A note on the security of CSIDH

Jean-François Biasse[1], Annamaria Iezzi[1], and Michael J. Jacobson, Jr.[2]

[1] Department of Mathematics and Statistics
University of South Florida
{biasse,aiezzi}@usf.edu
[2] Department of Computer Science
University of Calgary
jacobs@ucalgary.ca

Abstract. We propose a quantum algorithm for computing an isogeny between two elliptic curves E_1, E_2 defined over a finite field such that there is an imaginary quadratic order $\mathcal{O}$ satisfying $\mathcal{O} \simeq \mathrm{End}(E_i)$ for $i = 1, 2$. This concerns ordinary curves and supersingular curves defined over $\mathbb{F}_p$ (the latter used in the recent CSIDH proposal). Our algorithm has heuristic asymptotic run time $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ and requires polynomial quantum memory and $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ quantumly accessible classical memory, where Δ is the discriminant of $\mathcal{O}$. This asymptotic complexity outperforms all other available methods for computing isogenies.
We also show that a variant of our method has asymptotic run time $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ while requesting only polynomial memory (both quantum and classical).

1 Introduction

Given two elliptic curves E_1, E_2 defined over a finite field $\mathbb{F}_q$, the isogeny problem consists in computing an isogeny $\phi : E_1 \to E_2$, i.e. a non-constant morphism that maps the identity point on E_1 to the identity point on E_2. There are two different types of elliptic curves: ordinary and supersingular. The latter have very particular properties that impact the resolution of the isogeny problem. The first instance of a cryptosystem based on the hardness of computing isogenies was due to Couveignes [13], and its concept was independently rediscovered by Stolbunov [34]. Both proposals used ordinary curves.

Childs, Jao and Soukharev observed in [11] that the problem of finding an isogeny between two ordinary curves E_1 and E_2 defined over $\mathbb{F}_q$ and having the same endomorphism ring could be reduced to the problem of solving the Hidden Subgroup Problem (HSP) for a generalized dihedral group. More specifically,

Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf. This work was supported by the U.S. National Science Foundation under grant 1839805, by NIST under grant 60NANB17D184, and by the Simons Foundation under grant 430128.

let $K = \mathbb{Q}(\sqrt{t^2 - 4q})$ where t is the trace of the Frobenius endomorphism of the curves, and let $\mathcal{O} \subseteq K$ be the quadratic order isomorphic to the ring of endomorphisms of E_1 and E_2. Let $\mathrm{Cl}(\mathcal{O})$ be the ideal class group of $\mathcal{O}$. Classes of ideals act on isomorphism classes of curves with endomorphism ring isomorphic to $\mathcal{O}$. The problem of finding an isogeny between E_1 and E_2 can be then reduced to the problem of finding a "nicely" represented ideal $\mathfrak{a} \subseteq \mathcal{O}$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$ where $*$ is the action of $\mathrm{Cl}(\mathcal{O})$, $[\mathfrak{a}]$ is the class of $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$ and $\overline{E}_i$ is the isomorphism class of the curve E_i. Childs, Jao and Soukharev showed that this could be done by solving the HSP for $\mathbb{Z}_2 \ltimes \mathrm{Cl}(\mathcal{O})$. Let $N := |\mathrm{Cl}(\mathcal{O})| \sim \sqrt{|t^2 - 4q|}$. Using Kuperberg's sieve [27], this task requires $2^{O\left(\sqrt{\log(N)}\right)}$ queries to an oracle that computes the action of the class of an element in $\mathrm{Cl}(\mathcal{O})$. Childs et al. used a method with complexity in $2^{\tilde{O}\left(\sqrt{\log(N)}\right)}$ to evaluate this oracle, meaning that the total cost is $2^{\tilde{O}\left(\sqrt{\log(N)}\right)}$.

To avoid this subexponential attack, Jao and De Feo [23] described an analogue of these isogeny-based systems that works with supersingular curves. The endomorphism ring of such curves is a maximal order in a quaternion algebra. The non-commutativity of the (left)-ideals corresponding to isogenies between isomorphism classes of curves thwarts the attack mentioned above, but it also restricts the possibilities offered by supersingular isogenies, which are typically used for a Diffie–Hellman type of key exchange (known as SIDH) and for digital signatures. Most recently, two works revisited isogeny-based cryptosystems by restricting themselves to cases where the subexponential attacks based on the action of $\mathrm{Cl}(\mathcal{O})$ was applicable. The scheme known as CSIDH by Castryck et al. [10] uses supersingular curves and isogenies defined over $\mathbb{F}_p$, while the scheme of De Feo, Kieffer and Smith [15] uses ordinary curves with many practical optimizations. In both cases, the appeal of using commutative structures is to allow more functionalities, such as static-static key exchange protocols that are not possible with SIDH without an expensive Fujisaki–Okamoto transform [2].

Contributions. Let E_1, E_2 be two elliptic curves defined over a finite field such that there is an imaginary quadratic order $\mathcal{O}$ satisfying $\mathcal{O} \simeq \mathrm{End}(E_i)$ for $i = 1, 2$. Let $\Delta = \mathrm{disc}(\mathcal{O})$. In this note, we provide new insight into the security of CSIDH as follows:

1. We describe a quantum algorithm for computing an isogeny between E_1 and E_2 with heuristic asymptotic run time in $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ and with quantum memory in $\mathrm{Poly}\left(\log(|\Delta|)\right)$ and quantumly accessible classical memory in $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$.
2. We show that we can use a variant of this method to compute an isogeny between E_1 and E_2 in time $e^{\left(\frac{1}{\sqrt{2}}+o(1)\right)\sqrt{\ln(|\Delta|)\ln\ln(|\Delta|)}}$ with polynomial memory (both classical and quantum).

Our contributions bear similarities to the recent independent work of Bonnetain and Schrottenloher [7]. The main differences are that they rely on a generating

set $\mathfrak{l}_1, \ldots, \mathfrak{l}_u$ of the class group, where $u \in \Theta(\log(|\Delta|))$, provided with the CSIDH protocol, and that they primarily focused on practical improvements and concrete security levels. Their method inherits Kuperberg's asymptotic complexity which is in $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$. Section 4.2 elaborates on the differences between our algorithm and that of [7]. The run time of the variant described in Contribution 2 is asymptotically comparable to that of the algorithm of Childs, Jao and Soukharev [11], and to that of Bonnetain and Schrottenloher [7] (if its exact time complexity was to be worked out). The main appeal of our variant is the fact that it uses a polynomial amount of memory, which is likely to impact the performances in practice.

Our work is also connected to a recent and independent contribution of Jao, LeGrow, Leonardi and Ruiz–Lopez. The main claim of their work (slides are available online [24]) is an algorithm with heuristic time complexity in $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ that uses quantum polynomial memory and classical memory in $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$. Compared to our Contribution 2, the difference in terms of performances is that our method requires only polynomial classical memory. The main technical difference between our work and that of Jao et al. is an alternative approach to lattice reduction. Both works rely on unproven heuristics: ours pertains to the connectivity of the Caley Graph of the ideal class group of the ring of endomorphisms, while Jao et al. make the assumption that this class group is cyclic, leaving the question of non-cyclic class groups open. Note that by design, the class group is very likely to be cyclic in instances of this problem pertaining to the cryptanalysis of CSIDH. Therefore, the generalization of their method would mostly be of fundamental interest.

2 Mathematical background

An elliptic curve E defined over a finite field $\mathbb{F}_q$ of characteristic $p \neq 2, 3$ is a projective algebraic curve with an affine plane model given by an equation of the form $y^2 = x^3 + ax + b$, where a, $b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$. The set of points of an elliptic curve is equipped with an additive group law. Details about the arithmetic of elliptic curves can be found in many references, such as [33, Chap. 3].

Let E_1, E_2 be two elliptic curves defined over $\mathbb{F}_q$. An isogeny $\phi\colon E_1 \to E_2$ over $\mathbb{F}_q$ (resp. over $\overline{\mathbb{F}}_q$) is a non-constant rational map defined over $\mathbb{F}_q$ (resp. over $\overline{\mathbb{F}}_q$) which sends the identity point on E_1 to the identity point on E_2. The degree of an isogeny is its degree as a rational map, and an isogeny of degree ℓ is called an ℓ-isogeny. Two curves are isogenous over $\mathbb{F}_q$ if and only if they have the same number of points over $\mathbb{F}_q$ (see [36]). Moreover, E_1, E_2 are said to be isomorphic over $\mathbb{F}_q$, or $\mathbb{F}_q$-isomorphic, if there exist isogenies $\phi_1\colon E_1 \to E_2$ and $\phi_2\colon E_2 \to E_1$ over $\mathbb{F}_q$ whose composition is the identity. Two $\overline{\mathbb{F}}_q$-isomorphic elliptic curves have the same j-invariant given by $j := 1728\frac{4a^3}{4a^3+27b^2}$.

An order $\mathcal{O}$ in a number field K such that $[K : \mathbb{Q}] = n$ is a subring of K which is a $\mathbb{Z}$-module of rank n. The notion of ideal of $\mathcal{O}$ can be generalized to

fractional ideals, which are sets of the form $\mathfrak{a} = \frac{1}{d}I$ where I is an ideal of $\mathcal{O}$ and $d \in \mathbb{Z}_{>0}$. A fractional ideal I is said to be invertible if there exists a fractional ideal J such that $IJ = \mathcal{O}$. The invertible fractional ideals form a multiplicative group $\mathcal{I}$, having a subgroup consisting of the invertible principal ideals $\mathcal{P}$. The ideal class group $\mathrm{Cl}(\mathcal{O})$ is by definition $\mathrm{Cl}(\mathcal{O}) := \mathcal{I}/\mathcal{P}$. In $\mathrm{Cl}(\mathcal{O})$, we identify two fractional ideals $\mathfrak{a}, \mathfrak{b}$ if there is $\alpha \in K^*$ such that $\mathfrak{b} = (\alpha)\mathfrak{a}$, where $(\alpha) := \alpha\mathcal{O}$. We denote by $[\mathfrak{a}]$ the class of the fractional ideal $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$. The ideal class group is finite and its cardinality is called the class number $h_{\mathcal{O}}$ of $\mathcal{O}$. For a quadratic order $\mathcal{O}$, the class number satisfies $h_{\mathcal{O}} \leq \sqrt{|\Delta|}\ln(|\Delta|)$ (see [12, §5.10.1]), where Δ is the discriminant of $\mathcal{O}$.

Let E be an elliptic curve defined over $\mathbb{F}_q$. An endomorphism of E is either an isogeny defined over $\overline{\mathbb{F}}_q$ between E and itself, or the zero morphism. The set of endomorphisms of E forms a ring that is denoted by $\mathrm{End}(E)$. For each integer m, the multiplication-by-m map $[m]$ on E is an endomorphism. Therefore, we always have $\mathbb{Z} \subseteq \mathrm{End}(E)$. Moreover, to each isogeny $\phi\colon E_1 \to E_2$ corresponds an isogeny $\widehat{\phi}\colon E_2 \to E_1$ called its dual isogeny. It satisfies $\phi \circ \widehat{\phi} = [m]$ where $m = \deg(\phi)$. For elliptic curves defined over a finite field, we know that $\mathbb{Z} \subsetneq \mathrm{End}(E)$. In this particular case, $\mathrm{End}(E)$ is either an order in an imaginary quadratic field (and has $\mathbb{Z}$-rank 2) or a maximal order in a quaternion algebra ramified at p (the characteristic of the base field) and ∞ (and has $\mathbb{Z}$-rank 4). In the former case, E is said to be ordinary while in the latter it is called supersingular. When a supersingular curve is defined over $\mathbb{F}_p$, then the ring of its $\mathbb{F}_p$-endomorphisms, denoted by $\mathrm{End}_{\mathbb{F}_p}(E)$, is isomorphic to an imaginary quadratic order, much like in the ordinary case.

The endomorphism ring of an elliptic curve plays a crucial role in most algorithms for computing isogenies between curves. Indeed, if E is ordinary (resp. supersingular over $\mathbb{F}_p$), the class group of $\mathrm{End}(E)$ (resp. $\mathrm{End}_{\mathbb{F}_p}(E)$) acts transitively on isomorphism classes of elliptic curves having the same endomorphism ring. More precisely, the class of an ideal $\mathfrak{a} \subseteq \mathcal{O}$ acts on the isomorphism class of a curve E with $\mathrm{End}(E) \simeq \mathcal{O}$ via an isogeny of degree $\mathcal{N}(\mathfrak{a})$ (the algebraic norm of $\mathfrak{a}$). Likewise, each isogeny $\varphi\colon E \to E'$ where $\mathrm{End}(E) \simeq \mathrm{End}(E') \simeq \mathcal{O}$ corresponds (up to isomorphism) to the class of an ideal in $\mathcal{O}$. From an ideal $\mathfrak{a}$ and the ℓ-torsion (where $\ell = \mathcal{N}(\mathfrak{a})$), one can recover the kernel of φ, and then using Vélu's formulae [37], one can derive the corresponding isogeny. We denote by $[\mathfrak{a}] * \overline{E}$ the action of the ideal class of $\mathfrak{a}$ on the isomorphism class of the curve E. The typical strategy to evaluate the action of $[\mathfrak{a}]$ is to decompose it as a product of classes of prime ideals of small norm ℓ, and evaluate the action of each prime ideal as an ℓ-isogeny. This strategy was described by Couveignes [13], Galbraith–Hess–Smart [16], and later by Bröker–Charles–Lauter [9] and reused in many subsequent works.

Notation: In this paper, log denotes the base 2 logarithm while ln denotes the natural logarithm.

3 The CSIDH non-interactive key exchange

As pointed out in [17], the original SIDH key agreement protocol is not secure when using the same secret key over multiple instances of the protocol. This can be fixed by a Fujisaki–Okamoto transform [2] at the cost of a drastic loss of performance, requiring additional points in the protocol. These issues motivated the description of CSIDH [10] which uses supersingular curves defined over $\mathbb{F}_p$.

When Alice and Bob wish to create a shared secret, they rely on their secret keys $[\mathfrak{a}]$ and $[\mathfrak{b}]$ which are classes of ideals in the ideal class group of $\mathcal{O}$, where $\mathcal{O}$ is isomorphic to the $\mathbb{F}_p$-endomorphism ring of a supersingular curve E defined over $\mathbb{F}_p$. This key exchange procedure resembles the original Diffie–Hellman protocol [14]. Alice and Bob proceed as follow:

- Alice sends $[\mathfrak{a}] * \overline{E}$ to Bob.
- Bob sends $[\mathfrak{b}] * \overline{E}$ to Alice.

Then Alice and Bob can separately recover their shared secret

$$[\mathfrak{a}\mathfrak{b}] * \overline{E} = [\mathfrak{b}] * [\mathfrak{a}] * \overline{E} = [\mathfrak{a}] * [\mathfrak{b}] * \overline{E}.$$

The existence of a quantum subexponential attack forces the users to update the size of keys at a faster pace (or by larger increments) than in the regular SIDH protocol against which we only know quantum exponential attacks. This is partly compensated by the fact that elements are represented in $\mathbb{F}_p$, and are thus more compact than elements of $\mathbb{F}_{p^2}$ needed in SIDH (because the corresponding curves are defined over $\mathbb{F}_{p^2}$). Recommended parameter sizes and attack costs from [10] for 80, 128, and 256 bit security are listed in Table 1. In Table 1, the cost is in number of operations. These values do not account for the memory costs (the security estimates are therefore more conservative than if memory costs were accounted for). The NIST security levels are defined in the call for proposals for the Post Quantum Cryptography project [29]. Note that subsequent works such as that of Bonnetain and Schrottenloher [7] have suggested different values.

Table 1. Claimed security of CSIDH [10, Table 1].

NIST	$\log(p)$	Cost quantum attack	Cost classical attack
1	512	2^{62}	2^{128}
3	1024	2^{94}	2^{256}
5	1792	2^{129}	2^{448}

4 Asymptotic complexity of isogeny computation

In this section, we show how to combine the general framework for computing isogenies between curves whose endomorphism ring is isomorphic to a quadratic

order (due to Childs, Jao and Soukharev [11] in the ordinary case and to Biasse, Jao and Sankar in the supersingular case [5]) with the efficient algorithm of Biasse, Fieker and Jacobson [4] for evaluating the class group action to produce a quantum algorithm that finds an isogeny between E_1 and E_2. We give two variants of our method:

- Heuristic time complexity $2^{O(\log(|\Delta|))}$, polynomial quantum memory and quantumly accessible classical memory in $2^{O(\log(|\Delta|))}$.
- Heuristic time complexity $e^{\left(\frac{1}{\sqrt{2}}+o(1)\right)\sqrt{\ln(|\Delta|)\ln\ln(|\Delta|)}}$ with polynomial memory (both classical and quantum).

4.1 Isogenies from solutions to the Hidden Subgroup Problem

As shown in [5, 11], the computation of an isogeny between E_1 and E_2 such that there is an imaginary quadratic order with $\mathcal{O} \simeq \mathrm{End}(E_i)$ for $i = 1, 2$ can be done by exploiting the action of the ideal class group of $\mathcal{O}$ on isomorphism classes of curves with endomorphism ring isomorphic to $\mathcal{O}$. In particular, this concerns the cases of

- ordinary curves, and
- supersingular curves defined over $\mathbb{F}_p$.

Assume we are looking for $\mathfrak{a}$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$. Let $A = \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_k\mathbb{Z} \simeq \mathrm{Cl}(\mathcal{O})$ be the elementary decomposition of $\mathrm{Cl}(\mathcal{O})$. Then we define a quantum oracle $f : \mathbb{Z}/2\mathbb{Z} \ltimes A \to \{\text{quantum states}\}$ by

$$f(x, \boldsymbol{y}) := \begin{cases} |[\mathfrak{a}_{\boldsymbol{y}}] * \overline{E}_1\rangle & \text{if } x = 0, \\ |[\mathfrak{a}_{-\boldsymbol{y}}] * \overline{E}_2\rangle & \text{if } x = 1, \end{cases} \tag{1}$$

where $[\mathfrak{a}_{\boldsymbol{y}}]$ is the element of $\mathrm{Cl}(\mathcal{O})$ corresponding to $\boldsymbol{y} \in A$ via the isomorphism $\mathrm{Cl}(\mathcal{O}) \simeq A$. Let H be the subgroup of $\mathbb{Z}/2\mathbb{Z} \ltimes A$ of the periods of f. This means that $f(x, \boldsymbol{y}) = f(x', \boldsymbol{y}')$ if and only if $(x, \boldsymbol{y}) - (x', \boldsymbol{y}') \in H$. Then $H = \{(0, \boldsymbol{0}), (1, \boldsymbol{s})\}$ where $\boldsymbol{s} \in A$ such that $[\mathfrak{a}_{\boldsymbol{s}}] * \overline{E}_1 = \overline{E}_2$. The computation of $\boldsymbol{s}$ can thus be done through the resolution of the Hidden Subgroup Problem in $\mathbb{Z}/2\mathbb{Z} \ltimes A$. In [11, Sec. 5], Childs, Jao and Soukharev generalized the subexponential-time polynomial space dihedral HSP algorithm of Regev [30] to the case of an arbitrary Abelian group A. Its run time is in $e^{\left(\sqrt{2}+o(1)\right)\sqrt{\ln(|A|)\ln\ln(|A|)}}$ with a polynomial memory requirement. Kuperberg [27] describes a family of algorithms, one of which has running time in $e^{O\left(\sqrt{\log(|A|)}\right)}$ while requiring polynomial quantum memory and $e^{O\left(\sqrt{\log(|A|)}\right)}$ quantumly accessible classical memory. The high-level approach for finding an isogeny from the dihedral HSP is sketched in Algorithm 1.

Proposition 1. *Let $N = \#\,\mathrm{Cl}(\mathcal{O}) \sim \sqrt{|\Delta|}$. Algorithm 1 is correct and requires:*

- $e^{O\left(\sqrt{\log(N)}\right)}$ *queries to the oracle defined by* (1) *while requiring a* $\mathrm{Poly}(\log(N))$ *quantum memory and* $e^{O\left(\sqrt{\log(N)}\right)}$ *quantumly accessible classical memory overhead when using Kuperberg's second dihedral HSP algorithm [27] in Step 2.*

Algorithm 1 Quantum algorithm for evaluating the action in $\mathrm{Cl}(\mathcal{O})$

Input: Elliptic curves E_1, E_2, imaginary quadratic order $\mathcal{O}$ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$ such that there is $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ satisfying $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$.

Output: $[\mathfrak{a}]$

1: Compute $A = \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_k\mathbb{Z}$ such that $A \simeq \mathrm{Cl}(\mathcal{O})$.
2: Find $H = \{(0,0), (1,s)\}$ by solving the HSP in $\mathbb{Z}/2\mathbb{Z} \ltimes A$ with oracle (1).
3: **return** $[\mathfrak{a}_s]$

– $e^{\left(\sqrt{2}+o(1)\right)\sqrt{\ln(N)\ln\ln(N)}}$ *queries to the oracle defined by* (1) *while requiring only polynomial memory overhead when using the dihedral HSP method of [11, Sec. 5] in Step 2.*

Remark 1. The cost of Algorithm 1 is dominated by Step 2. Indeed, Step 1 can be done by using an algorithm for solving the HSP in a commutative group. Even when the dimension grows to infinity, this step is known to run in polynomial time [6].

Remark 2. Algorithm 1 only returns the ideal class $[\mathfrak{a}]$ whose action on $\overline{E}_1$ gives us $\overline{E}_2$. This is all we are interested in as far as the analysis of isogeny-based cryptosystems goes. However, this is not an isogeny between E_1 and E_2. We can use this ideal to derive an actual isogeny by evaluating the action of $[\mathfrak{a}]$ using the oracle of Section 4.2 together with the method of [9, Alg. 4.1]. This returns an isogeny $\phi : E_1 \to E_2$ as a composition of isogenies of small degree $\phi = \prod_i \phi_i^{e_i}$ with the same time complexity as Algorithm 1. Also note that the output fits in polynomial space if the product is not evaluated, otherwise, it needs $2^{\tilde{O}\left(\sqrt[3]{\log(N)}\right)}$ memory.

4.2 The quantum oracle

To compute the oracle defined in (1), Childs, Jao and Soukharev [11] used a purely classical subexponential method derived from the general subexponential class group computation algorithm of Hafner and McCurley [19]. This approach, mentioned in [10], was first suggested by Couveignes [13]. In a recent independent work [7], Bonnetain and Schrottenloher used a method that bears similarities with our oracle described in this section. They combined a quantum algorithm for computing the class group with classical methods from Biasse, Fieker and Jacobson [4, Alg. 7] for evaluating the action of $[\mathfrak{a}]$ with a precomputation of $\mathrm{Cl}(\mathcal{O})$. More specifically, let $\mathfrak{l}_1, \ldots, \mathfrak{l}_u$ be prime ideals used to create the secret ideal $\mathfrak{a}$ of Alice. This means that there are (small) $(e_1, \ldots, e_u) \in \mathbb{Z}^u$ such that $\mathfrak{a} = \prod_i \mathfrak{l}_i^{e_i}$. Let $\mathcal{L}$ be the lattice of relations between $\mathfrak{l}_1, \ldots, \mathfrak{l}_u$, i.e. the lattice of all the vectors $(f_1, \ldots, f_u) \in \mathbb{Z}^u$ such that $\prod_i \mathfrak{l}_i^{f_i}$ is principal. In other words, the ideal class $\left[\prod_i \mathfrak{l}_i^{f_i}\right]$ is the neutral element of $\mathrm{Cl}(\mathcal{O})$. The high-level approach used in [7] deriving from [4, Alg. 7] is the following:

1. Compute a basis B for $\mathcal{L}$.

2. Find a BKZ-reduced basis B' of $\mathcal{L}$.
3. Find $(h_1, \ldots, h_u) \in \mathbb{Z}^u$ such that $[\mathfrak{a}] = \left[\prod_i \mathfrak{l}_i^{h_i}\right]$.
4. Use Babai's nearest plane method on B' to find short $(h'_1, \ldots, h'_u) \in \mathbb{Z}^u$ such that $[\mathfrak{a}] = \left[\prod_i \mathfrak{l}_i^{h'_i}\right]$.
5. Evaluate the action of $\left[\prod_i \mathfrak{l}_i^{h'_i}\right]$ on $\overline{E}_1$ by applying repeatedly the action of the $\mathfrak{l}_i$ for $i = 1, \ldots, u$.

Steps 1 and 2 can be performed as a precomputation. Step 1 takes quantum polynomial time by using standard techniques for solving an instance of the Abelian Hidden Subgroup Problem in $\mathbb{Z}^u$ where $p = 4l_1 \cdots l_u - 1$ for small primes $l_1, \ldots, l_u$.

The oracle of Childs, Jao and Soukharev [11] has asymptotic time complexity in $2^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ and requires subexponential space due to the need for the storage of the ℓ-th modular polynomial $\Phi_\ell(X, Y)$ for ℓ up to $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$. Indeed, the size of $\Phi_\ell(X, Y)$ is proportional to ℓ. The oracle of Bonnetain and Schrottenloher [7] relies on BKZ [31] lattice reduction in a lattice in $\mathbb{Z}^u$. Typically, $u \in \Theta(\log(p)) = \Theta(\log(|\Delta|))$, since $\sum_{q \leq l} \log(q) \in \Theta(l)$. In addition to not having a proven space complexity bound, the complexity of BKZ cannot be in $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ unless the block size is at least in $\Theta\left(\sqrt{\log(|\Delta|)}\right)$, which forces the overall complexity to be at best in $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$.

Our strategy differs from that of Bonnetain and Schrottenloher on the following points:

- Our algorithm does not require the basis $\mathfrak{l}_1, \ldots, \mathfrak{l}_u$ provided with CSIDH.
- The complexity of our oracle is in $e^{\tilde{O}\left(\sqrt[3]{\log(|\Delta|)}\right)}$ (instead of $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ for the method of [7]), thus leading to an overall complexity of $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ (instead of $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ for the method of [7]).
- We specify the use of a variant of BKZ with a proven poly-space complexity.

To avoid the dependence on the parameter u, we need to rely on the heuristics stated by Biasse, Fieker and Jacobson [4] on the connectivity of the Caley graph of the ideal class group when a set of edges is $S \subseteq \{\mathfrak{p} : \mathcal{N}(\mathfrak{p}) \in \mathrm{Poly}(\log(|\Delta|))\}$ with $\#S \leq \log(|\Delta|)^{2/3}$ where Δ is the discriminant of $\mathcal{O}$. By assuming [4, Heuristic 2], we state that each class of $\mathrm{Cl}(\mathcal{O})$ has a representation over the class of ideals in S with exponents less than $e^{\log^{1/3}(|\Delta|)}$. A quick calculation shows that there are asymptotically many more such products than ideal classes, but their distribution is not well enough understood to conclude that all classes decompose over S with a small enough exponent vector. Numerical experiments reported in [4, Table 2] showed that decompositions of random ideal classes over the first $\log^{2/3}(|\Delta|)$ split primes always had exponents significantly less than $e^{\log^{1/3}(|\Delta|)}$.

Table 2. Maximal exponent occurring in short decompositions (over 1000 random elements of the class group). Table 2 of [4].

$\log_{10}(\lvert\Delta\rvert)$	$\log^{2/3}(\lvert\Delta\rvert)$	Maximal coefficient	$e^{\log^{1/3}(\lvert\Delta\rvert)}$
20	13	6	36
25	15	8	48
30	17	7	61
35	19	9	75
40	20	10	91
45	22	14	110
50	24	13	130

Heuristic 1 (With parameter $c > 1$) *Let $c > 1$ and $\mathcal{O}$ be an imaginary quadratic order of discriminant Δ. Then there are $(\mathfrak{p}_i)_{i \leq k}$ for $k = \log^{2/3}(|\Delta|)$ split prime ideals of norm less than $\log^c(|\Delta|)$ whose classes generate $\mathrm{Cl}(\mathcal{O})$. Furthermore, each class of $\mathrm{Cl}(\mathcal{O})$ has a representative of the form $\prod_i \mathfrak{p}_i^{n_i}$ for $|n_i| \leq e^{\log^{1/3}|\Delta|}$.*

A default choice for our set S could be the first $\log^{2/3}(|\Delta|)$ split primes of $\mathcal{O}$ (as in Table 2). We can derive our results under the weaker assumption that the $\log^{2/3}(|\Delta|)$ primes generating the ideal class group do not have to be the first consecutive primes. Assume we know that $\mathrm{Cl}(\mathcal{O})$ is generated by at most $\log^{2/3}(|\Delta|)$ distinct classes of the split prime ideals of norm up to $\log^c(|\Delta|)$ for some constant $c > 0$. Our algorithm needs to first identify these prime ideals as they might not be the first consecutive primes. Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ be the prime ideals of norm up to $\log^c(|\Delta|)$. We first compute a basis for the lattice $\mathcal{L}$ of vectors $(e_1, \ldots, e_k)$ such that $\prod_i \mathfrak{p}_i^{e_i}$ is principal (in other words, the ideal class $[\prod_i \mathfrak{p}_i^{e_i}]$ is trivial). Let M be the matrix whose rows are the vectors of a basis of $\mathcal{L}$. There is a polynomial time (and space) algorithm that finds a unimodular matrix U such that

$$UM = H = \begin{bmatrix} h_{1,1} & 0 & \ldots & 0 \\ \vdots & h_{2,2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ * & * & \ldots & h_{k,k} \end{bmatrix},$$

where H is in Hermite Normal Form [35]. The matrix H represents the unique upper triangular basis of $\mathcal{L}$ such that $h_{i,i} > 0$, and $h_{j,j} > h_{i,j}$ for $i > j$. Every time $h_{i,i} = 1$, this means that we have a relation of the form $[\mathfrak{p}_i] = \left[\prod_{j<i} \mathfrak{p}_j^{-h_{i,j}}\right]$. In other words, $[\mathfrak{p}_i] \in \langle [\mathfrak{p}_1], \ldots, [\mathfrak{p}_{i-1}] \rangle$. On the other hand, if $h_{i,i} \neq 1$, then $[\mathfrak{p}_i] \notin \langle [\mathfrak{p}_1], \ldots, [\mathfrak{p}_{i-1}] \rangle$. Our algorithm proceeds by computing the HNF of M, and every time $h_{i,i} \neq 1$, it moves $\mathfrak{p}_i$ to the beginning of the list of primes, and moves the column i to the first column, recomputes the HNF and iterates the process. In the end, the first $\log^{2/3}(|\Delta|)$ primes in the list generate $\mathrm{Cl}(\mathcal{O})$.

Algorithm 2 Computation of $\log^{2/3}(|\Delta|)$ primes that generate $\mathrm{Cl}(\mathcal{O})$

Input: Order $\mathcal{O}$ of discriminant Δ and $c > 0$.
Output: $\log^{2/3}(|\Delta|)$ split primes whose classes generate $\mathrm{Cl}(\mathcal{O})$.
1: $S \leftarrow \{\text{Split primes } \mathfrak{p}_1, \ldots, \mathfrak{p}_k \text{ of norm less than } \log^c(|\Delta|)\}$.
2: $\mathcal{L} \leftarrow$ lattice of vectors $(e_1, \ldots, e_k)$ such that $\prod_i \mathfrak{p}_i^{e_i}$ is principal using [6].
3: Compute the matrix $H \in \mathbb{Z}^{k \times k}$ of a basis of $\mathcal{L}$ in HNF using [35, Ch. 6].
4: **for** $j = k$ down to $\log^{2/3}(|\Delta|) + 1$ **do**
5: **while** $h_{j,j} \neq 1$ **do**
6: Insert $\mathfrak{p}_j$ at the beginning of S.
7: Insert the j-th column at the beginning of the list of columns of H.
8: $H \leftarrow \mathrm{HNF}(H)$.
9: **end while**
10: **end for**
11: **return** $\{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ for $s = \log^{2/3}(|\Delta|)$.

Proposition 2. *Assuming Heuristic 1 for the parameter c, Algorithm 2 is correct and runs in polynomial time in* $\log(|\Delta|)$.

Proof. Step 2 can be done in quantum polynomial time with the S-unit algorithm of Biasse and Song [6]. Assuming that $\log^{2/3}(|\Delta|)$ primes of norm less than $\log^c(|\Delta|)$ generate $\mathrm{Cl}(\mathcal{O})$, the loop of Steps 5 to 9 is entered at most j times as one of $[\mathfrak{p}_1], \ldots, [\mathfrak{p}_j]$ must be in the subgroup generated by the other $j-1$ ideal classes. The HNF computation runs in polynomial time, therefore the whole procedure runs in polynomial time. □

Once we have $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$, we compute with Algorithm 3 a reduced basis B' of the lattice $\mathcal{L} \subseteq \mathbb{Z}^s$ of the vectors $(e_1, \ldots, e_s)$ such that $[\prod_i \mathfrak{p}_i^{e_i}]$ is trivial, and we compute the generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_l$ such that $\mathrm{Cl}(\mathcal{O}) = \langle \mathfrak{g}_1 \rangle \times \cdots \times \langle \mathfrak{g}_l \rangle$ together with vectors $\boldsymbol{v}_i$ such that $\mathfrak{g}_i = \prod_j \mathfrak{p}_j^{v_{i,j}}$.

Lemma 1. *Let $\mathcal{L}$ be an n-dimentional lattice with input basis $B \in \mathbb{Z}^{n \times n}$, and let $\beta < n$ be a block size. Then the BKZ variant of [21] used with Kannan's enumeration technique [26] returns a basis $\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_n$ such that*

$$\|\boldsymbol{b}'_1\| \leq e^{\frac{n}{\beta}\ln(\beta)(1+o(1))}\lambda_1(\mathcal{L}),$$

using time $\mathrm{Poly}(n, \mathrm{Size}(B))\beta^{\beta\left(\frac{1}{2e}+o(1)\right)}$ *and polynomial space.*

Proof. According to [21, Th. 1], $\|\boldsymbol{b}'_1\| \leq 4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3}\lambda_1(\mathcal{L})$ where γ_β is the Hermite constant in dimension β. As asymptotically $\gamma_\beta \leq \frac{1.744\beta}{2\pi e}(1+o(1))$ (see [25]), we get that $4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3} \leq e^{\frac{n}{\beta}\ln(\beta)(1+o(1))}$. Moreover, this reduction is obtained with a number of calls to Kannan's algorithm that is bounded by $\mathrm{Poly}(n, \mathrm{Size}(B))$. According to [22, Th. 2], each of these calls takes time $\mathrm{Poly}(n, \mathrm{Size}(B))\beta^{\beta\left(\frac{1}{2e}+o(1)\right)}$ and polynomial space, which terminates the proof. □

Algorithm 3 Precomputation for the oracle

Input: Order $\mathcal{O}$ of discriminant Δ and $c > 0$.

Output: Split prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ whose classes generate $\mathrm{Cl}(\mathcal{O})$ where $s = \log^{2/3}(|\Delta|)$, reduced basis B' of the lattice $\mathcal{L}$ of vectors $(e_1, \ldots, e_s)$ such that $\left[\prod_i \mathfrak{p}_i^{e_i}\right]$ is trivial, generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_l$ such that $\mathrm{Cl}(\mathcal{O}) = \langle \mathfrak{g}_1 \rangle \times \cdots \times \langle \mathfrak{g}_l \rangle$ and vectors $\boldsymbol{v}_i$ such that $\mathfrak{g}_i = \prod_j \mathfrak{p}_j^{v_{i,j}}$.

$\mathfrak{p}_1, \ldots, \mathfrak{p}_s \leftarrow$ output of Algorithm 2.
$\mathcal{L} \leftarrow$ lattice of vectors $(e_1, \ldots, e_s)$ such that $\prod_i \mathfrak{p}_i^{e_i}$ is principal.
Compute a BKZ-reduced matrix $B' \in \mathbb{Z}^{s \times s}$ of a basis of $\mathcal{L}$ with block size $\log^{1/3}(|\Delta|)$.
Compute $U, V \in \mathrm{GL}_s(\mathbb{Z})$ such that $UB'V = \mathrm{diag}(d_1, \ldots, d_s)$ is the Smith Normal Form of B'.
$l \leftarrow \min_{i \leq s}\{i \mid d_i \neq 1\}$. For $i \leq l$, $\boldsymbol{v_i} \leftarrow i$-th column of V.
$V' \leftarrow V^{-1}$. For $i \leq l$, $\mathfrak{g}_i \leftarrow \prod_{j \leq s} \mathfrak{p}_j^{v'_{i,j}}$.
return $\{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$, B', $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_l\}$, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l\}$.

Proposition 3. *Assuming Heuristic 1 for c, Algorithm 3 is correct, runs in time* $e^{\tilde{O}\left(\sqrt[3]{\log(|\Delta|)}\right)}$ *and has polynomial space complexity.*

The precomputation of Algorithm 3 allows us to design the quantum circuit that implements the function described in (1). Generic techniques due to Bennett [3] convert any algorithm taking time T and space S into a reversible algorithm taking time $T^{1+\epsilon}$, for an arbitrary small $\epsilon > 0$, and space $O(S \log T)$. From a high-level point of view, this is simply the adaptation of the method of Biasse–Fieker–Jacobson [4, Alg. 7] to the quantum setting.

Algorithm 4 Quantum oracle for implementing f defined in (1)

Input: Curves E_1, E_2. Order $\mathcal{O}$ of discriminant Δ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$. Split prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ whose classes generate $\mathrm{Cl}(\mathcal{O})$ where $s = \log^{2/3}(|\Delta|)$, reduced basis B' of the lattice $\mathcal{L}$ of vectors $(e_1, \ldots, e_s)$ such that $\left[\prod_i \mathfrak{p}_i^{e_i}\right]$ is trivial, generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_l$ such that $\mathrm{Cl}(\mathcal{O}) = \langle \mathfrak{g}_1 \rangle \times \cdots \times \langle \mathfrak{g}_l \rangle$ and vectors $\boldsymbol{v}_i$ such that $\mathfrak{g}_i = \prod_j \mathfrak{p}_j^{v_{i,j}}$. Ideal class $[\mathfrak{a}_{\boldsymbol{y}}] \in \mathrm{Cl}(\mathcal{O})$ represented by the vector $\boldsymbol{y} = (y_1, \ldots, y_l) \in \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_l\mathbb{Z} \simeq \mathrm{Cl}(\mathcal{O})$, and $x \in \mathbb{Z}/2\mathbb{Z}$.

Output: $f(x, \boldsymbol{y})$.

$\boldsymbol{y} \leftarrow \sum_{i \leq l} y_i \boldsymbol{v_i} \in \mathbb{Z}^s$ (now $[\mathfrak{a}_{\boldsymbol{y}}] = \left[\prod_i \mathfrak{p}_i^{y_i}\right]$).
Use Babai's nearest plane method with the basis B' to find $\boldsymbol{u} \in \mathcal{L}$ close to $\boldsymbol{y}$.
$\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{u}$.
If $x = 0$ **then** $\overline{E} \leftarrow \overline{E}_1$ **else** $\overline{E} \leftarrow \overline{E}_2$.
for $i \leq s$ **do**
    for $j \leq y_i$ **do**
        $\overline{E} \leftarrow [\mathfrak{p}_i] * \overline{E}$.
    end for
end for
return $|\overline{E}\rangle$.

To bound the run time of Algorithm 4, we need to assume that the BKZ-reduced basis computed in Algorithm 3 has good geometric properties. We assume the following standard heuristic.

Heuristic 2 (Geometric Series Assumption) *The basis B' computed in Algorithm 3 satisfies the Geometric Series Assumption (GSA): there is $0 < q < 1$ such that $\|\widehat{\boldsymbol{b}'_i}\| = q^{i-1}\|\boldsymbol{b_1}\|$ where $\left(\widehat{\boldsymbol{b}'_i}\right)_{i\leq n}$ is the Gram-Schmidt basis corresponding to B'.*

Proposition 4. *Assuming Heuristic 1 for some $c > 1$ and Heuristic 2, Algorithm 4 is correct and runs in quantum time $e^{\tilde{O}\left(\sqrt[3]{\log(|\Delta|)}\right)}$ and has polynomial space complexity.*

Proof. Each group action of Step 7 is polynomial in $\log(p)$ and in $\mathcal{N}(\mathfrak{p}_i)$. Moreover, Babai's algorithm runs in polynomial time and returns $\boldsymbol{u}$ such that

$$\|\boldsymbol{y} - \boldsymbol{u}\| \leq \frac{1}{2}\sqrt{\sum_i \|\widehat{\boldsymbol{b}'_i}\|^2} \leq \frac{1}{2}\sqrt{n}\|\boldsymbol{b'}_1\| \in e^{\tilde{O}\left(\sqrt[3]{\log(|\Delta|)}\right)}.$$

Therefore, the y_i are in $e^{\tilde{O}\left(\sqrt[3]{\log(|\Delta|)}\right)}$, which is the cost of Steps 5 to 9. The main observation allowing us to reduce the search to a close vector to the computation of a BKZ-reduced basis is that Heuristic 1 gives us the promise that there is $\boldsymbol{u} \in \mathcal{L}$ at distance less than $e^{\sqrt[3]{\log(|\Delta|)}(1+o(1))}$ from $\boldsymbol{y}$. □

Corollary 1. *Let E_1, E_2 be two elliptic curves and $\mathcal{O}$ be an imaginary quadratic order of discriminant Δ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$. Then assuming Heuristic 1 for some constant $c > 0$, there is a quantum algorithm for computing $[\mathfrak{a}]$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$ with:*

- *heuristic time complexity $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$, polynomial quantum memory and $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ quantumly accessible classical memory,*
- *heuristic time complexity $e^{\left(\frac{1}{\sqrt{2}}+o(1)\right)\sqrt{\ln(|\Delta|)\ln\ln(|\Delta|)}}$ with polynomial memory (both classical and quantum).*

Remark 3. We referred to Heuristic 1 as Biasse, Fieker and Jacobson [4] provided numerical data supporting it. Heuristic 1 may be relaxed in the proof of the $e^{O\left(\sqrt{\log(|\Delta|)}\right)}$ asymptotic run time. As long as a number k in $\tilde{O}\left(\log^{1-\varepsilon}(|\Delta|)\right)$ of prime ideals of polynomial norm generate the ideal class group and that each class has at least one decomposition involving exponents less than $e^{\tilde{O}\left(\log^{1/2-\varepsilon}(|\Delta|)\right)}$, the result still holds by BKZ-reducing with block size $\beta = \sqrt{k}$.

For the poly-space variant, these conditions can be relaxed even further. It is known under GRH that a number k in $\tilde{O}\left(\log(|\Delta|)\right)$ of prime ideals of norm less than $12\log^2(|\Delta|)$ generate the ideal class group. We only need to argue that each class can be decomposed with exponents bounded by $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$. Then by using the oracle of Algorithm 4 with block size $\beta = \sqrt{k}$, we get a run time of $e^{\tilde{O}\left(\sqrt{\log(|\Delta|)}\right)}$ with a poly-space requirement.

5 A Remark on subgroups

It is well-known that the cost of quantum and classical attacks on isogeny based cryptosystems is more accurately measured by the size of the subgroup generated by the ideal classes used in the cryptosystem. As stated in [10, Sec. 7.1], in order to ensure that this is sufficiently large with high probability, the class group must have a large cyclic subgroup of order M, where M is not much smaller than the class number N. Assuming the Cohen-Lenstra heuristics this will be the case with high probability and, according to Hamdy and Saidak [20], one even expects a large prime-order subgroup.

It is an open problem as to whether the knowledge of smaller subgroups of the class group can be exploited to reduce the security of CSIDH; the current belief (see [10, p.20]) is that there is no way to do this. There are nevertheless minor considerations that can easily be taken into account when selecting CSIDH parameters to minimize risk in this regard, stemming from the practical difficulties in constructing quadratic fields whose class numbers have a given divisor.

Constructing system parameters for which the class number has a known divisor could be done by a quantum adversary using the polynomial-time algorithm to compute the class group and trial-and-error. Using classical computation, this is in most cases infeasible because the recommended discriminant sizes are too large to compute the class number. Known methods to construct discriminants for which the class number has a given divisor M use a classical result of Nagell [28] relating the problem to finding discriminants $\Delta = c^2 D$ that satisfy $c^2 D = a^2 - 4b^M$ for integers a, b, c. These methods thus produce discriminants that are exponential in M, too large for practical purposes.

The one exception where classical computation can be used to find class numbers with a known divisor is when the divisor $M = 2^k$. Bosma and Stevenhagen [8] give an algorithm, formalizing methods described by Gauss [18, Sec. 286] and Shanks [32], to compute the 2-Sylow subgroup of the class group of a quadratic field. In addition to describing an algorithm that works in full generality, they prove that the algorithm runs in expected time polynomial in $\log(|\Delta|)$. Using this algorithm would enable an adversary to use trial-and-error efficiently to generate random primes p until the desired power of 2 divides the class number.

The primes p recommended for use with CSIDH are not amenable to this method, because they are congruent to $3 \bmod 4$, guaranteeing that the class number of the non-maximal order of discriminant $-4p$ is odd. However, in Section 4 of [10], the authors write that they pick $p \equiv 3 \pmod 4$ because it makes it easy to write down a supersingular curve, but that "in principle, this constraint is not necessary for the theory to work". We suggest that restricting to primes $p \equiv 3 \pmod 4$ is also desirable in order to avoid unnecessary potential vulnerabilities via the existence of even order subgroups.

6 Conclusion

We described two variants of a quantum algorithm for computing an isogeny between two elliptic curves E_1, E_2 defined over a finite field such that there is

an imaginary quadratic order $\mathcal{O}$ satisfying $\mathcal{O} \simeq \mathrm{End}(E_i)$ for $i = 1, 2$ with $\Delta = \mathrm{disc}(\mathcal{O})$. Our first variant runs in heuristic asymptotic run time $2^{O\left(\sqrt{\log(|\Delta|)}\right)}$ and requires polynomial quantum memory and $2^{O\left(\sqrt{\log(|\Delta|)}\right)}$ quantumly accessible classical memory. The second variant of our algorithm relying on Regev's dihedral HSP solver [30] runs in time $e^{\left(\frac{1}{\sqrt{2}}+o(1)\right)\sqrt{\ln(|\Delta|)\ln\ln(|\Delta|)}}$ while relying only on polynomial (classical and quantum) memory. These variants of the HSP-based algorithms for computing isogenies have the best asymptotic complexity, but we left the assessment of their actual cost on specific instances such as the proposed CSIDH parameters [10] for future work. Some of the constants involved in lattice reduction were not calculated, and more importantly, the role of the memory requirement should be addressed in light of the recent results on the topic [1].

Acknowledgments

The authors thank Léo Ducas for useful comments on the memory requirements of the BKZ algorithm. The authors thank Noah Stephens-Davidowitz for information on the resolution of the approximate CVP. The authors also thank Tanja Lange and Benjamin Smith for useful comments on an earlier version of this draft.

References

1. G. Adj, D. Cervantes-Vázquez, J.-J. Chi-Domínguez, A. Menezes, and F. Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. Cryptology ePrint Archive, Report 2018/313, 2018. `https://eprint.iacr.org/2018/313`.
2. R. Azarderakhsh, D. Jao, and C. Leonardi. Post-quantum static-static key agreement using multiple protocol instances. In C. Adams and J. Camenisch, editors, *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*, pages 45–63. Springer, 2017.
3. C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989.
4. J.-F. Biasse, C. Fieker, and M. J. Jacobson, Jr. Fast heuristic algorithms for computing relations in the class group of a quadratic order, with applications to isogeny evaluation. *LMS Journal of Computation and Mathematics*, 19(A):371390, 2016.
5. J.-F. Biasse, D. Jao, and A. Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In W. Meier and D. Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2014.
6. J.-F. Biasse and F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In

R. Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 893–902. SIAM, 2016.
7. X. Bonnetain and A. Schrottenloher. Quantum security analysis of csidh and ordinary isogeny-based schemes. Cryptology ePrint Archive, Report 2018/537, 2018. `https://eprint.iacr.org/2018/537`.
8. W. Bosma and P. Stevenhagen. On the computation of quadratic 2-class groups. *Journal de Thorie des Nombres de Bordeaux*, 8(2):283–313, 1996.
9. R. Bröker, D. Xavier Charles, and K. Lauter. Evaluating large degree isogenies and applications to pairing based cryptography. In S. Galbraith and K. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, Lecture Notes in Computer Science, pages 100–112. Springer, 2008.
10. W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. CSIDH: An efficient post-quantum commutative group action. Cryptology ePrint Archive, Report 2018/383, 2018. `https://eprint.iacr.org/2018/383`, to appear in Asiacrypt 2018.
11. A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1 – 29, 2013.
12. H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, 1991.
13. J.-M. Couveignes. Hard homgeneous spaces. http://eprint.iacr.org/2006/291.
14. W. Diffie and M. Helman. New directions in cryptography. *IEEE Transactions on Information Society*, 22(6):644–654, november 1976.
15. L. De Feo, J. Kieffer, and B. Smith. Towards practical key exchange from ordinary isogeny graphs. Cryptology ePrint Archive, Report 2018/485, 2018. `https://eprint.iacr.org/2018/485`, to appear in Asiacrypt 2018.
16. S. Galbraith, F. Hess, and N. Smart. Extending the GHS weil descent attack. In L. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 29–44. Springer, 2002.
17. S. Galbraith, C. Petit, B. Shani, and Y. B. Ti. On the security of supersingular isogeny cryptosystems. In J. H. Cheon and T. Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91, 2016.
18. C. F. Gauß. *Disquisitiones Arithmeticae*. Springer Verlag, 1986. English edition: translated by A.A. Clark.
19. J. Hafner and K. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of American Mathematical Society*, 2:839–850, 1989.
20. S. Hamdy and F. Saidak. Arithmetic properties of class numbers of imaginary quadratic fields. *JP Journal of Algebra, Number Theory and Applications*, 6(1):129–148, 2006.
21. G. Hanrot, X. Pujol, and D. Stehlé. Terminating BKZ. *IACR Cryptology ePrint Archive*, 2011:198, 2011.
22. G. Hanrot and D. Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In A. Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, vol-

ume 4622 of *Lecture Notes in Computer Science*, pages 170–186. Springer Berlin Heidelberg, 2007.
23. D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Proceedings of the 4th International Conference on Post-Quantum Cryptography*, PQCrypto'11, pages 19–34, Berlin, Heidelberg, 2011. Springer-Verlag.
24. D. Jao, J. LeGrow, C. Leonardi, and L. Ruiz-Lopez. A subexponential-time, polynomial quantum space algorithm for inverting the cm action. Slides of presentation at the MathCrypt conference, 2018. `https://drive.google.com/file/d/15nkb9j0GKyLujYfAb8Sfz3TjBY5PWOCT/view`.
25. A. Kabatyanskii and V. Levenshtein. Bounds for packings. on a sphere and in space. *Proulcmy Peredacha informatsü*, 14:1–17, 1978.
26. R. Kannan. Improved algorithms for integer programming and related lattice problems. In D. Johnson, S. Fagin, M. Fredman, D. Harel, R. Karp, N. Lynch, C. Papadimitriou, R. Rivest, W. Ruzzo, and J. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 193–206. ACM, 1983.
27. G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In S. Severini and F. Brandão, editors, *8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013, May 21-23, 2013, Guelph, Canada*, volume 22 of *LIPIcs*, pages 20–34. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
28. T. Nagell. Über die Klassenzahl imaginär-quadratischer Zahlkörper. *Abh. Math. Sem. Univ. Hamburg*, 1:140–150, 1922.
29. National Institute of Standards and Technology. Post quantum cryptography project. `https://csrc.nist.gov/projects/post-quantum-cryptography`, 2018.
30. O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. arXiv:quant-ph/0406151.
31. C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, September 1994.
32. D. Shanks. Gauss's Ternary Form Reduction and the 2-Sylow Subgroup. *Mathematics of Computation*, 25(116):837–853, 1971.
33. J. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate texts in Mathematics*. Springer-Verlag, 1992.
34. A. Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. in Math. of Comm.*, 4(2):215–235, 2010.
35. A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Department of Computer Science, Swiss Federal Institute of Technology – ETH, 2000.
36. J. Tate. Endomoprhisms of abelian varieties over finite fields. *Inventiones Mathematica*, 2:134–144, 1966.
37. J. Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.