Weak instances of SIDH variants under improved torsion-point attacks

Péter Kutas[1], Chloe Martindale[2], Lorenz Panny[3], Christophe Petit[1], and Katherine E. Stange[4]

[1] School of Computer Science, University of Birmingham, UK
P.Kutas@bham.ac.uk, christophe.f.petit@gmail.com
[2] Department of Computer Science, University of Bristol, UK
chloe.martindale@bristol.ac.uk
[3] Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Netherlands
lorenz@yx7.cc
[4] Department of Mathematics, University of Colorado Boulder, Colorado, USA
kstange@math.colorado.edu

Abstract. SIDH is a post-quantum key exchange algorithm based on the presumed difficulty of computing isogenies between supersingular elliptic curves. However, the exact hardness assumption SIDH relies on is not the pure isogeny problem; attackers are also provided with the action of the secret isogeny restricted to a subgroup of the curve. Petit [21] leverages this information to break variants of SIDH in polynomial time, thus demonstrating that exploiting torsion-point information can lead to an attack in some cases. The contribution of this paper is twofold: First, we revisit and improve the techniques of [21] to span a broader range of parameters. Second, we construct SIDH variants designed to be weak against the resulting attacks; this includes weak choices of starting curve under moderately imbalanced parameters as well as weak choices of base field under balanced parameters. We stress that our results do not reveal any weakness in the NIST submission SIKE [19]. However, they do get closer than previous attacks in several ways and may have an impact on the security of SIDH-based group key exchange [2] and certain instantiations of B-SIDH [7].

1 Introduction

With the advent of quantum computers, currently deployed cryptographic protocols based on integer-factorization or discrete-logarithm problems will need to

* Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf. Péter Kutas and Christophe Petit were supported by EPSRC grant EP/S01361X/1. Katherine E. Stange was supported by NSF-CAREER CNS-1652238. This work was supported in part by the Commission of the European Communities through the Horizon 2020 program under project number 643161 (ECRYPT-NET) and in part by NWO project 651.002.004 (CHIST-ERA USEIT). Date of this document: 2020-05-27.

be replaced by new post-quantum cryptographic algorithms. Isogeny-based cryptography is a relatively new field within post-quantum cryptography. An *isogeny* is a non-zero rational map between elliptic curves that also preserves the group structure, and isogeny-based cryptography is based on the conjectured hardness of finding isogenies between elliptic curves over finite fields.

In recent years, isogeny-based cryptography has been receiving increased interest, partly due to the fact that isogeny-based key exchange has the smallest key sizes of all current post-quantum candidates while still performing at a reasonable speed. Isogeny-based schemes also appear to be fairly flexible; for example, a relatively efficient post-quantum non-interactive key exchange protocol called CSIDH [5] is built on isogeny assumptions.

The *Supersingular Isogeny Diffie–Hellman* protocol, or *SIDH*, was the first practical isogeny-based key-exchange protocol, proposed by Jao and De Feo in 2011 [14]. The most natural way to attack SIDH is to solve the following problem:

Problem 1. For a large prime p and smooth coprime integers A and B, given two supersingular elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E/\mathbb{F}_{p^2}$ connected by a degree-A isogeny $\phi\colon E_0 \to E$, and given the action of ϕ on the B-torsion of E_0, recover ϕ.[5]

Notice that this problem gives the attacker more information than the 'pure' isogeny problem, where the goal is to find an isogeny between two given curves without any further hints. The best known way to break SIDH by treating it as a pure isogeny problem is a claw-finding approach on the isogeny graph having both classical and quantum complexity $O(\sqrt{A} \cdot \mathrm{polylog}(p))$ [15].[6] However, Problem 1 could be easier than finding isogenies in general, and indeed a line of work started in [21] and continuing with this paper suggests that this may hold at least for some instantiations.

The additional torsion-point information clearly does aid *active* attackers: In 2016, Galbraith, Petit, Shani, and Ti [10] presented an active attack against SIDH that sends manipulated key-exchange messages and checks whether the key exchange succeeds, recovering the secret within $O(\log A)$ queries. To mitigate this attack, [10] proposes using the Fujisaki–Okamoto transform, which generically renders a CPA-secure public-key encryption scheme CCA-secure and therefore makes those so-called *reaction attacks* impossible. *Supersingular Isogeny Key Encapsulation*, or *SIKE* [13] for short, is essentially the result of applying (a variant of) the Fujisaki–Okamoto transform to SIDH. It is the only isogeny-based submission to NIST's standardization project for post-quantum cryptography [19] and is currently a contender in Round 2 of the process.

A particular choice made in SIKE is that one of the two curves, the 'starting curve' E_0, is a special curve: It is defined over $\mathbb{F}_p$ and has small-degree non-scalar endomorphisms, both of which are very rare properties within the set of all

[5] These constraints do not necessarily uniquely determine ϕ, but any efficiently computable isogeny from E_0 to E is usually enough to recover the SIDH secret [10].

[6] Note that the naïve meet-in-the-middle approach has prohibitively large memory requirements. Collision finding à la van Oorschot–Wiener thus performs better in practice, although its time complexity is worse in theory [1].

possible supersingular curves defined over $\mathbb{F}_{p^2}$. On its own, this fact does not seem to have any negative security implications for SIKE, but [10] shows that given an explicit description of *both* curves' endomorphism rings, it is (under reasonable heuristic assumptions) possible to recover the secret isogeny; hence, breaking SIKE can be reduced to computing endomorphism rings of supersingular elliptic curves in some sufficiently explicit representation.

In 2017, Petit [21] introduced a method to solve some instances of Problem 1 based on endomorphisms of the special starting curve. It uses the given action of the secret isogeny on a large torsion subgroup to recover the isogeny itself, giving a *passive* polynomial-time attack on non-standard variants of SIDH satisfying $B > A^4 > p^4$. However, for efficiency reasons, both A and B are practically taken to be about the size of $\sqrt{p}$; thus this attack does not apply to the SIKE parameters.

1.1 Our contributions

We improve upon and extend Petit's 2017 *torsion-point attacks* [21] in several ways. We argue in Section 3 that the imbalance conditions can be relaxed to $B > A^2 > p^2$ or $B > A^3 > p^{\frac{3}{2}}$, and that furthermore allowing for arbitrarily large B/A gives an attack for $AB \approx p$. We also show that even a mild imbalance of parameters leads to a heuristic improvement over the generic meet-in-the-middle or claw-finding attack, and we show the relationship between the extremity of the imbalance and the estimated complexity of the torsion-point attack.

Recall also that in SIKE, the starting curve E_0 is taken to be the curve with j-invariant 1728.[7] In Section 4 we introduce the notion of an 'insecure' starting curve, of which the curve with j-invariant 1728 is *not* an example. We give a heuristic polynomial-time torsion-point attack on SIDH instances using an insecure starting curve for E_0 with $B > A^2$ (note the lack of condition of p), and an attack of classical complexity $O(p^{\frac{2}{5}} \cdot \text{polylog}(p))$ and quantum complexity $O(p^{\frac{1}{8}} \cdot \text{polylog}(p))$ on SIDH instances using an insecure starting curve for E_0 with $B \approx A \approx p^{\frac{1}{2}}$. Note that this is as in SIKE, but starting from an insecure starting curve instead of the curve with j-invariant 1728; insecure curves could potentially be utilized as backdoors. We also give the relationship between the extremity of the imbalance of the parameters and the complexity of the torsion-point attack applied to this case of insecure starting curves, and argue that we expect there to be exponentially many insecure curves. Finally, we show that it is possible to construct special primes p, together with an appropriate A and B, for which torsion-point attacks are especially effective, even when using balanced parameters $A \approx B$ and/or using a starting curve with j-invariant 1728.

We stress that none of our attacks apply to the NIST Round 2 candidate SIKE: for each attack described in this paper one aspect of SIKE needs to be changed (e.g., the balance of the degrees of the secret isogenies, the starting curve, or the base field prime). There are, however, SIDH variants in the literature for which there are realistic parameter sets where our attacks may be more

[7] One can also take a neighbour, but this does not affect the security analysis.

relevant. For example, the recent n-party group key exchange proposal [2] can be interpreted for cryptanalysis purposes as an unbalanced (two-party) SIDH instance with $B \approx A^{n-1}$ and $AB \approx p$. The torsion-point attacks that we describe in Section 3, specifically with the balance given in Heuristic 1 Equation (10), give rise to, for example, a quantum attack of complexity $O(A^{0.4} \cdot \text{polylog}(p))$ on a 3-party key exchange with $AB \approx p^{1.15}$, a 10-party key exchange with $AB \approx p^{1.04}$, or a 100-party key exchange with $AB \approx p^{1.004}$. Furthermore, the attack variant that allows the protocol to use an insecure curve as a starting curve is heuristically classical polynomial-time for three or more parties. As a second example, we consider the recent proposal B-SIDH [7]; we show that this could in unlucky instances fall into the range in which torsion-point attacks are faster than meet-in-the-middle or claw finding. In the case of B-SIDH, it is definitely possible to use parameters for which our attacks do not apply, but care should be taken to do this.

Acknowledgements. We thank Victoria de Quehen for her invaluable input, especially for sharing her ideas from concurrent work and for her careful reading and advice during editing. Thanks to Daniel J. Bernstein for his help with Section 3.4, and to John Voight for answering a question concerning Section 4.3. We would also like to thank the anonymous reviewers for their useful feedback.

2 Preliminaries

2.1 Notation

Throughout this paper, we will neglect factors polynomial in $\log p$. Thus, we abbreviate $O(g \cdot \text{polylog}(p))$ as $O^*(g)$. Similarly, 'smooth' without further qualification always means polylog(p)-smooth. We let $\mathcal{B}_{p,\infty}$ denote the quaternion algebra ramified at p and ∞, for which we use a fixed $\mathbb{Q}$-basis $\langle 1, \mathbf{i}, \mathbf{j}, \mathbf{ij} \rangle$ such that $\mathbf{j}^2 = -p$ and $\mathbf{i}$ is a nonzero endomorphism of minimal norm with $\mathbf{ij} = -\mathbf{ji}$.

2.2 The Supersingular Isogeny Diffie–Hellman protocol

We give a high-level description of SIDH [14]. The public parameters of the system are two smooth coprime numbers A and B, a prime p of the form $p = ABf - 1$, where f is a small cofactor, and a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ together with points $P_A, Q_A, P_B, Q_B \in E_0$ such that $E_0[A] = \langle P_A, Q_A \rangle$ and $E_0[B] = \langle P_B, Q_B \rangle$.

The protocol then proceeds as follows:

1. Alice chooses a random cyclic subgroup of $E_0[A]$ as $G_A = \langle P_A + [x_A]Q_A \rangle$ and Bob chooses a random cyclic subgroup of $E_0[B]$ as $G_B = \langle P_B + [x_B]Q_B \rangle$.
2. Alice computes the isogeny $\phi_A : E_0 \to E_0/\langle G_A \rangle =: E_A$ and Bob computes the isogeny $\phi_B : E_0 \to E_0/\langle G_B \rangle =: E_B$.

3. Alice sends the curve E_A and the two points $\phi_A(P_B), \phi_A(Q_B)$ to Bob. Similarly, Bob sends $\big(E_B, \phi_B(P_A), \phi_B(Q_A)\big)$ to Alice.
4. Alice and Bob use the given torsion points to obtain the shared secret curve $E_0/\langle G_A, G_B\rangle$. To do so, Alice computes $\phi_B(G_A) = \phi_B(P_A) + [x_A]\phi_B(Q_A)$ and uses the fact that $E_0/\langle G_A, G_B\rangle \cong E_B/\langle \phi_B(G_A)\rangle$. Bob proceeds analogously.

The SIKE proposal [13] suggests various choices of (p, A, B) depending on the targeted security level: All parameter sets use powers of two and three for A and B, respectively, with $A \approx B$ and $f = 1$. For example, the smallest parameter set suggested in [13] uses $p = 2^{216} \cdot 3^{137} - 1$. Other constructions belonging to the SIDH 'family tree' of protocols use different types of parameters [7, 2, 23].

We may assume knowledge of $\mathrm{End}(E_0)$: The only known way to construct supersingular elliptic curves is by reduction of elliptic curves with CM by a small discriminant (which implies small-degree endomorphisms), or by isogeny walks starting from such curves (where knowledge of the path reveals the endomorphism ring, therefore requiring a trusted setup). A common choice when $p \equiv 3 \pmod 4$ is $j(E_0) = 1728$ or a small-degree isogeny neighbour of that curve [13].

2.3 Petit's torsion-point attacks

Most known attacks on SIDH proceed by solving the general isogeny problem, or reduce to computing endomorphism rings. However, SIDH is based on Problem 1 introduced above, in which an adversary also gets the action of the secret isogeny on the B-torsion of the starting curve E_0.

Remark 1. Problem 1 is a slight generalization of the Computational Supersingular Isogeny (CSSI) Problem introduced in [14]. Here we do not require A and B to be prime powers (just smooth) and we do not require p to have a special form. We remark that some instances of Problem 1 require superpolynomial space, as the extension fields required to represent $\ker\phi$ and $E_0[B]$ generally have degree at least linear in A and B. Broadly speaking, the interesting cases are the 'efficient' instantiations where computing ϕ and its action on $E_0[B]$ takes time and space polynomial in $\log p$, $\log A$, and $\log B$.

We outline Petit's approach [21] to solve some instances of Problem 1. The main steps are the following:

1. Compute a non-scalar $\theta \in \mathrm{End}(E_0)$ for which there exist $d, e \in \mathbb{Z}$ such that $\deg(\phi \circ \theta \circ \widehat{\phi} + [d]) = Be$ with e smooth and relatively small.
2. Compute $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$ using the fact that the action of ϕ is known on the B-torsion.
3. Compute $\ker(\tau - [d]) \cap E[A]$ and from that compute ϕ itself.

Notice that Step 1 can be done as precomputation as it only depends on E_0, but not on the particular public key under attack. (The degree of τ depends on the degree of ϕ but not on which degree-A isogeny ϕ happens to be.)

First we address Steps 2 and 3. In Step 2, τ can be decomposed into isogenies as $\eta \circ \psi$, where the degree of ψ is B and the degree of η is e. The isogeny ψ can

be computed since θ is known and we know the action of ϕ (and thus of $\widehat{\phi}$) on $E_0[B]$ (resp. $E[B]$). Then η can be found by meet-in-the-middle using $O^*(\sqrt{e})$ operations. In Step 3 we have $\ker(\widehat{\phi}) \subseteq \ker(\tau - [d]) \cap E[A]$, and in fact they are usually equal. They are not equal if and only if $\ker(\tau - [d])$ contains $E[M]$ for some divisor M of A; it is shown in [21, Section 4.3] how to resolve this issue.

The complexity of the algorithm clearly depends on the size of e, thus the efficiency of the algorithm is dependent on the effectiveness of Step 1. While the endomorphism ring of E_0 is usually known, it is not obvious how to find an element θ as above. For example, in SIKE, the starting curve has j-invariant 1728,[8] whose endomorphism ring is (up to small denominators) generated by the Frobenius $\pi : (x, y) \mapsto (x^p, y^p)$ and the automorphism $\iota : (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$, hence Step 1 reduces to solving the norm equation

$$A^2(pa^2 + pb^2 + c^2) + d^2 = Be; \tag{1}$$

the left-hand-side of this equation is just the degree of $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$ when $\theta = a\iota\pi + b\pi + c\iota$. Petit [21] gives an algorithm to solve Equation (1) when $A > p$ and $B > A^4$: The main idea is to choose e such that Be is a square modulo A^2, solve for d modulo A^2, and then solve for c modulo p. What remains is the equation $a^2 + b^2 = \frac{Be - d^2 - c^2A^2}{pA^2}$, which can be solved efficiently by Cornacchia's algorithm if the right-hand side is efficiently factorizable; else the procedure is restarted with a new choice of e. Under the conditions $A > p$ and $B > A^4$, this algorithm can be expected to find a suitable solution in polynomial time. This already suggests that there exist parameters for which Problem 1 is easier than the general supersingular isogeny problem.

3 Improved torsion-point attacks

In this section we generalize and improve the torsion-point attacks from Petit's 2017 paper [21]. Our setup is as in [21]: we study SIDH instances in which Alice and Bob use $E_0/\mathbb{F}_p\colon y^2 = x^3 + x$ as a starting curve, where p is a prime congruent to 3 (mod 4),[9] Alice's secret isogeny has degree $A \approx p^\alpha$, and Bob's secret isogeny has degree $B \approx p^\beta$. SIKE consists of such instances with $\alpha \approx \beta \approx 1/2$ and $\alpha + \beta < 1$, but in our analysis we will allow α and β to vary. The case $\alpha + \beta > 1$ may seem artificial to readers mostly familiar with traditional SIDH or SIKE [14, 12], but note that [21] and B-SIDH [7] propose cryptographically interesting variants of SIDH with such parameters; furthermore, studying such parameters helps to improve our understanding for the case $\alpha + \beta \approx 1$, cf. Figure 3. We assume without loss of generality that $A \leq B$ and that we are attacking Alice's key, i.e., the secret isogeny is of degree A and we are given the action of ϕ on the B-torsion of E_0.

[8] Note that the newest version of [13] changed the starting curve to a 2-isogenous neighbour, but this does not affect the asymptotic complexity of the (in fact, any) attack and thus we will stick with the original starting curve for simplicity.

[9] More generally, these attacks apply for any 'special' starting curve in the sense of [17].

Petit's 2017 classical, polynomial-time attack [21] requires unbalanced parameters, unlike those in SIKE, namely $\beta > 4\alpha > 4$. In this section, we argue that even a mild imbalance between α and β may result in a better (quantum) attack than the generic claw-finding/meet-in-the-middle algorithm, thus far considered to be the best attack for any parameters not broken by [21]. We also reduce the degree of imbalance needed for the polynomial-time algorithm to apply. Once more, we stress that these results are based on heuristic assumptions and ignore factors polynomial in $\log p$. The results of this section are summarized in Figures 1, 2, and 3 which will be justified by Heuristic 1. Note that Figures 1 and 2 represent a trade-off: The closer A and B are to each other, the bigger their product AB must be for the attacks to apply, and conversely reducing AB requires a stronger imbalance. Figure 3 shows that allowing for extremely unbalanced parameters $B \gg A$, we approach an attack on $AB \approx p$.

Our results suggest that the choice $\alpha \approx \beta \approx 1/2$ made in SIKE also minimizes the applicability of the torsion-point attack avenue. As we will show in Section 4, it is possible to improve on the meet-in-the-middle/claw-finding complexity for balanced parameters with a different starting curve, but with the SIKE starting curve it does not seem possible to get an attack via this method. However, since any unbalance can lead to a lower attack complexity, our results do affect other SIDH variants such as group key exchange [2]; see Figure 4.

Remark 2. A couple of notes on the choices made in Figures 1, 2, and 3:

- Algorithms with complexity polynomial in $\log p$ correspond to $\mathscr{C} = 0$.
- The complexity of the attack is measured as a power of A, the degree of Alice's secret isogeny. Together with our assumption that $A \leq B$, this allows for easy comparison with the 'generic attack', i.e., classical or quantum claw finding, both of which have complexity $O^*(A^{1/2})$ [15].

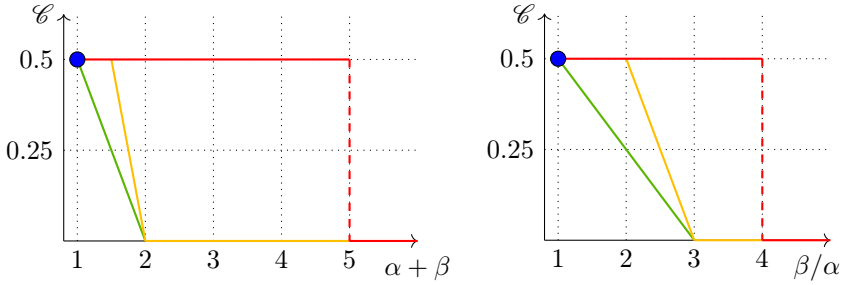

Figure 1. Attack complexity for $\alpha = 1/2$ and varying β, given by $O^*(A^{\mathscr{C}}) = O^*(p^{\mathscr{C}/2})$.
Red: Complexity of previous best known attack.
Yellow: Classical complexity of our attack, optimized for minimal $\alpha + \beta$ with $\alpha = 1/2$.
Green: Quantum complexity of our attack, optimized for minimal $\alpha + \beta$ with $\alpha = 1/2$.
Blue: SIKE parameters.

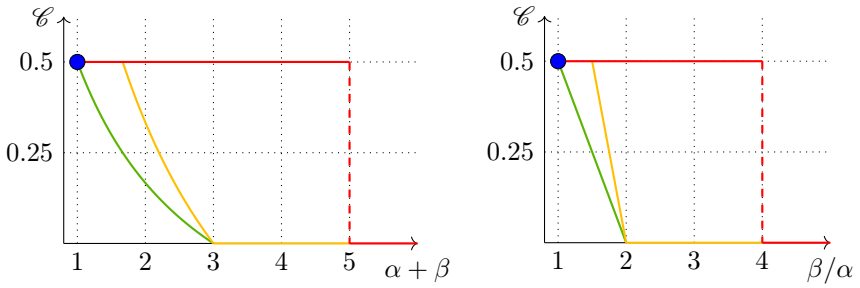

Figure 2. Attack complexity as α and β vary, given by $O^*(A^{\mathscr{C}}) = O^*(p^{\alpha \cdot \mathscr{C}})$.
Red: Complexity of previous best known attack.
Yellow: Classical complexity of our attack, optimized for minimal β/α.
Green: Quantum complexity of our attack, optimized for minimal β/α.
Blue: SIKE parameters.

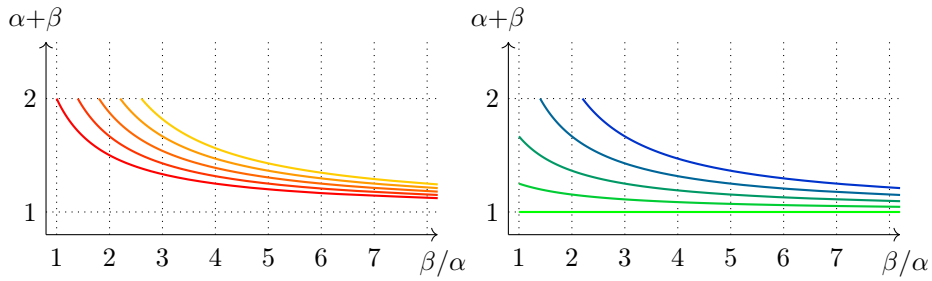

Figure 3. Possible choices of α, β for a classical attack (left) of complexity $O^*(A^{\mathscr{C}})$ for $\mathscr{C} = 0.5, 0.4, 0.3, 0.2, 0.1$ and a quantum attack (right) of complexity $O^*(A^{\mathscr{C}})$ for $\mathscr{C} = 0.5, 0.4, 0.3, 0.2, 0.1$, allowing α to be arbitrarily small.

3.1 Improved balance of the polynomial-time attack

Recall that in Figures 1 and 2 polynomial-time attacks correspond to the horizontal line at $\mathscr{C} = 0$: with this in mind, we see an improvement from a balance of $B > A^4 > p^4$ as in [21] to a balance of $B > A^3 > p^{\frac{3}{2}}$ or $B > A^2 > p^2$. This improvement comes from one simple trick, explained below.

Petit's attack solves Problem 1 (in which we want to compute the isogeny ϕ) by computing $\theta \in \mathrm{End}(E_0)$ and $a, b, c, d \in \mathbb{Z}$ such that there exists a small smooth integer e for which

$$A^2(pa^2 + pb^2 + c^2) + d^2 = \deg(\phi \circ \theta \circ \widehat{\phi} + [d]) = Be. \tag{2}$$

The restrictions $B > A^4$ and $A > p$ are to ensure that Petit's algorithm to find a solution (a, b, c, d, e) terminates in polynomial time and outputs a sufficiently small, smooth e.

We show in the following theorem that (2) can be relaxed to

$$A^2(pa^2 + pb^2 + c^2) + d^2 = \deg(\phi \circ \theta \circ \widehat{\phi} + [d]) = B^2e.$$

This in turn allows us to relax the balance of A and B to $B > A^3 > p^{\frac{3}{2}}$ or $B > A^2 > p^2$ to ensure that we find a sufficiently small solution (a, b, c, d, e), just by applying the same algorithm as Petit [21]. We do not repeat the algorithm here for conciseness, as we give a more general algorithm in the next section that also encompasses our non-polynomial time attacks; the balance of A and B is also addressed in the analysis in the following section.

Theorem 3. *Let A, B be coprime smooth integers. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let ϕ be a secret isogeny of degree A from E_0 to some curve E, and suppose that we are given the action of ϕ on $E_0[B]$. Furthermore, assume we are given a trace-zero endomorphism $\theta \in \mathrm{End}(E_0)$, an integer d coprime to B, and a smooth integer e such that*

$$\deg(\phi \circ \theta \circ \widehat{\phi} + [d]) = B^2 e.$$

Then we can compute ϕ in time $O^*(\sqrt{e})$.

Proof. Let $\tau = \phi \circ \theta \circ \widehat{\phi} + [d]$. Since the degree of τ is $B^2 e$, it can be decomposed as $\tau = \psi' \circ \eta \circ \psi$ where ψ and ψ' are isogenies of degree B and η is an isogeny of degree e. The isogeny ψ can be computed from the given action on the B-torsion as in Section 2.3.

To compute the isogeny ψ', we claim that $\ker(\widehat{\psi'})$ contains $\tau(E[B])$ with index dividing two. Thus, we can first evaluate τ on the B-torsion using the given action of θ, then find $\ker(\widehat{\psi'})$ by potentially brute-forcing a 2-isogeny, and finally compute ψ' from $\widehat{\psi'}$ and run the rest of the algorithm for each choice of ψ' the brute-force-of-η step yields.

We now prove the claim: First, $\widehat{\psi'} \circ \tau = [B] \circ \eta \circ \psi$ establishes that $\ker \widehat{\psi'} \supseteq \tau(E[B])$. We show that $\ker(\tau) \not\supseteq E[m]$ for any $m > 2$ dividing B. Suppose that τ decomposes as $\tau' \circ [m]$ for $\tau' \in \mathrm{End}(E)$, $m \in \mathbb{Z}$. Then m divides $\mathrm{trace}(\tau) = 2d$, but note that $\gcd(m, 2d) \mid 2$ since d was assumed coprime to B. Thus, the subgroup of $E[B]$ killed by τ is isomorphic to $\mathbb{Z}/B \times T$ with $T \leq \mathbb{Z}/2$, which proves $|\tau(E[B])| \in \{B, B/2\}$ and therefore $[\ker(\widehat{\psi'}) : \tau(E[B])] \mid 2$.

Finally, for each choice of ψ', we attempt to recover the isogeny η by a generic meet-in-the-middle algorithm, which runs in time $O^*(\sqrt{e})$ since e is smooth. Note that if $e \in O^*(1)$, then the entire algorithm runs in time $\mathrm{polylog}(p)$. □

For (a neighbour of) the initial curve used in SIKE [12] we deduce the following:

Corollary 4. *Let $p \equiv 3 \pmod 4$ and E_0 a curve with j-invariant 1728. Consider coprime smooth integers A, B and suppose that we are given an integer solution (a, b, c, d, e), with e smooth, to the equation*

$$A^2(pa^2 + pb^2 + c^2) + d^2 = B^2 e\,. \tag{3}$$

Then we can solve Problem 1 with the above parameters in time $O^*(\sqrt{e})$.

Proof. The left side of (3) is the norm form of an index-4 suborder of $\mathrm{End}(E_0)$.

3.2 Non-polynomial time torsion-point attacks

In this section we generalize Petit's polynomial-time attack to allow for any attacks with a better complexity than $O^*(A^{1/2})$, that is, attacks that improve upon the best known generic attack. Recall that $A = p^\alpha$ and $B = p^\beta$ are the degrees of Alice and Bob's secret isogenies respectively, and that we measure the complexity of the overall attack relative to A by writing it as $O^*(A^{\mathscr{C}})$. The attack, following the approach of Petit [21] together with the improvements described above, naturally splits into two stages: First, the 'precomputation' phase (Algorithm 1) in which a solution to (3) is computed — notably, this depends only on the parameters (p, A, B) and not on the concrete public key under attack. Second, the 'online' phase (Algorithm 2) in which we utilize said solution to recover the secret isogeny as in Theorem 3 for a specific public key. Our modifications to Petit's method come in three independent guises, and the resulting algorithm is shown in Algorithm 3:

– Precomputation phase

- **Larger *d*:** When computing a solution to Equation (3), we fix e and then try up to A^δ values for d until the equation has solutions. This allows us to further relax the constraints between A, B and p, at the price of an exhaustive search of cost $O^*(A^\delta) = O^*(p^{\alpha\delta})$.

– Online phase

- **Larger *e*:** We search for a solution to Equation (3) where e is any smooth number $\leq A^\epsilon$ with $\epsilon \in [0, 1]$, whereas in [21] the integer e was required to be polynomial in p. This relaxes the constraints on A and B, at a price of a $O^*(e^{\frac{1}{2}}) = O^*(p^{\frac{1}{2}\alpha\epsilon})$ computation (to retrieve the endomorphism η defined in the proof of Theorem 3).
- **Smaller *A*:** We first naïvely guess part of the secret isogeny and then apply Petit's techniques only on the remaining part for each guess. More precisely, we iterate through isogenies of degree $A^\gamma \mid A$, with $\gamma \in [0, 1]$, and for each possible guess we apply Petit's techniques on Problem 1 with $A' := A^{1-\gamma} = p^{\alpha(1-\gamma)}$ in place of A. The Diophantine equation to solve thus turns into

$$A'^2(pa^2 + pb^2 + c^2) + d^2 = B^2 e\,. \tag{4}$$

Algorithm 1 (Solving the norm equation; precomputation)

Input: SIDH parameters $p, A = p^\alpha, B = p^\beta$.
Input: Attack parameters $\delta, \gamma, \epsilon \in [0, 1]$, with $A^\gamma \mid A$.
Output: A solution (a, b, c, d, e) to (4) with $A' = A^{1-\gamma}$ and $e \leq A^\epsilon$ smooth.

Pick a smooth number $e \leq A^\epsilon$ which is a square modulo A'^2.
Compute d_0 as the smallest positive integer such that $d_0^2 \equiv eB^2 \pmod{A'^2}$.
for $d' = 1, 2, ..., \lfloor A^\delta \rfloor$ such that $d_0 + A'^2 d' < \sqrt{e}B$ **do**
    Let $d = d_0 + A'^2 d'$.
    Find the smallest positive integer c such that $c^2 A'^2 = eB^2 - d^2 \pmod{p}$, or **continue** if no such c exists.
    if $eB^2 > d^2 + c^2 A'^2$ **then**
        Try finding (a, b) such that $a^2 + b^2 = \frac{eB^2 - d^2 - c^2 A'^2}{A'^2 p}$.
        If a solution is found, **return** (a, b, c, d, e).

Algorithm 2 (Recovering the secret isogeny; online phase)

Input: All the inputs and corresponding output of Algorithm 1.
Input: An instance of Problem 1 with those parameters, namely a curve E and points $P, Q \in E[B]$ where there exists a degree-A isogeny $\varphi : E_0 \to E$ and P, Q are the images by φ of a canonical basis of $E_0[B]$.
Output: An isogeny φ matching the constraints given by the input.

for $\varphi_g : E \to E'$ an A^γ-isogeny **do**
    Compute $P' = [A^{-\gamma} \bmod B]\, \varphi_g(P)$ and $Q' = [A^{-\gamma} \bmod B]\, \varphi_g(Q)$.
    Use Theorem 3 to compute $\varphi' : E_0 \to E'$ of degree $A' = A^{1-\gamma}$, assuming that P' and Q' are the images by φ' of the canonical basis of $E_0[B]$, or conclude that no such isogeny exists.
    if φ' is found **then**
        return $\varphi = \widehat{\varphi_g} \circ \varphi'$

Algorithm 3 (Solving Problem 1)

Invoke Algorithm 1 and then Algorithm 2.

Heuristic 1. *For $\mathscr{C} \in [0, 0.5]$ Algorithm 3 can be expected to terminate successfully in time $O^*(A^{\mathscr{C}}) = O^*(p^{\alpha \cdot \mathscr{C}})$ when*

$$\alpha + \beta \geq 2 - \mathscr{C} \text{ and } \beta/\alpha \geq 3 - 2 \cdot \mathscr{C} \tag{5}$$

or

$$\alpha + \beta \geq \frac{3 - \mathscr{C}}{1 + \mathscr{C}} \text{ and } \beta/\alpha \geq 2 - \mathscr{C} \tag{6}$$

or

$$\alpha + \beta \approx 1 + 2\frac{1 - \mathscr{C}}{\beta/\alpha - 1 + 2\mathscr{C}} \tag{7}$$

when run on a classical computer, and

$$\alpha + \beta \geq 2 - 2 \cdot \mathscr{C} \text{ and } \beta/\alpha \geq 3 - 4 \cdot \mathscr{C} \tag{8}$$

or

$$\alpha + \beta \geq \frac{3 - 2 \cdot \mathscr{C}}{1 + 2 \cdot \mathscr{C}} \text{ and } \beta/\alpha \geq 2 - 2 \cdot \mathscr{C} \tag{9}$$

or

$$\alpha + \beta \approx 1 + 2\frac{1 - 2\mathscr{C}}{\beta/\alpha - 1 + 4\mathscr{C}} \tag{10}$$

when run on a quantum computer.

We will justify Heuristic 1 using Heuristic 2 and Lemma 5.

Heuristic 2. *We expect Steps 1 to 7 above to produce a solution to Equation (3) with A' instead of A if and only if*

$$2\beta + \alpha\epsilon \;>\; \max\left\{4\alpha - 4\alpha\gamma + 2\alpha\delta,\; 2 + 2\alpha - 2\alpha\delta - 2\alpha\gamma\right\}.$$

Justification. By construction we expect $d_0 \approx A'^2$, $d \approx A'^2 A^\delta \approx A^{2(1-\gamma)+\delta}$ and $eB^2 \approx A^\epsilon B^2$, so the 'for' loop in Algorithm 1 will run for A^δ iterations if

$$2\alpha(2(1 - \gamma) + \delta) < \alpha\epsilon + 2\beta.$$

The value c is then computed as a square root modulo p. We therefore expect $c \approx p$ most of the time, and $c \approx pA^{-\delta}$ with a probability $A^{-\delta}$, namely a constant number of times over all possible choices for d. For this particular c, we have $c^2 A'^2 \approx p^2 A^{-2\delta} A'^2 \approx p^{2-2\alpha\delta+2\alpha(1-\gamma)}$ and we expect to satisfy the second 'if' condition in Step 6 when

$$2 - 2\alpha\delta + 2\alpha(1 - \gamma) < \alpha\epsilon + 2\beta.$$

The two inequalities together give Heuristic 2.

Lemma 5. *Assume Heuristic 2 is satisfied.*

1. *The complexity of Algorithm 1 is $O^*(A^\delta)$ classically and $O^*(A^{\frac{\delta}{2}})$ quantumly.*

2. *The complexity of Algorithm 2 is* $O^*(A^{\gamma+\frac{1}{2}\epsilon})$ *classically and* $O^*(A^{\frac{1}{2}(\gamma+\epsilon)})$ *quantumly.*

Proof. The loop in Algorithm 1 has A^δ steps, each with polynomial complexity (use Cornacchia for Step 9). Quantumly this search takes a square root of the classical cost. The loop in Algorithm 2 has approximately A^γ steps, and the main cost in each step is a meet-in-the-middle attack to recover an isogeny of smooth degree $e \approx A^\epsilon$. On a classical computer the cost is approximately $O^*(A^\gamma e^{\frac{1}{2}}) = O^*(A^{\gamma+\frac{1}{2}\epsilon})$. Using quantum search to guess the correct degree-A^γ isogeny φ_g and claw-finding routines for brute-forcing the degree-e isogeny in Step 13 (assuming only square-root complexity for the latter [15]), Algorithm 2 has quantum complexity $O^*(A^{\frac{1}{2}\gamma}e^{\frac{1}{2}}) = O^*(A^{\frac{1}{2}(\gamma+\epsilon)})$. □

Justification of Heuristic 1. Set $\mathscr{C} = \delta/2$. To get (5), plug in $\alpha = 1/2$, $\gamma = 0$, and $\epsilon = 2\delta$ into Heuristic 2 and Lemma 5. To get (6), plug in $\alpha \approx \frac{1}{1+\delta}$, $\gamma = \delta$, $\epsilon = 0$. To get (8), plug in $\alpha = 1/2$, $\gamma = \delta$, and $\epsilon = 0$. To get (9), plug in $\alpha \approx \frac{1}{1+\delta}$, $\gamma = \delta$, and $\epsilon = 0$. To get (7) and (10), plug in $\alpha < \frac{1}{1+\delta}$, $\gamma = \delta$, $\epsilon = 0$, and $\beta \approx 1 + \alpha(1-2\delta)$.

3.3 Impact on variants of SIDH

The group key exchange protocol in [2] with k parties can be broken by solving an instance of Problem 1 with $A \approx p^{\frac{1}{k}}$ and $B \approx p^{\frac{k-1}{k}}$. Although our attacks only apply for $AB > p$, from Figure 3, or equivalently Heuristic 1 Equations (7) and (10), we see that as the imbalance increases, the attack applies for AB approaching p. In particular, for a large number of parties k, the product AB does not have to be much larger than p for an (exponential) torsion-point attack to apply.

Recently, Costello [7] suggested to use parameters such that AB is a large divisor of p^2-1, and potentially as large as p^2-1, for his B-SIDH scheme. Suppose that parameters A and B are chosen with $AB \approx p^2$, then a mild imbalance can lead to an attack: For example, by Heuristic 1 Equation (9), if $B > A^{\frac{5}{3}}$ then we have a quantum attack of complexity $O^*(A^{\frac{1}{6}})$. See Figure 4 for an image of the parameters in which torsion-point attacks apply to this scheme. There are many parameter sets for B-SIDH in which the user is safe from our torsion-point attacks, so all that is necessary is for care to be taken to avoid such imbalanced cases.

Our attacks apply to, at best, edge cases of each scheme: in group key exchange typically $AB < p$ (unless an unorthodox choice is made) and in B-SIDH typically $A \approx B$ (unless an unorthodox choice is made). However, it is not infeasible that someone implementing a group key exchange protocol may borrow ideas from B-SIDH in order to improve the efficiency for certain parties in the group key exchange, especially given the scarcity of appropriate base field primes for group key exchange following [2]. Such a combined group key exchange with ideas from B-SIDH could easily yield a torsion-point attack: For example, even for 3 parties, parameters with $AB \approx p^2$ lead to an quantum attack of complexity $O^*(A^{1/8})$, a fourth-root improvement over the generic attack.

3.4 Improvement prospects

In this section we consider how future improvements on the resolution of Equation (3) might impact the hardness of Problem 1. We first estimate the minimal size of e for a given set of parameters (p, A, B).

Heuristic 3. *Solutions (a, b, c, d, e) to Equation (3) can be expected to satisfy*

$$e^2 \geq \tfrac{A^3 p}{B^2}\,.$$

Justification. We consider solutions with $e \leq M$ for some fixed bound M. Since all summands on the left-hand side are non-negative, they cannot be bigger than the upper bound MB^2 of the right-hand side. This yields the bounds

$$a \leq \tfrac{\sqrt{M}B}{A\sqrt{p}}\,; \qquad b \leq \tfrac{\sqrt{M}B}{A\sqrt{p}}\,; \qquad c \leq \tfrac{\sqrt{M}B}{A}\,; \qquad d \leq \sqrt{M}B\,.$$

Hence the number of possible assignments of the variables e, a, b, c, d is about

$$M \cdot \tfrac{\sqrt{M}B}{A\sqrt{p}} \cdot \tfrac{\sqrt{M}B}{A\sqrt{p}} \cdot \tfrac{\sqrt{M}B}{A} \cdot \sqrt{M}B = \tfrac{M^3B^4}{A^3p}.$$

Heuristically modeling both left- and right-hand side as uniformly random in the range $\{0, \ldots, MB^2\}$, this implies the expected number of solutions is about

$$\tfrac{M^3B^4}{A^3p} / (MB^2) = \tfrac{M^2B^2}{A^3p}\,.$$

Solving this for one expected solution yields the claimed estimate.

Heuristic 4. *Assume that we are given a solution to Equation 3 for parameters as in Heuristic 3. Then we expect to solve Problem 1*

1. *in classical time $O^*(1)$ when $B > p^{\frac{1}{2}} A^{\frac{3}{2}}$,*
2. *with classical complexity $O^*(A^{\frac{1}{2}})$ when $B > p^{\frac{1}{2}} A^{\frac{1}{2}}$, and*
3. *with quantum complexity $O^*(A^{\frac{1}{2}})$ when $B > p^{\frac{1}{2}}$.*

Justification. As we are given a solution to Equation (3), we no longer need the precomputation steps of Algorithm 3. Heuristic 3 gives the constraint

$$2(\beta + \alpha\epsilon) \geq 1 + 3\alpha(1 - \gamma), \tag{11}$$

which we now use in place of Heuristic 2 in our analysis to optimally balance parameters.

1. For polynomial-time attacks we take $\epsilon = \gamma = 0$. Plugging into Inequality (11) gives $2\beta > 1 + 3\alpha$, hence the result.
2. Increasing either γ or ϵ will contribute to relaxing Inequality (11), and by Lemma 5 we need $\gamma + \frac{1}{2}\epsilon < \frac{1}{2}$. Substituting γ for $\frac{1}{2}(1 - \epsilon)$ in (11), gives

   $$2\beta > 1 + \alpha\left(\tfrac{3}{2} - \tfrac{\epsilon}{2}\right).$$

   Taking $\epsilon \approx 1$, $\gamma = 0$ this simplifies to $2\beta > 1 + \alpha$, hence the result.

3. Increasing either γ or ϵ will contribute to relaxing Inequality (11), and by Lemma 5 we need $\gamma + \epsilon < 1$. Substituting γ for $1 - \epsilon$ in (11), we get

$$2\beta > 1 + \alpha\epsilon.$$

Taking $\epsilon = 0$, $\gamma \approx \frac{1}{2}$ this simplifies to $2\beta > 1$, hence the result. □

Remark 6. In the group key exchange protocol [2] with k parties we have $A \approx p^{\frac{1}{k}}$ and $B \approx p^{\frac{k-1}{k}}$. A better solver for Equation (3) could give a quantum attack when $k > 2$, a classical attack when $k > 3$, and a (classical) polynomial-time attack when $k > 5$.

Remark 7. In contexts where several instances of Problem 1 need to be solved with the same parameters, Algorithm 1 only needs to be executed once. In this case the algorithm's parameters can be tweaked to reduce the average cost per instance.

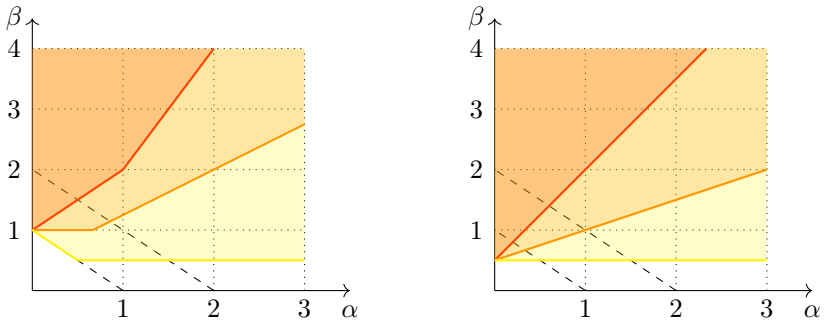

Figure 4. Performance of our current attacks (left) and hypothetical attacks assuming a polynomial-time solver for Equation 3 (right). Here $A \approx p^{\alpha}$ and $B \approx p^{\beta}$. Parameters (α, β) above the red, orange and yellow curves are parameters admitting a polynomial-time attack, an improvement over the best classical attacks, and an improvement over the best quantum attacks respectively. Parameters below the dashed lines are those allowing $AB \mid (p-1)$ and $AB \mid (p^2-1)$ as in [12, 13, 7].

4 Insecure instances

In this section we give various new instances for which we can solve Problem 1. Recall that we denote by $A \leq B$ the degrees of Alice's and Bob's secret isogenies respectively, and we set $A \approx p^{\alpha}$ and $B \approx p^{\beta}$. For all the instances studied in Section 3, for our attack methods to give an improvement on the complexity of meet-in-the-middle we need $AB > p$ (see Figures 1, 2, 3, or 4). Furthermore, we

only expect solutions to Equation 3 with polynomially small e[10] when $B > A^3 > p^{\frac{3}{2}}$ or $B > A^2 > p^2$. However, so far we have only considered cases where the starting curve has j-invariant 1728. In Section 4.1 we explore the question: for given A, B do there exist other starting curves for which we can solve Problem 1 with a better balance? We will call such curves insecure curves (which we make more precise in Definition 8), and quantify the number of insecure curves in Section 4.3.

In Sections 4.4 and 4.5, we also consider insecure choices for p, A, and B for which we can solve Problem 1 starting from the curve with j-invariant 1728.

4.1 Insecure curves

This section introduces the concept of *insecure* curves and how to find such curves. Roughly speaking, these are curves which, if used as starting curves for the SIDH protocol, would be susceptible to an attack utilizing the given action on torsion points under only moderately imbalanced parameters A, B; in particular, an imbalance which is independent of p. In fact, when we allow for non-polynomial time attacks we get an asymptotic improvement on the best known attack for balanced, SIDH parameters (starting of course from an insecure curve, unlike in SIKE). If this attack was not known, these curves could have been utilized as *backdoor* curves, for example by suggesting the use of such a curve as a standardized starting curve. The notion of insecure curves is dependent on the parameters A, B, which motivates the following definition:

Definition 8. *Let p be a prime number and A, B be coprime positive integers. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Then we say that E_0 is $(A, B, \mathscr{C})$-insecure if for any secret isogeny $\phi : E_0 \to E$ of degree A we can compute ϕ in time $O^*(A^{\mathscr{C}})$ given (E, A, B) and the action of ϕ on $E_0[B]$. If we can compute ϕ in polynomial time, then E_0 is called (A, B)-insecure.*

We summarize the complexity of our attack on SIDH instances starting at insecure curves in Figure 5; this figure follows from Theorem 16. Note that these attacks do apply to balanced parameters with $AB \approx p$ and give a significant improvement on the meet-in-the-middle claw-finding complexity for these cases. We stress however that this relies on using a weak starting curve and hence does not give an attack on SIKE when using the proposed (neighbour of a) starting curve with j-invariant 1728, unless there happens to be short path from this starting curve to an insecure curve.

Algorithm 4 computes (A, B)-insecure curves in heuristic polynomial time, under the assumption that we have a factoring oracle (see Theorem 9).

Theorem 9. *For every A, B with the property that $B > A^2$, given an oracle for factoring, Algorithm 4 succeeds in heuristic polynomial time. Furthermore, the representation of θ is efficient, i.e., one can evaluate θ on any point of the curve in polynomial time.*

[10] Recall that this is necessary to obtain a polynomial-time online cost in our attack.

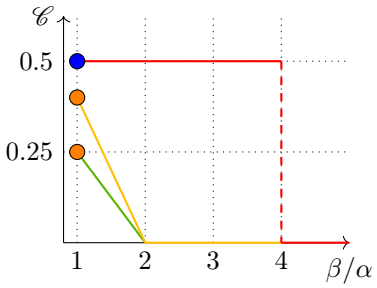

Figure 5. Balance of $A = p^\alpha$ and $B = p^\beta$ for which we can find a $(A, B, \mathscr{C})$-insecure curve E_0 in time $O^*(A^{\mathscr{C}})$. An SIDH instance starting at E_0 has security $O^*(A^{\mathscr{C}})$ against our attack.
Red: Complexity of previous best known attack.
Yellow: Classical complexity of our attack when starting at a $(A, B, \mathscr{C})$-insecure curve.
Green: Quantum complexity of our attack when starting at a $(A, B, \mathscr{C})$-insecure curve.
Blue: SIKE parameters.
Orange: SIKE-like parameters, but starting instead from an $(A, B, \mathscr{C})$-insecure curve.

Algorithm 4 Computing (A, B)-insecure elliptic curves

Input: A prime $p \equiv 3 \pmod 4$ and smooth coprime integers A, B with $B > A^2$.
Output: A supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ which is (A, B)-insecure, and θ, d, e that satisfy the conditions of Theorem 3.

Set $e := 1$.
Find an integer d such that $d^2 \equiv B^2 e \pmod{A^2}$.
if d is not coprime to B **then**
    Set e to the next square and go to Step 2.
if $\frac{B^2e-d^2}{A^2}$ is square modulo p **then**
    Find rational a, b, c such that $pa^2 + pb^2 + c^2 = \frac{B^2e-d^2}{A^2}$.
else
    Set e to the next square and go to Step 2.
Set $\theta = a\mathbf{ij} + b\mathbf{j} + c\mathbf{i}$.
Compute a maximal order $\mathcal{O} \subseteq \mathcal{B}_{p,\infty}$ containing θ.
Compute an elliptic curve E_0 whose endomorphism ring is isomorphic to $\mathcal{O}$.
return $j(E_0)$, θ, d, e.

Remark 10. It is important that θ has an efficient representation as it might not have a smooth degree.

Remark 11. This (im)balance is sufficient to break (in polynomial time) a variant of the group key exchange protocol [2] for three or more parties starting at any (A, B)-insecure curve (this does not break the protocol proposed in [2] as there the starting curve has j-invariant 1728).

Before proving Theorem 9 we need the following easy lemma:

Lemma 12. *Let p be a prime congruent to 3 modulo 4. Let D be a positive integer. Then the quadratic form $Q(x_1, x_2, x_3, x_4) = px_1^2 + px_2^2 + x_3^2 - Dx_4^2$ has a nontrivial integer zero if and only if D is a quadratic residue modulo p.*

Proof. The proof is essentially a special case of [25, Proposition 10] but we give a brief sketch of the proof here. If D is a quadratic residue modulo p, then $px_1^2 + px_2^2 + x_3^2 - Dx_4^2$ has a solution in $\mathbb{Q}_p$ by setting $x_1 = x_2 = 0$, $x_4 = 1$ and applying Hensel's lemma to the equation $x_3^2 = D$. The quadratic form Q also has local solutions everywhere else (the 2-adic case involves looking at the equation mod 8 and applying a 2-adic version of Hensel's lemma). If D is not a quadratic residue modulo p then one has to choose x_3 and x_4 to be divisible by p. When solving the equation $Q(x_1, x_2, x_3, x_4)$ one divides by p, then the equation mod p reduces to $x_1^2 + x_2^2 q \equiv 0 \pmod{p}$. This does not have a solution as p is congruent to 3 modulo 4. Finally, one can show that this implies that Q does not have a zero in $\mathbb{Q}_p$. □

Proof (of Theorem 9). The main idea is to apply Theorem 3 in the following way: using Algorithm 4, we find integers D, d, and e, with e polynomially small and D a quadratic residue mod p, such that $A^2 D + d^2 = B^2 e$, and an element $\theta \in \mathcal{B}_{p,\infty}$ of trace zero and such that $\theta^2 = -D$. We then construct a maximal order $\mathcal{O} \subseteq \mathcal{B}_{p,\infty}$ containing θ and an elliptic curve E_0 with $\mathrm{End}(E_0) \cong \mathcal{O}$.

Most steps of Algorithm 4 obviously run in polynomial time, although some need further explanation. We expect $d^2 \approx A^4$ since we solved for d modulo B^2, and we expect e to be small since heuristically we find a quadratic residue after a small number of tries. Then the right-hand-side in Step 6 should be positive since $B > A^2$, so by Lemma 12 step 6 returns a solution using Simon's algorithm [25], assuming an oracle for factoring $\frac{B^2 e - d^2}{A^2}$. For Step 10, we can apply either of the polynomial time algorithms [11, 26] for finding maximal orders containing a fixed order in a quaternion algebra, which again assume a factoring oracle. Step 11 can be accomplished using the heuristically polynomial time algorithm from [22, 9] which returns both E_0 and an efficient representation of θ — the endomorphism θ has large degree, so its representation as a rational function is large, but it can be represented compactly as a linear combination of smooth-degree endomorphisms. □

Remark 13. The algorithm uses factorization twice. In Section B we discuss how one can ensure in practice that the numbers to be factored have an easy factorization.

Remark 14. Note that Theorem 9 is not an if-and-only-if statement. In particular, there might exist A and B and an elliptic curve E_0 which is (A, B)-insecure even if the above inequalities are not satisfied.

Remark 15. Weak curves also have a constructive application: An improvement on the recent paper [8] using Petit's attack to build a one-way function 'SÉTA'. In this scheme, the secret key is a secret isogeny to a curve E_s that starts from the elliptic curve with j-invariant 1728 and the message is the end point of a

secret isogeny from E_s to some curve E_m, together with the image of some torsion points. The reason for using j-invariant 1728 is in order to apply Petit's attack constructively. One could instead use a weak curve; this provides more flexibility to the scheme as one does not need to disclose the starting curve and the corresponding norm equation is easier to solve.

4.2 Non-polynomial time attacks for insecure curves

In this section we give a further generalization of Algorithm 3 to utilize some extra techniques available to us when the starting curve E_0 is an insecure curve. Recall, as above, that $A \leq B$ are the degrees of Alice's and Bob's secret isogenies respectively, and $A \approx p^{\alpha}$ and $B \approx p^{\beta}$. Recall the definition of an $(A, B, \mathscr{C})$-insecure curve E_0 from Definition 8; in particular that such a curve gives rise to a torsion point attack of complexity $O^*(A^{\mathscr{C}})$.

We show in this section that for $\alpha \approx \beta$, we can modify Algorithm 4 to compute a classically $(A, B, \frac{2}{5})$-insecure curve or a quantumly $(A, B, \frac{1}{4})$-insecure curve. We also show how the attack on insecure curves improves for imbalanced parameters; see Figure 5 for a comparison of previous results with Theorem 16.

Theorem 16. *Following our notation convention, we write $O^*(A^0)$ time for (heuristic) polynomial time and similarly $(A, B, \approx 0)$-insecure curve for (A, B)-insecure curve.*

- *Let $\mathscr{C} \in [0, 0.4]$. For every A, B such that $B > A^{2-\frac{5}{2}\cdot\mathscr{C}}$, a classical adversary can compute a $(A, B, \mathscr{C})$-insecure curve in time $O^*(A^{\mathscr{C}})$, assuming an oracle for factoring.*
- *Let $\mathscr{C} \in [0, 0.25]$. For every A, B such that $B > A^{2-4\cdot\mathscr{C}}$, a quantum adversary can compute a $(A, B, \mathscr{C})$-insecure curve in heuristic polynomial time.*

Proof. Modify Algorithm 4 as follows:

- Assume that we will guess part of the isogeny with degree $A^{\gamma} \mid A$, and use $A' = A^{1-\gamma}$ instead of A.
- Instead of starting from $e = 1$, start the loop at e such that $B^2 e > A'^4$.
- Choose $A^{\epsilon'}$ random values of $e \leq A^{\epsilon}$ (note e is not necessarily an integer square) until there exists d such that $d^2 = B^2 e \bmod (A')^2$,

$$B^2 e - d^2 > 0, \tag{12}$$

  and $B^2 e - d^2$ is a square modulo p. Once these values of d and e are found, continue like in Algorithm 4, Step 6.

The attacker can now follow Algorithm 2 to compute the secret isogeny, using the endomorphism θ from Algorithm 4 for the necessary precomputed values.

We now analyze the complexity of running the modified Algorithm 4 followed by Algorithm 2. The two quadratic residuosity conditions are heuristically satisfied one in four times, so we ignore them in this analysis. The cost of Algorithm 4

modified in this way becomes $O^*(A^{\epsilon'})$ for a classical adversary and $O^*(A^{\frac{\epsilon'}{2}})$ for a quantum adversary.

Note also that by construction we have $e \leq A^{\epsilon}$, so the cost of running Algorithm 2 will be $O^*(A^{\gamma+\frac{\epsilon}{2}})$ for a classical adversary and $O^*(A^{\frac{\gamma+\epsilon}{2}})$ for a quantum adversary, following the same reasoning as in the complexity analysis of Algorithm 3.

We now look at the conditions for existence of a solution in Algorithm 4. Note that d is a priori bounded by $(A')^2 = A^{2(1-\gamma)}$. However, after trying A^{ϵ} values for e we may hope to find some d bounded by $A^{2(1-\gamma)-\epsilon}$. To satisfy (12) we need

$$2\beta > \alpha(4 - 4\gamma - 2\epsilon' - \epsilon),$$

and by construction we also need $\epsilon' \leq \epsilon$.

For a classical adversary, setting $\epsilon = \epsilon' = 2\gamma = \mathscr{C}$ gives the result. For a quantum adversary, setting $\epsilon = \epsilon = 0$ and $\gamma = 2 \cdot \mathscr{C}$ gives the result. □

Remark 17. We found these choices for $\epsilon, \epsilon', \gamma$ by solving the following optimization problems for $\alpha = \beta = \frac{1}{2}$, so at least in that case (which corresponds to SIKE) we expect there to be no better choice with respect to overall complexity: For the best classical attack when $\alpha = \beta = \frac{1}{2}$ we solved the following optimization problem:[11]

$$\min_{\substack{4\gamma+2\epsilon'+\epsilon\geq 2,\\ \epsilon\geq\epsilon'}} \max\left\{\epsilon', \gamma + \frac{\epsilon}{2}\right\}.$$

For the best quantum attack when $\alpha = \beta = \frac{1}{2}$ we solved the following optimization problem:

$$\min_{\substack{4\gamma+2\epsilon'+\epsilon\geq 2\\ \epsilon\geq\epsilon'}} \max\left\{\frac{\epsilon'}{2}, \frac{\gamma+\epsilon}{2}\right\}.$$

4.3 Counting insecure curves

Having shown how to find insecure curves and how to exploit them, a natural question to ask is how many of these curves we can construct using the methods of the previous section. Recall that the methods above search for an element $\theta \in \mathcal{B}_{p,\infty}$ with reduced norm D. Theorem 18, due to Onuki [20], suggests they can be expected to produce exponentially (in $\log D$) many different maximal orders, and using Lemma 19 we prove this rigorously for the interesting case of (A, B)-insecure curves with $AB \approx p$ and $A^2 < B < A^3$ (cf. Theorem 9).

We first recall some notation from [20]. The set $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$ consists of the reductions modulo p of all elliptic curves over $\overline{\mathbb{Q}}$ with complex multiplication by $\mathcal{O}$. Each curve $E = \mathcal{E} \bmod p$ in this set comes with an optimal embedding $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$, referred to as an 'orientation' of E, and conversely, [20, Prop. 3.3] shows that — up to conjugation — each oriented curve (E, ι) defined

[11] This can be done for example using https://online-optimizer.appspot.com/.

over $\overline{\mathbb{F}}_p$ is obtained by the reduction modulo p of a characteristic-zero curve; in other words, either (E, ι) or $(E^{(p)}, \iota^{(p)})$ lies in $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$. Onuki proves:

Theorem 18 ([20, Theorem 3.4]). *Let K be an imaginary quadratic field such that p does not split in K, and $\mathcal{O}$ an order in K such that p does not divide the conductor of $\mathcal{O}$. Then the ideal class group $\mathrm{cl}(\mathcal{O})$ acts freely and transitively on $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$.*

Thus, it follows from well-known results about imaginary quadratic class numbers [24] that asymptotically, there are $h(-D) \in \Omega(D^{1/2-\varepsilon})$ many insecure elliptic curves *counted with multiplicities* given by the number of embeddings of $\mathcal{O}$. However, it is not generally clear that this corresponds to many distinct *curves* (or maximal orders). As an (extreme) indication of what could go wrong, consider the following: there seems to be no obvious reason why in some cases the entire orbit of the group action of Theorem 18 should *not* consist only of one elliptic curve with lots of independent copies of $\mathcal{O}$ in its endomorphism ring.

We can however at least prove that this doesn't always happen. In fact, in the case that D is small enough relative to p, one can show that there cannot be more than one embedding of $\mathcal{O}$ into any maximal order in $B_{p,\infty}$, implying that the $h(-D)$ oriented supersingular elliptic curves indeed must constitute $h(-D) \approx \sqrt{D}$ distinct quaternion maximal orders:

Lemma 19. *Let $\mathcal{O}$ be a maximal order in $\mathcal{B}_{p,\infty}$. If $D \equiv 3, 0 \pmod 4$ is a positive integer smaller than p, then there exists at most one copy of the imaginary quadratic order of discriminant $-D$ inside $\mathcal{O}$.*

Proof. This follows readily from Theorem 2' of [16].

This lemma together with Theorem 9 suffice to prove that there are $\Theta(h(-D))$ many (A, B)-insecure maximal orders under the restrictions that $B > A^2$ and $D < p$. Consider the case (of interest) in which $AB \approx p$: Following the same line of reasoning as in the proof of Theorem 9 we have that $B^2/A^2 - A^2 \approx D$, which if $D < p \approx AB$ implies that $B \lessapprox A^3$. Hence, as advertised above, Lemma 19 suffices to prove that there are $\Theta(h(-D))$ many (A, B)-insecure maximal orders under the restriction that $AB \approx p$ and roughly $A^2 < B < A^3$. For larger choices of B, it is no longer true that there is only one embedding of $\mathcal{O}$ into a quaternion maximal order: indeed, at some point $h(-D)$ will exceed the number $\Theta(p)$ of available maximal orders, hence there must be repetitions. While it seems hard to imagine cases where the orbit of $\mathrm{cl}(\mathbb{Z}[\theta])$ covers only a negligible number of curves (recall that θ was our endomorphism of reduced norm D), we do not currently know how (and under which conditions) to rule out this possibility.

Remark 20. Having obtained any one maximal order $\mathfrak{O}$ that contains θ, it is efficient to compute more such orders (either randomly or exhaustively): For any ideal $\mathfrak{a}$ in $\mathbb{Z}[\theta]$, another maximal order with an optimal embedding of $\mathbb{Z}[\theta]$ is the right order of the left ideal $\mathcal{I} = \mathfrak{O}\mathfrak{a}$. (One way to see this: $\mathfrak{a}$ defines a horizontal isogeny with respect to the subring $\mathcal{O}$; multiplying by the full endomorphism ring does not change the represented kernel subgroup; the codomain of an isogeny

described by a quaternion left ideal has endomorphism ring isomorphic to the right order of that ideal. Note that this is similar to a technique used by [6] in the context $\mathcal{O} \subseteq \mathbb{Q}(\pi)$.)

4.4 Insecure p for given A and B with starting vertex 1728

Another way of constructing insecure instances of an SIDH-style key exchange is to keep the starting vertex as $j = 1728$ (or close to it), keep A and B smooth or powersmooth but not necessarily only powers of 2 and 3 (as in SIKE), and construct the base field prime p to make $j = 1728$ into an (A, B)-insecure curve.

An easy way of constructing such a p is to perform Steps 1 and 2 of Algorithm 4, and then let $D := \frac{B^2 e - d^2}{A^2}$. Allowing p to be a variable, we can solve

$$D = p(a^2 + b^2) + c^2$$

in variables $a, b, c, p \in \mathbb{Z}$, p prime, as follows. Factor $D - c^2$ for small c until the result is of the form pm where p is a large prime congruent to 3 modulo 4 and m is a number representable as a sum of squares.[12]

Then $\theta = aj + bk + ci$ is in the endomorphism ring of the curve with $j = 1728$, hence this curve is (A, B)-insecure for such primes p. (Note that, in this construction, we cannot expect to preserve a relationship such as $p = ABf - 1$, for some small $f \in \mathbb{Z}$.)

As an (unbalanced) example, let us choose $A = 2^{216}$ and $B = 3^{300}$ and $e = 1$. Then we can choose d to be B modulo A^2. Let $D = \frac{B^2 - d^2}{A^2}$, and now produce two primes. First we choose $c = 53$, then $D - c^2$ is a prime number (i.e., $a = 1,\ b = 0$). For the second instance let $c = 355$, then $D - c^2$ is 5 times a prime number (i.e., $a = 2,\ b = 1$). Both of these primes are congruent to 3 modulo 4.

For a powersmooth example, choose A to be the product of every second prime from 3 up to 317, and B to be the product of the other odd primes ≤ 479. With $e = 4$, we can choose d to equal B modulo A^2 and D as described. Then $D - 153^2$ is a prime congruent to 3 modulo 4 (i.e. $a = 1$, $b = 0$).

4.5 Insecure $A \approx B$ for $j = 1728$

For $A \approx B$, it seems difficult to find (A, B)-insecure curves. However, in this section we show that certain choices of (power)smooth parameters A and B allow us to find f such that $j = 1728$ is insecure over $\mathbb{F}_p$ with $p = ABf - 1$.

One approach to this is to find Pythagorean triples $A^2 + d^2 = B^2$ where A and B are coprime and (power)smooth; these are insecure in the sense of the previous sections, with $e = c = 1, a = b = 0$ and $j = 1728$. With this construction, we can then use *any* $p \equiv 3 \pmod 4$, in particular one of the form $p = ABf - 1$.

Problem 2. Find Pythagorean triples $B^2 = A^2 + d^2$ such that A and B are coprime and smooth (or powersmooth).

[12] Some choices of A and B result in $D \equiv 2 \pmod 4$ which is an obstruction to this method.

Pythagorean triples can be parameterized in terms of Gaussian integers. To be precise, primitive integral Pythagorean triples $a^2 = b^2 + c^2$ are in bijection with the Gaussian integers $z = m + ni$ with $\gcd(m, n) = 1$ via the correspondence $(a, b, c) = (\mathrm{N}(z), \mathrm{Re}(z^2), \mathrm{Im}(z^2))$. The condition that m and n are coprime is satisfied if we take z to be a product of split Gaussian primes, i.e. $z = \prod_i w_i$ where $\mathrm{N}(w) \equiv 1 \pmod 4$ is prime, taking care to avoid simultaneously including a prime and its conjugate. Thus the following method applies provided that B is taken to be an integer divisible only by primes congruent to 1 modulo 4, and $B > A$.

In order to guarantee that $B = \mathrm{N}(z)$ is powersmooth, one may take many small w_i. In order to guarantee that B is smooth, it is convenient to take $z = w^k$ for a single small Gaussian prime w, and a large composite power k.

It so happens that the sequence of polynomials $\mathrm{Re}(z^k)$ in variables n and m (recall $z = n + mi$) factors generically into relatively small factors for composite k, so that, when $B^2 = A^2 + d^2$, we can expect that A is frequently smooth or powersmooth. In practice, running a simple search using this method, one very readily obtains example insecure parameters:

$$
\begin{aligned}
B &= 5^{105} \\
A &= 2^2 \cdot 11 \cdot 19 \cdot 29 \cdot 41 \cdot 59 \cdot 61 \cdot 139 \cdot 241 \cdot 281 \cdot 419 \cdot 421 \cdot 839 \cdot 2381 \cdot 17921 \\
&\quad \cdot 21001 \cdot 39761 \cdot 74761 \cdot 448139 \cdot 526679 \cdot 771961 \cdot 238197121 \\
d &= 3^2 \cdot 13 \cdot 79 \cdot 83 \cdot 239 \cdot 307 \cdot 2801 \cdot 3119 \cdot 3361 \cdot 3529 \cdot 28559 \cdot 36791 \cdot 53759 \\
&\quad \cdot 908321 \cdot 3575762705759 \cdot 23030958433523039
\end{aligned}
$$

For this example, if we take $p = 105AB - 1$, we obtain a prime which is 3 modulo 4. Note that here $B \approx 2^{244}$ and $A \approx 2^{238}$. Many other primes can easily be obtained (replacing 105 with 214, 222, etc).

Remark 21. When choosing parameter sets to run B-SIDH [7], if the user is very unlucky, they could hit upon an instance of a weak prime. With this in mind, it would be prudent to check that a given combination of A, B, and p doesn't fall into this category before implementing such a B-SIDH instance.

5 Conclusion

Our results do not affect the security of SIKE, but we show that (under some heuristics) Problem 1 is easier than the 'pure' isogeny problem in far more generality than previously known [21], and we give attacks that may apply to some instantiations of SIDH variants [13, 7]. In particular, we have demonstrated, for A and B the degrees of Alice's and Bob's secret isogenies respectively:

1. A heuristic polynomial-time attack when $B > A^2 > p^2$ or $B > A^3 > p^{\frac{3}{2}}$.
2. A heuristic argument that even a mild imbalance of parameters leads to a improvement over the generic meet-in-the-middle or claw-finding attack.

3. A heuristic argument that smaller solutions to Equation (3) should exist, and that if future work yields an efficient method to find these solutions, they can be exploited to get an attack for a further improved balance (in particular including parameters satisfying $AB \mid (p+1)$).
4. A heuristic polynomial-time torsion-point attack on SIDH instances using an *insecure* starting curve with $B > A^2$, and an attack of classical complexity $O^*(p^{\frac{2}{5}})$ and quantum complexity $O^*(p^{\frac{1}{8}})$ on SIDH instances using an insecure starting curve with $B \approx A \approx p^{\frac{1}{2}}$.
5. An argument that there are exponentially many insecure curves, and a proof when $A^3 > B > A^2$.
6. The existence of special primes p, together with appropriate A and B, for which torsion-point attacks are especially effective, even when using balanced parameters $A \approx B$ and/or using a starting curve with j-invariant 1728.

As a result of this work, we recommend against instantiating supersingular-isogeny algorithms with imbalanced parameters. Furthermore, caution is advised if using a starting curve or base field prime of unknown or suspicious origin.

References

[1] Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. *IACR Cryptology ePrint Archive*, 2018:313, 2018.

[2] Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. Practical supersingular isogeny group key agreement. *IACR Cryptology ePrint Archive*, 2019:330, 2019.

[3] Christian Batut, Karim Belabas, Dominique Bernardi, Henri Cohen, and Michel Olivier. *User's Guide to PARI-GP*. Université de Bordeaux I.

[4] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997.

[5] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427. Springer, 2018.

[6] Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. Rational isogenies from irrational endomorphisms. In *EUROCRYPT (2)*, volume 12106 of *LNCS*, pages 523–548. Springer, 2020.

[7] Craig Costello. B-SIDH: Supersingular Isogeny Diffie-Hellman using twisted torsion. *IACR Cryptology ePrint Archive*, 2019:1145, 2019.

[8] Cyprien Delpech de Saint Guilhem, Péter Kutas, Christophe Petit, and Javier Silva. SÉTA: Supersingular encryption from torsion attacks. *IACR Cryptology ePrint Archive*, 2019:1291, 2019.

[9] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In *EUROCRYPT (3)*, volume 10822 of *LNCS*, pages 329–368. Springer, 2018.

[10] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *ASIACRYPT (1)*, volume 10031 of *LNCS*, pages 63–91, 2016.

[11] Gábor Ivanyos and Lajos Rónyai. Finding maximal orders in semisimple algebras over $\mathbb{Q}$. *Computational Complexity*, 3(3):245–261, 1993.

[12] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation. *Submission to [19]*, 2017. https://sike.org.

[13] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation. *Updated version of [12] for round 2 of [19]*, 2019.

[14] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, pages 19–34. Springer, 2011.

[15] Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, volume 11692 of *LNCS*, pages 32–61. Springer, 2019.

[16] Masanobu Kaneko. Supersingular j-invariants as singular moduli mod p. *Osaka Journal of Mathematics*, 26(4):849–855, 1 1989.

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

[18] Jonathan Love and Dan Boneh. Supersingular curves with small non-integer endomorphisms. *arXiv preprint* arXiv:1910.03180, 2019.

[19] National Institute of Standards and Technology. Post-quantum cryptography standardization, December 2016. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[20] Hiroshi Onuki. On oriented supersingular elliptic curves. *arXiv preprint* arXiv:2002.09894, 2020.

[21] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *ASIACRYPT (2)*, volume 10625 of *LNCS*, pages 330–353. Springer, 2017.

[22] Christophe Petit and Kristin E. Lauter. Hard and easy problems for supersingular isogeny graphs. *IACR Cryptology ePrint Archive*, 2017:962, 2017.

[23] Rajeev Anand Sahu, Agnese Gini, and Ankan Pal. Supersingular isogeny-based designated verifier blind signature. *IACR Cryptology ePrint Archive*, 2019:1498, 2019.

[24] Carl Ludwig Siegel. Über die Classenzahl quadratischer Zahlkörper. *Acta Arithmetica*, pages 83–86, 1935.

[25] Denis Simon. Quadratic equations in dimensions 4, 5 and more. *Preprint*, 2005.

[26] John Voight. Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms. In *Quadratic and higher degree forms*, pages 255–298. Springer, 2013.

A Additional examples of insecure primes

In the examples in Subsection 4.4, we let $A = 2^{216}$, $B = 3^{300}, e = 1$. We let d to be B modulo A^2, and $D = \frac{B-d^2}{A^2}$, where

$$\begin{aligned} D = {} & 168964203332467019300662458467972858204530430466926 12\ldots \\ & \ldots 341602757052612968476197336341477871394161800713702 53\ldots \\ & \ldots 151875694583397987452872630971686172791991823800180. \end{aligned}$$

We first choose $c = 53$, then $D - c^2$ is a prime number (i.e., $a = 1,\ b = 0$),

$$\begin{aligned} p = {} & 168964203332467019300662458467972858204530430466926 12\ldots \\ & \ldots 341602757052612968476197336341477871394161800713702 53\ldots \\ & \ldots 151875694583397987452872630971686172791991823797371. \end{aligned}$$

When $c = 355$, then $D - c^2$ is 5 times a prime number, namely,

$$\begin{aligned} p = {} & 337928406664934038601324916935945716409060860933852 24\ldots \\ & \ldots 683205514105225936952394672682955742788323601427405 06\ldots \\ & \ldots 303751389166795974905745261943372345583983647348 31. \end{aligned}$$

Both of these primes are congruent to 3 modulo 4.

We also give additional examples of Pythagorean triples as described in Section 4.5. In particular, let

$$\begin{aligned} B &= 17^{60}, \\ A &= 2^5 \cdot 3^2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 \cdot 41 \cdot 47 \cdot 59 \cdot 61 \cdot 101 \cdot 181 \cdot 191 \cdot 199 \cdot 239 \cdot 421 \\ &\quad \cdot 541 \cdot 659 \cdot 769 \cdot 2281 \cdot 16319 \cdot 30119 \cdot 285599 \cdot 391679 \cdot 1039081 \cdot 1109159 \end{aligned}$$

For this, $177AB - 1 \equiv 3 \pmod 4$ is prime. Finally, a powersmooth example is given by

$$\begin{aligned} B &= 5^8 \cdot 13^4 \cdot 17^4 \cdot 29^4 \cdot 37^4 \cdot 41^4 \cdot 53^4 \cdot 61^4 \cdot 73^4 \cdot 89^4 \cdot 97^4, \\ A &= 2^4 \cdot 3 \cdot 7 \cdot 11 \cdot 23 \cdot 31 \cdot 127 \cdot 199 \cdot 811 \cdot 2903 \cdot 155383 \cdot 842041 \cdot 933199 \cdot 1900147 \\ &\quad \cdot 8333489 \cdot 21629743 \cdot 30583723 \cdot 69375497 \end{aligned}$$

For this, $19AB - 1 \equiv 3 \pmod 4$ is prime.

B Implementation

In this section we report on computations regarding Algorithm 4 for some concrete parameters. We chose parameters $A = 2^{216},\ B = 3^{300},\ p = AB \cdot 277 - 1$. It is easy to see that we can choose $e = 1$ and d equal to B modulo A^2. Now we need to factor $\frac{B^2-d^2}{A^2}$. The way we chose d makes it easy as $\frac{B^2-d^2}{A^2} = \frac{B-d}{A^2}(B+d)$.

This is something which applies in other cases as well, and to make sure that factorization is easy one can try choices of d until factoring $B+d$ is feasible (e.g., $B+d$ is a prime number). For completeness, the factorization of $\frac{B^2-d^2}{A^2}$ is

$$\begin{aligned}
&2^2 \cdot 5 \cdot 23 \cdot 359 \cdot 2089 \cdot 39733 \cdot 44059 \cdot 74353 \cdot \\
&376287243430425811904334555393892643554045789647043 47 \dots \\
\dots\, &59039416676945740598806299461624575502089058332472952 \dots \\
\dots\, &94279089212441484219144994 63.
\end{aligned}$$

Once the factorization is known one can apply Simon's algorithm, implemented in Pari/GP [3] as `qfsolve()`, to compute a rational solution to the equation $pa^2 + pb^2 + c^2 = \frac{B^2-d^2}{A^2}$. A rational solution is given by

$$\begin{aligned}
a = {} & 323191234965367868432544587656085530956635685218723 34 \dots \\
\dots\, & 297530315749275438736572/z \\
b = {} & 379028937360168807771938548752530455531754575730671 91 \dots \\
\dots\, & 240634037840067475117556 0/z \\
c = {} & 854371287774171360224239413215855057617571606157987 39 \dots \\
\dots\, & 724060756960541951688471438700203893240926171912847 23 \dots \\
\dots\, & 809057988350649555534072083205999014782820898065439 45 \dots \\
\dots\, & 266931422175906643935346/z,
\end{aligned}$$

where

$$\begin{aligned}
z = {} & 879783485770113354174532396490993822256500213758092 20 \dots \\
\dots\, & 482035444121140799326417957094912384646917067558511 9.
\end{aligned}$$

Once θ is computed one has to compute an order $\mathcal{O}_0$ which contains θ. This can be accomplished in various ways. One way is to find a θ' such that $\theta\theta' + \theta'\theta = 0$ and θ'^2 is an integer multiple of the identity. This amounts to finding the kernel of the linear map $\eta \mapsto \theta\eta + \eta\theta$, which is a 2-dimensional vector space over $\mathbb{Q}$ (i.e., one chooses an element in this kernel and then multiplies it with a suitable integer). It is preferable to construct $\mathcal{O}_0$ in this way so that the discriminant of the order is the square of the reduced norm of $\theta\theta'$. In particular, if we choose a θ' whose norm is easy to factor, then the discriminant is also easy to factor. One has a lot of flexibility in choosing θ' and lattice reduction techniques help finding one which is sufficiently small and has an easy factorization. Note that the norm of θ' will always be divisible by p since the discriminant of every order is a multiple of p (and the norm of θ is coprime to p). Finally, one can compute a maximal order containing $\mathcal{O}_0$ using MAGMA's [4] `MaximalOrder()` function.

C Open problem: application to (less) imbalanced SIKE

Our attacks from Section 4.1 and 4.3 apply to insecure curves with balanced parameters. It is an interesting question whether these can be used to solve

Problem 1 starting from the curve with j-invariant 1728: a natural approach here is to translate torsion information from the starting curve to an insecure curve, solve Problem 1 on the insecure curve and use it to find the secret isogeny. We state two easy propositions on when this is possible:

Proposition 22. *Suppose that E is an $(A, B, \mathscr{C})$-insecure curve. Suppose furthermore that there is an isogeny of degree M from E_0 (the curve with j-invariant 1728) to E for which M and B are coprime. Then any SIDH private key (for public parameters A, B and starting curve E_0) can be recovered in time $O^*(M^2 A^{\mathscr{C}})$.*

Proposition 23. *Let E be an (A, B')-insecure supersingular elliptic curve. Let E_0 be the elliptic curve with j-invariant 1728, and assume an isogeny $\psi : E_0 \to E$ of smooth degree d is given. Then for every B for which $(B' \cdot d) \mid B$ the curve E_0 is (A, B)-insecure.*

Proposition 22 can only be applied if M is small. If B is a power of 3, then Proposition 23 can be applied if one can find a maximal order where the corresponding elliptic curve is close to the starting curve in the 3-isogeny graph. If we are (un)lucky and there is a sufficiently insecure curve close to the elliptic curve with j-invariant 1728, then extending the techniques from [18] could potentially lead to attacks against imbalanced SIDH, but with that imbalance being independent of p.

Proof (of Proposition 23). First we show how to compute a basis of the B'-torsion of E, which lies in the image of ψ. Let $N = B' \cdot d$. The conditions of the propositions imply that N divides B. Let P_d be a generator of the kernel of ψ and let $P_N \in E_0$ be a point of order N such that $P_d = B' P_N$ (such a P_N can be computed efficiently as B' is smooth). Compute $Q_N \in E_0[N]$ such that P_N, Q_N are a basis of the N-torsion. The order of $\psi(P_N)$ is B' as $B'\psi(P_N) = \psi(P_d) = 0$ and $\widehat{\psi} \circ \psi = [d]$. Similarly, the order of $\psi(Q_N)$ is a multiple of B', thus there exists an integer m such that the order of $\psi(mQ_N)$ is exactly B'. We show that $\psi(P_N)$ and $\psi(mQ_N)$ are $\mathbb{Z}/B'\mathbb{Z}$-independent and thus are a basis of $E[B']$. Suppose there exist $\alpha, \beta \in \mathbb{Z}/B'\mathbb{Z}$ such that

$$\alpha\psi(P_N) + \beta\psi(mQ_N) = 0, \tag{13}$$

then $\alpha P_N + m\beta Q_N$ is in the kernel of ψ. Thus $\alpha P_N + m\beta Q_N = B'\gamma P_N$ for some $\gamma \in \mathbb{Z}/B'\mathbb{Z}$, which happens if and only if $\alpha \equiv B'\gamma \pmod{N}$ and $m\beta \equiv 0 \pmod{N}$. Now we are done since this implies that $\alpha \equiv 0 \pmod{B'}$, which by (13) implies that $\beta\psi(mQ_N) = 0$, hence also $\beta \equiv 0 \pmod{B'}$ (since the order of $\psi(mQ_N)$ was B').

Let $\phi : E_0 \to E$ be an isogeny of degree A. Then even though ϕ is not known to us we can compute the isogeny $\psi' : E \to E'$ such that the kernel of ψ' is generated by $\phi(P_d)$ since d divides B. Suppose the kernel of ϕ is generated by a point A (which is not known). Let $\phi' : E \to E'$ be the isogeny whose kernel is generated by $\psi(A)$. If we can compute ϕ' then we can also compute ϕ since

the degree of ψ and ϕ are coprime (so $[d]A$ generates the same subgroup as A). In the first paragraph we have shown that we can compute the action of ϕ' on the B'-torsion of E. We can now compute ϕ' in polynomial time since E was (A, B')-insecure (and the order of $\psi(A)$ is A since A and d are coprime).