report 2023/1576 (2023). https://eprint.iacr.org/2023/1576
79. Størmer, C.: Quelques théorèmes sur l'équation de pell x2- dy2=±1 et leurs applications. Christiania Videnskabens Selskabs Skrifter, Math. Nat. Kl(2) **48** (1897)

80. Tani, S.: Claw finding algorithms using quantum walk. Theoret. Comput. Sci. **410**(50), 5285–5297 (2009)
81. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.6) (2025). https://www.sagemath.org
82. Vélu, J.: Isogénies entre courbes elliptiques. Comptes-Rendus de l'Académie des Sciences **273**, 238–241 (1971)
83. Wesolowski, B.: Orientations and the supersingular endomorphism ring problem. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 345–371. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07082-2_13