Torsion point attacks on "SIDH-like" cryptosystems

Péter Kutas[1] and Christophe Petit[1,2]

[1] School of Computer Science, University of Birmingham, UK
P.Kutas@bham.ac.uk
[2] Laboratoire d'Informatique,
Université libre de Bruxelles, Belgium
christophe.f.petit@gmail.com

Abstract. Isogeny-based cryptography is a promising approach for post-quantum cryptography. The best-known protocol following that approach is the *supersingular isogeny Diffie-Hellman protocol* (SIDH); this protocol was turned into the CCA-secure key encapsulation mechanism SIKE submitted to NIST post-quantum standardization process, which has remained in the third round as an "alternate" candidate.
Isogeny-based cryptography generally relies on the conjectured hardness of computing an isogeny between two isogenous elliptic curves, and most cryptanalytic work referenced on SIKE's webpage exclusively focuses on that problem. Interestingly, the hardness of this problem is sufficient for neither SIDH nor SIKE. In particular, these protocols reveal additional information on the secret isogeny, in the form of images of specific torsion points through the isogeny.
This paper surveys existing cryptanalysis approaches exploiting this often called "torsion point information", summarizes their current impact on SIKE and related algorithms, and suggests some research directions that might lead to further impact.

1 Introduction

Isogeny-based cryptography is a promising candidate for post-quantum cryptography. It originates from Courveignes's seminal work [12] where he introduced the notion of hard homogenous spaces and instantiatied it with ordinary elliptic curves, and Charles, Goren and Lauter's hash function [9] (CGL) based on isogenies of supersingular elliptic curves. In 2011 de Feo and Jao introduced SIDH [23] and in the recent years field has blossomed for example with the introduction of CSIDH [7] (the only post-quantum scheme which provides non-interactive key exchange), SQISign and many more isogeny-based schemes. SIKE [21], which is a key encapsulation mechanism derived from SIDH, is currently a 3rd round alternate candidate in NIST's post-quantum standardization project.

Most isogeny-based protocols today are based on the hardness of computing isogenies between supersingular elliptic curves. However, only CGL hash function [9] and the GPS signature scheme [19] only rely on this "pure" isogeny problem. In SIDH protocol, parties send over torsion point images, which motivates the study of the following problem:

Problem 1.1 (Supersingular Isogeny with Torsion (SSI-T)). For a 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 an unknown degree-A isogeny $\phi\colon E_0 \to E$, and given the restriction of ϕ to the B-torsion of E_0, compute ϕ.

In [23] a more specific version of the SSI-T problem is called the CSSI problem. Computing isogenies between supersingular elliptic curves is a natural algorithmic question which has been studied for a long time, but the SSI-T problem is specific to SIDH and its variants. It is natural to wonder how the SSI-T problem relates to the pure isogeny problem. The aim of this survey paper is to give a summary of results which exploit the extra information in various ways. Our goal is to explain these techniques, assess their impact and warn designers of future protocols to take these results into account. The current state of the art is that SIKE is not affected by these attacks.

The structure of the paper is as follows. In Section 2 we recall basic mathematical results on supersingular elliptic curves and quaternion algebras and the SIDH protocol. In Section 3 we discuss active attacks on SIDH, namely the GPST attack [18] and its extensions. In Section 4 we discuss how the endomorphism ring computation problem relates to the security of SIDH and the SSI-T problem in general. In Section 5 we discuss passive torsion-point attacks which originate from [29] and were later improved significantly in [13]. In Section 6 we discuss the quantum hidden-shift attack from [26]. Finally, in Section 7 we discuss open problems which could shape the future of torsion-point attacks.

2 Supersingular isogeny Diffie-Hellman and its variants

We refer to [31] and [17] for general background on elliptic curves and isogeny-based cryptography. The following high-level description of SIDH [23] and some of its variants relevant to Problem 1.1 are taken nearly verbatim from [13, Section 2.1].

Recall that $E[N]$ denotes the N-torsion subgroup of an elliptic curve E and $[m]$ denotes scalar multiplication by m. 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 $_{p^2}$ 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 [21] 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 [21] uses $p = 2^{216} \cdot 3^{137} - 1$. Other constructions belonging to the SIDH 'family tree' of protocols use different types of parameters [1, 11, 30].

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: see [8, 27]), or by isogeny walks starting from such curves (where knowledge of the path reveals the endomorphism ring, thus requiring 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 [21]. Various variants of SIDH exist in the literature.

In [1] the authors propose an n-party key agreement. The idea is to use primes of the form $p = f\prod_{i=1}^{n}\ell_i^{e_i} - 1$ where ℓ_i is the i-th prime number, the i-th party's secret isogeny has degree $\ell_i^{e_i}$, the i-th participant provides the images of a basis of the $\prod_{j=1}^{n}\ell_j^{e_j}/\ell_i^{e_i}$ torsion, and f is a small cofactor. They choose the starting curve to be of 1728 and choose the e_i in such a way that all the $\ell_i^{e_i}$ are of roughly the same size. This is an example of an SIDH-like protocol; for this protocol to be secure it is required that Problem 1.1 be hard when $A = \ell_1^{e_1}$ and $B = f\prod_{i=2}^{n}\ell_i^{e_i}$.

Another example of a SIDH-like scheme is B-SIDH [11]. In B-SIDH, the prime has the property that $p^2 - 1$ is smooth (as opposed to just $p - 1$ being smooth) and $A \approx B \approx p$. It would seem that choosing parameters this way one has to work over $\mathbb{F}_{p^4}$ but in fact the scheme simultaneously works with the curve and its quadratic twist (i.e., a curve which is not isomorphic to the original curve over $\mathbb{F}_{p^2}$ but has the same) and avoids the use of extension fields. The main advantage of B-SIDH is that the base-field primes used can be considerably smaller than the primes used in SIDH.

3 Active attacks

3.1 GPST and variants

Since SIDH is a key exchange analogous to classical Diffie-Hellman, it is a natural question whether parties could use static keys. In 2016 Galbraith, Petit, Shani and Ti [18] proposed an active attack on SIDH if one party has a static key. The main idea of the attack is to send over maliciously generated torsion points and check whether the key exchange was successful or not. After every key exchange the adversary will learn one more bit from the secret key.

In order to describe the attack we define the following oracle which abstracts the method described above.

Definition 3.1. *Let α be a secret integer. Let E, E_1, E_1', P, Q be a tuple such that P, Q generate $E_1[A]$. Then the oracle returns "true" if $E_1' \cong E_1/\langle P + \alpha Q\rangle$ and returns "false" otherwise.*

The motivation for this oracle comes from the way the SIDH key exchange is computed. Alice receives $\phi_B(P_A)$ and $\phi_B(Q_A)$ and compoutes the curve

$$E_B/\langle \phi_B(P_A) + \alpha\phi_B(Q_A)\rangle.$$

The key exchange is successful if both parties computed the same curve (up to isomorphism). Unfortunately, there is no way to tell without knowing ϕ_B whether the points sent over are truly the images of P_A and Q_A under ϕ_B or are just some other basis of $E_B[A]$. For simplicity we suppose that $A = 2^n$ but the attack generalizes to arbitrary smooth degree isogeny.

Remark 3.2. There is a pretty simple attack if one is allowed to send over points of order smaller than A. Namely we do a honest key exchange where we send over $\phi_B(P_A), \phi_B(Q_A)$then in the kth step we send over $\phi_B(2^{k-1}P_A), \phi_B(2^{k-1}Q_A)$. This will essentially reveal the isogeny path from E_B to E_{AB}, from which the secret is easily deduced. However, such an attack is easily detectable as the order of points can be checked by using pairings.

Let $P_A + \alpha Q_A$ be the secret kernel generator of Alice. The first step of the attack is a genuine key exchange: Bob chooses an isogeny $\phi_B : E \to E_B$ with kernel $P_B + \beta Q_B$, sends over $\phi_B(P_A), \phi_B(Q_A)$ and computes the common curve $E/\langle P_A + \alpha Q_A, P_B + \beta Q_B\rangle$. Let $R = \phi_B(P_A)$ and $S = \phi_B(Q_A)$. Our first goal is to determine the least significant bit of α. The trick is to send over E_B and points R , $S + 2^{n-1}R$. Then Alice computes $E_B/\langle R + \alpha(S + 2^{n-1}R)\rangle$ which is isomorphic to

- $E_B/\langle R + \alpha S\rangle$ if and only if α is even
- $E_B/\langle R + \alpha S + 2^{n-1}R\rangle$ if and only if α is odd.

Let $E_{AB} = E_B/\langle R+\alpha S\rangle$. Now sending $(E, E_B, R, S+2^{n-1}R, E_{AB})$ to the oracle determines the least significant bit of α: if the oracle returns true, then α is even, otherwise α is odd.

In order to compute the remaining bits of α, we write α in the form $\sum_{i=0}^{n-1} \alpha_i 2^i$. Let s_k denote the partial sum $s_k = \sum_{i=0}^{k-1} \alpha_i 2^i$. Suppose now that we have already computed s_k and our goal is to compute α_k. Then we send over the following points:

$$(1 - s_k 2^{n-k-1})R,\ S + 2^{n-k-1}R$$

Then Alice computes $E_B/\langle (1 - 2^{n-k-1})R + \alpha S + 2^{n-k-1}R\rangle$ which is isomorphic to E_{AB} if α_k is even and isomorphic to $E_B/\langle R + \alpha S + 2^{n-1}R\rangle$ if α is odd. This implies that knowing s_k we can compute α_k by one oracle call. It is clear that after n calls to the oracle we retrieve the static secret key α.

There are various countermeasures against the GPST attack. The most efficient and standard way is to use the Fujisaki-Okamoto transform. This is how the

IND-CCA2-secure scheme SIKE [21] is obtained. However, for some applications this is not desirable, namely when both parties' keys are static.

In 2017 Azarderaksh et al. [2] introduced a variant of SIDH called k-SIDH. The main idea is the following. Alice and Bob choose k different secret isogenies (al of Alice's isogenies are of degree 2^m and all of Bob's isogenies are of degree 3^l) and they compute k^2 SIDH key-exchanges (as each pair of secrets corresponds to one key exchange). Finally, they hash the k^2 different j-invariants to obtain a shared secret. The efficiency of k-SIDH is navigated by the size of k. Public key sizes grow linearly in k and the number SIDH key exchanges is a quadratic function of k. In the original paper [2] the authors gave a brief security analysis and suggested to use $k = 60$. Such a large k makes the scheme very impractical, so it is important to have a clearer security analysis of k-SIDH. In particular, is 2-SIDH secure? In [15] Dobson et al. demonstrated an attack against 2-SIDH which generalizes to larger k. The complexity of the attack is exponential in k but it breaks the scheme in polynomial time for small k. They suggest that $k = 46$ is already potentially a secure choice. Their attack in the $k = 2$ case is far from trivial as the GPST attack does not generalize in a straightforward manner (it gives an exponential complexity even in the $k = 2$ case). Their key idea is to compute additional information at each step. In GPST one only has to keep track of the computed bits of α. In the 2-SIDH attack on the other hand, one has to compute each step in the isogeny graph plus preimages of certain points. The bottleneck of the algorithm is the computation of these various preimages as they require a lot of oracle calls.

Since k-SIDH is quite impractical, it is natural to attempt to speed it up. Jao and Urbanik [35] proposed a way of lowering the number of key exchanges by using automorphisms of the starting curves. This way one secret corresponds to three curves which lowers the size of the public keys and the communication cost. However. the attack from [15] can be extended to the Jao-Urbanik scheme [3] in a way that actually exploits the relationship between the three isomorphic curves. If you compare state-of-the-art attacks on both schemes, then the analysis in [3] suggests that k-SIDH is actually more efficient (this may change in the future if an improved attack on k-SIDH cannot be adapted to the Jao-Urbanik scheme). Jao and Urbanik also suggest to switch from 2-isogenies to 11 or 13-isogenies as it increases the attack complexity more than it increases computational costs.

It is still an open problem whether there exists some variant of k-SIDH which is efficient and avoids these known attacks.

3.2 Fault attacks

In GPST attack and its variants, one party purposely produces erroneous torsion points, and recovers information on the secret key from (changes in) the shared curve E_{AB}. When fault attacks are feasible, an alternative approach is to force the other party to make faulty computations.

In SIDH protocol, isogenies are computed in a sequential way, as the composition of several low degree isogenies. In [20], a loop-abort fault attack is described where one party can force the other one to stop that computation after an arbitrary number of steps, and return the current curve rather than the final one. This provides an oracle similar to the one used in the GPST attack, and the key can be recovered similarly.

In [34], another fault model is considered where some register value is replaced by a random value during computation. If this happens to a register containing part of the x-coordinate of P_B, then the resulting x coordinate is still a point on the curve with a probability roughly 1/2, but is likely to have an order that is not coprime with $\deg \phi_A$. As a result its image will reveal part of the isogeny, more precisely multiplying the image by the cofactor (its order divided by the gcd between its order and $\deg \phi_A$) produces a point in the kernel of its dual. We refer to [34] for details.

4 Reduction to the endomorphism ring computation problem

Computing the endomorphism of a supersingular elliptic curve is a classical problem in computational number theory. Given an elliptic curve E defined over a finite field of characteristic p, the problem is to find $\mathrm{End}(E)$. The first algorithm to solve this is described in Kohel's thesis [25] and was later improved by Delfs-Galbraith [14] to a running time of $\tilde{O}(p^{1/2})$. The most recent algorithm [16] is a slight variation with essentially the same complexity $O(\log(p)^2 p^{1/2})$. The best known quantum algorithm is due to Biasse, Jao and Sankar [4] and has a running time of $\tilde{O}(p^{1/4})$.

It is a natural to ask how finding isogenies between supersingular elliptic curves relates to computing endomorphism rings. The KLPT algorithm [24] implies that if one knows the endomorphism rings of both curves, then one can compute an isogeny between them. For cryptographic applications, a much more natural question is the following. Let ϕ be a secret isogeny of degree d between E_1 and E_2. Find ϕ if the endomorphism rings of E_1 and E_2 are known.

Let us first recall some facts about isogenies between supersingular elliptic curves. Let E_1, E_2 be supersingular elliptic curves defined over $\mathbb{F}_p^2$. Then the set $\mathrm{Hom}(E_1, E_2)$ of isogenies between E_1 and E_2 has a very specific structure. First, $\mathrm{Hom}(E_1, E_2)$ is a $\mathbb{Z}$-lattice as the integer linear combination of isogenies from E_1 to E_2 is again an isogeny from E_1 to E_2. Furthermore, let $\sigma_1 \in \mathrm{End}(E_1)$, $\sigma_2 \in \mathrm{End}(E_2)$ and $\phi \in \mathrm{Hom}(E_1, E_2)$. Then $\phi \circ \sigma_1 \in \mathrm{Hom}(E_1, E_2)$ and $\sigma_2 \circ \phi \in \mathrm{Hom}(E_1, E_2)$. In other words $\mathrm{Hom}(E_1, E_2)$ is a left $\mathrm{End}(E_2)$ and a right $\mathrm{End}(E_1)$-module. In particular the next lemma shows that $\mathrm{Hom}(E_1, E_2)$ is isomorphic to a left ideal of $\mathrm{End}(E_2)$:

Lemma 4.1. *[36, 42.2.7] Let $Hom(E_2, E_1)$ denote the set of isogenies from E_2 to E_1 and let $\mathcal{O}_1$ and $\mathcal{O}_2$ denote the endomorphism rings of E_1 and E_2 respectively. Let I be a connecting ideal of $\mathcal{O}_1$ and $\mathcal{O}_2$ and let ϕ_I denote the corresponding isogeny. Then the map $\phi_I^* : Hom(E_1, E_2) \to I$, $\psi \mapsto \psi \circ \phi_I$ is an isomorphism of left $\mathcal{O}_1$-modules.*

One can also show that the rank of $\mathrm{Hom}(E_1, E_2)$ as a $\mathbb{Z}$-lattice is 4. The KLPT algorithm also implies that if the endomorphism rings of E_1 and E_2 are known, then one can compute a $\mathbb{Z}$-basis of $\mathrm{Hom}(E_1, E_2)$ as it is isomorphic to a connecting left ideal. Note that such a basis is given as elements of the quaternion algebra and not as rational maps as their degree can be large and not smooth (thus writing down the coefficients of the rational functions would take exponential time in $\log\, p$).

The first algorithm relating endomorphism ring comutation and computing isogenies of a specific degree is from [18]. The main observation is that in SIDH the secret isogeny has degree approximately $\sqrt{p}$. Heuristically, such an isogeny should be in general the shortest isogeny between two randomly selected curves, which gives the following attack. Compute a $\mathbb{Z}$-basis of $\mathrm{Hom}(E_1, E_2)$ using the KLPT algorithm. Then find the shortest element in $\mathrm{Hom}(E_1, E_2)$ using the LLL-algorithm. Heuristically, this should be the secret isogeny one is looking for. The authors demonstrate this with experiments in MAGMA.

The algorithm implies that in SIDH if the endomorphism ring of E and E_A is known, then one can recover the secret isogeny ϕ_A in polynomial time. However, in B-SIDH the respective curves are no longer close (the curves are roughly p apart), thus the algorithm from [18] fails. It is a natural question whether one can extend the algorithm from [18] to be applicable to B-SIDH as well. This is especially important because for B-SIDH such an attack would be more efficient than a meet-in-the-middle attack (which is currently not true for SIDH).

The main idea of [33] is that one can exploit the torsion information provided to generalize the attack from [18] to a wide variety of parameters. Note that the algorithm in [18] did not use the torsion information at all; it solely relied on the curves being close. We sketch the attack from [33]. Similarly, one computes an LLL-reduced basis of $\mathrm{Hom}(E_1, E_2)$, let these be $\phi_1, \phi_2, \phi_3, \phi_4$. Then the secret isogeny ϕ can be written as $\phi = \sum_{i=1}^{4} x_i \phi_i$ where the x_i are integers. Using the torsion information provided one can determine the x_i modulo B by solving a system of linear equations. Why is this information useful? The reason is that an LLL-reduced basis has the property that one can bound the x_i using the smallest degree element in $\mathrm{Hom}(E_1, E_2)$ and the degree of the secret isogeny. This way if $|x_i| < B/2$, then a modulo B solution can be uniquely lifted to an integer solution. This way one can retrieve the secret isogeny whenever $A/B < 8\sqrt{p}$. When looking at SIDH or B-SIDH as a key exchange, one can assume that $B > A$, so this should apply to any reasonable instantiation of SIDH.

It is still an open problem whether one can recover a secret isogeny of degree d between curves with known endomorphism rings in general. Indeed, both previously described algorithms use some extra information, namely closeness of the curves or torsion-point information.

5 Shifted endomorphism attacks

In this section we discuss algorithms for the SSI-T problem. The central questions are the following:

- For which parameters A, B, p can one solve SSI-T in polynomial time?
- For which parameters A, B, p can we do better than generic meet-in-the-middle algorithms?

The first work in this area is Petit's algorithm [29], which was first improved in [6] and then further improved in [13]. The starting point is the following. Let $\phi : E_1 \to E_2$ be an isogeny of degree A and suppose we know the action of ϕ on the B-torsion. Let $\theta \in \text{End}(E_1)$ (given by some efficient representation). Then one knows how $\phi \circ \theta \circ \hat{\phi}$ acts on $E_2[B]$. Furthermore, this is also true for any τ of the form $\phi \circ \theta \circ \hat{\phi} + [d]$ for any integer d. Why is this useful? The key idea of [29] is to choose θ in a way that $\deg(\phi \circ \theta \circ \hat{\phi} + [d]) = Be$ for some small e. Let $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$. Then one can decompose τ as $\psi \circ \eta$ where $\deg(\psi) = B$ and $\deg(\eta) = e$. One knows ψ as the action of τ is known on $E_2[B]$, and η can be computed by a generic meet-in-the-middle algorithm. Finally, one can obtain $\ker(\hat{\psi})$ as the intersection $\ker(\tau - [d]) \cap E_2[A]$.

The key part of the attack is the appropriate choice of θ, which requires knowledge of (at least part of) the endomorphism ring of E_1. However, in many applications E_1 is the special curve defined by the equation $y^2 = x^3 + x$ for which the structure of the endomorphism is known. Finding a suitable endomorphism $\theta \in \text{End}(E_1)$ then is equivalent to finding an integer solution (a, b, c, d, e) with small e to the following equation:

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

There is a natural strategy for solving this equation. First one solves it modulo A^2 by choosing d and e appropriately. Then one checks whether $Be - d^2$ is a square modulo p. If not, then one chooses a different d and e. If it is, then one finds c such that $c^2 \equiv \pmod{\frac{Be-d^2}{A^2}}$. Finally, one checks whether $\frac{\frac{Be-d^2}{A^2} - c^2}{p}$ is the sum of two squares. If yes, then one finds a, b using Cornacchia's algorithm. If not, then one starts over with a new d and e. It can be shown that heuristically, one does not need to iterate too many times. This is a simple algorithm but it fails for many parameter sets. The reason for this is that c^2 is usually of size $O(p^2)$ meaning that for many parameters even though one does not get local obstructions, the number $\frac{\frac{Be-d^2}{A^2} - c^2}{p}$ is negative, hence never a sum of two squares. In [29] it is shown that this does not happen when $A > p$ and $B > A^4$ in which case one can solve SSI-T in polynomial time.

Follow-up papers improve on Petit's original algorithm by relaxing the condition on θ and relating the algorithm to different equations. In [6] the authors use triangular decompositions and certain endomorphisms with many eigenvalues to derive the following equation:

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

In [13] the authors derive two new improvements: the dual isogeny method and the Frobenius method. The dual isogeny method also reduces to Equation 2 but uses a more direct approach. Namely if one can find θ such that $\deg(\phi \circ \theta \circ \hat{\phi} +$

$[d]) = B^2e$, then $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$ can be decomposed as $\tau = \psi \circ \eta \circ \psi'$ where $\deg(\psi) = \deg(\psi') = B$ and $\deg(\eta) = e$. The isogenies ψ and η can be computed in a similar fashion as before. The isogeny ψ' can be computed by essentially looking at $\tau(E_2[B])$. Another way to understand this approach is the following. Even though τ is not known apriori, its action on $E_2[B]$ is known. Thus one can look at τ as a 2×2 matrix with entries from $\mathbb{Z}/B\mathbb{Z}$. One can derive ψ by looking at the kernel of this matrix and one can compute ψ' by looking at the image of this matrix.

One can solve Equation 2 with the same method as the one presented for solving Equation 1. This provides a polynomial-time method whenever $B > pA$. However, heuristics show that a solution should exist for a much wider variety of parameters for example when $p \approx AB$ and $B > A^4$, but finding such a solution is still an important open problem. Why would an algorithm to compute these solutions be interesting? In variations and applications of SIDH one often uses special primes in order to be able to carry out computations over small extension fields. In particular there are two classes of primes which are used: SIDH primes of the form $p = ABf - 1$ where f is a small cofactor and B-SIDH primes where $p^2 - 1 = AB$ and A, B are smooth. For SIDH primes the previous approaches fail as in both approaches $B > p$. For B-SIDH primes the dual isogeny approach already has some impact: namely when $B > A^2$, then one can solve the SSI-T problem in polynomial time. This has no impact on the actual scheme proposed in B-SIDH [11] because there the parameters are balanced.

The main idea of the Frobenius approach outlined in [13] is the following. In the dual approach η needed to have small degree, as it was computed by a generic meet-in-the-middle algorithm. However, when the degree of η is a small multiple of p, then it can also be computed by applying the Frobenius and then brute-forcing the rest. This results in an alternative equation:

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

Now one can solve this equation by first setting $c = 0$ and $d = pd'$ and dividing by p. Then one obtains the equation

$$A^2(a^2 + b^2) + pd'^2 = B^2e. \tag{4}$$

Now the solving strategy is similar as before but one does not have to solve modulo p this time, just modulo A^2 and then hope that $\frac{B^2e - pd'^2}{A^2}$ is a sum of two squares. If not, then one can again iterate until a solution is found. This algorithm is implemented and can be found at `https://github.com/torsion-attacks-SIDH/6party`.

The main appeal of the Frobenius method is that it runs in polynomial time whenever $B > \sqrt{p}A^2$. In particular this applies when $p \approx AB$ and $B > A^5$. Note that it still does not apply to SIKE as there $A \approx B$. However, the choice of choosing balanced parameters in SIKE is essentially is only motivated by having the same security level for Alice and Bob. In many SIDH applications the parameters are not balanced [5],[22] and future protocols nay arise using unbalanced parameters.

All the previously described attacks run in polynomial time. However, it also makes sense to look at exponential-time attacks which outperform generic meet-in-the-middle algorithms. A general framework for these types of attacks is the following. One first guesses part of the secret isogeny and then one runs a torsion-point attack possibly with a larger e. If the torsion-point attack fails, then one guesses a different starting isogeny. This way one can obtain improvements for parameter sets which are less unbalanced. The state-of-the-art in this regard is summarized in Figure 1.

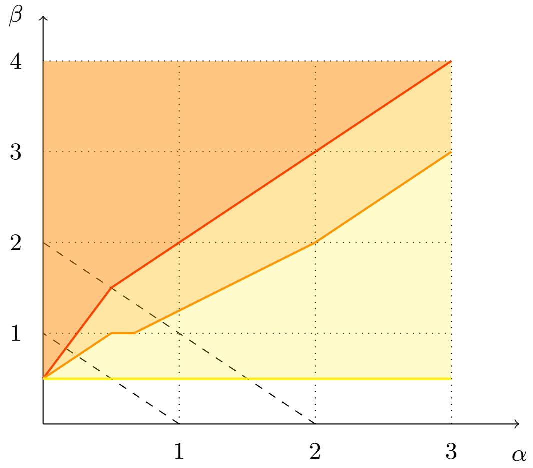

Fig. 1. Performance of attacks from [13]. Here $A \approx p^{\alpha}$ and $B \approx p^{\beta}$. Parameters (α, β) above the red, orange and yellow curves are parameters admitting a polynomial-time attack, an improvement over the best classical attacks, and an improvement over the best quantum attacks respectively. Parameters below the upper dashed line are those allowing $AB \mid (p^2 - 1)$ as in [11]. Parameters below the lower dashed line are those allowing $AB \mid (p - 1)$ as in [21, 22].

All these attacks assume that the starting curve is a special curve, namely the curve with j-invariant 1728 (the attack extends naturally to starting curves close to this curve). Starting from a random curve thwarts all these attacks. However, in certain scenarios it is not easy to detect that the starting curve was honestly generated (e.g., by taking a random walk starting from the curve $y^2 = x^3 + x$). Thus a natural question is the following: given A, B, p can one maliciously construct a starting curve for SIDH from which one can retrieve the secret key in polynomial time? When $B > A^2$, then the answer is yes. The main idea is looking at Equation 2 from a different perspective. In previous approaches one was looking for a specific θ on a specific starting curve. Instead one can try to look for the curve and the endomorphism together. This way one can look for θ in the entire quaternion algebra $B_{p,\infty}$ instead of restricting to one maximal order. This way we get Equation 2 but a, b, c do not need to be integers, only $pa^2 + pb^2 + c^2$ has to be an integer as it is the norm of an endomorphism (only integral elements of $B_{p,\infty}$ arise as endomorphisms). This way we can solve the equation modulo A^2 and then one is left with the equation:

$$pa^2 + pb^2 + c^2 = \frac{B^2e - d^2}{A^2}$$

Since we are now looking for rational solutions, we find a nontrivial zero of the homogeneous equation $pa^2 + pb^2 + c^2 - \frac{B^2e-d^2}{A^2}z^2$. This has a zero if and only if $B^2e - d^2$ is a quadratic residue modulo p, so again we have to iterate a couple of times for this to occur. Then one can find a solution using Simon's algorithm [32]. This way one has found θ but not the curve. Finding the curve can be obtained by finding a maximal order containing θ and translating it to a supersingular elliptic curve whose endomorphism ring is isomorphic to that order. In [13] the curves containing such a θ are called (A, B)-backdoor curves. The number of these curves is exponential in log p. Note that the condition for the existence of such a curve is $B > A^2$, so it does not depend on p. However, again this seemingly does not apply to balanced SIDH parameters. Even though one cannot break SIDH in polynomial time from a backdoor starting curve, in [13] it is shown that one can derive algorithms which even though are exponential, are faster than meet-in-the-middle algorithms.

This seems to suggest that against all intuition it is probably safer to instantiate SIDH starting from $y^2 = x^3 + x$, then from a random curve if there is no guarantee that the curve was generated honestly. Note that for SIDH one can actually derive a random starting curve by multiparty computation techniques but in many applications such an approach might not be feasible.

Finally, all these methods are ineffective if one could hash onto the supersingular isogeny graph, i.e., generate a random supersingular curve whose endomorphism ring is unknown to everyone. The techniques of this section again highlight the importance of the hashing problem.

6 Quantum hidden shift attack

In this Section we present a quantum subexponential algorithm for the SSI-T problem for certain parameter sets. One of the main fundamental differences between SIDH and CSIDH is that CSIDH is clearly based on a group action, namely the class group of $\mathbb{Z}[\sqrt{-p}]$ acts freely and transitively on supersingular elliptic curves defined over $\mathbb{F}_p$. It is well-understood how to compute the action of an ideal class of smooth norm on a given curve E. Furthermore, since the class group is commutative, the action provides a commutative group action which realizes the Hard Homogeneous Spaces concept of Couveignes [12]. In the SIDH setting one does not have similar natural group action due to the non-commutative nature of the full endomorphism ring (quaternion maximal orders have class groups but they are non-commutative). The implications of this are twofold: on one hand this makes SIDH less flexible (i.e., it is harder to derive further schemes from the core idea) on the other hand it possibly makes it immune to Kuperberg's algorithm.

There is however a different framework that applies to general supersingular elliptic curves as well. Let $f : I \rightarrow O$ be an injective one-way function and let G be a finite abelian group acting freely and transitively on I. Furthermore, suppose that if $f(i)$ is known (but i is not necessarily known), then one can compute $f(g * i)$. We call such an oracle a *malleability oracle*. In [26] it is shown that if one has access to a malleability oracle, then one can invert f in subexponential time. It is also shown that this framework applies to CSIDH and is essentially the same attack as the one proposed by Child, Jao and Soukharev [10]. However, surprisingly one can apply this framework to the SSI-T problem as well.

Let E be a supersingular elliptic curve. Let I be the set of cyclic subgroups of order A, and let O be the set of supersingular elliptic curves at distance A from E. Then $f : I \rightarrow O$ is defined by the mapping $f(\langle K\rangle) = E/\langle K\rangle$. Let θ be an endomorphism of E and let $E/\langle X\rangle$ be a curve of distance A from E. Then if the degree of θ is coprime to A, then $E/\langle\theta(X)\rangle$ is also a curve of distance A from E. Let $O = \mathrm{End}(E)$. Then this idea defines an action of $(O/AO)^*$ on the curves of distance A from E. It can be shown that $(O/AO)^* \cong GL_2(\mathbb{Z}/A\mathbb{Z})$. Since θ and $\lambda\theta$ where $\lambda \in \mathbb{Z}$ define the same action, it is actually more natural to consider the action of $PGL_2(\mathbb{Z}/A\mathbb{Z})$ on the set of curves of distance A from E. There are several questions at this point:

1. Is f injective?
2. Since $PGL_2(\mathbb{Z}/A\mathbb{Z})$ is non-commutative, how to choose the acting group G?
3. How do you compute $E/\langle\theta(X)\rangle$ without knowing X?

The first two questions are mere technicalities. One can split I in a way so that for each subset f is injective. In addition one can restrict to an abelian subgroup of $PGL_2(\mathbb{Z}/A\mathbb{Z})$ to make the action free and transitive on each of these subsets.

The answer to question 3 is more involved and this is the only part where the attack uses torsion point images. Let $E_X = E/\langle X\rangle$ and let $\phi : E \rightarrow E_X$ be a secret isogeny of degree A. Suppose we know the action of ϕ on $E[B]$. Our goal is to compute $E/\langle\theta(X)\rangle$ for an endomorphism θ. One has a commutative diagram described in Figure 2. Instead of focusing on the isogeny from E to $E/\langle\theta(X)\rangle$ we can go the other way on the diagram. Namely from E to E_X and then from E_X to $E/\langle\theta(X)\rangle$. The second step can be computed if the degree of θ divides B as we know the action of ϕ on the B-torsion. However, in general θ will not satisfy this property. The way to go around this issue is the following. Since we are working in O/AO we can choose a different representative of the coset containing θ. This means that we can switch from θ to any θ' which has the exact same action on the A-torsion. Now the goal is to find a $\theta' \in \mathrm{End}(E)$ such that $\theta' = \theta + A\theta''$ where $\theta'' \in \mathrm{End}(E)$ and the degree of θ' divides B. This can be achieved for special θ-s which one has to take into account when selecting the subgroup G of $PGL_2(\mathbb{Z}/A\mathbb{Z})$ for the group action. A particular choice for which this feasible is to use θ-s from $\mathbb{Z}[i]$ and the starting curve E with j-invariant

1728. Further improvements are also possible by using the Frobenius isogeny in a similar fashion to shifted endomorphism ring attacks. The conclusion is that the attack runs in subexponential time whenever $B > pA^4$.

Even though this is a worse attack complexity then the ones achieved with shifted endomorphisms, this attack highlights the fact that for certain parameter sets an efficient group action on the SIDH keyspace is possible. This further highlights how the SSI-T problem is different from the pure isogeny problem.

$$\begin{tikzcd}
E \arrow[r, "\varphi"] \arrow[d, "\theta"'] & E_A \arrow[d] \\
E \arrow[r] & E/\theta(\ker\varphi) \cong E_A/\varphi(\ker\theta)
\end{tikzcd}$$

Fig. 2. SIDH key exchange instance with isogenies φ and the endomorphism θ.

7 Open problems

There are various open problems that remain. Probably the most interesting questions is whether shifted endomorphism attacks and hidden shift attacks can be combined in some fashion. So far these attacks exploit torsion information in a different fashion so a common approach could be beneficial.

Furthermore, there is plenty of room for improvement in both approaches separately. In the dual isogeny approach, finding better solutions to Equation 2 is a clear path for improvement. Furthermore, in [13] there is an outline of a uniform approach which encompasses both the dual and the Frobenius approach. Possibly a more general viewpoint could also lead to improvements.

In the quantum attack the current approach only utilizes a small fraction of $PGL_2(\mathbb{Z}/A\mathbb{Z})$ in order to fit the framework needed for Kuperberg's algorithm. A natural way of extending this result could be to use a larger acting group and relating the issue of finding the secret isogeny to a hidden subgroup problem as opposed to a hidden shift problem.

Finally, all these approaches apply to elliptic curves. It is natural to study higher genus analogues of the SSI-T problem and whether the approaches generalize to higher genera.

8 Conclusion

SIKE's security relies on the "pure" isogeny problem (given two curves, find an isogeny between), but also on a variant which, among other specificities, provides the attacker with the images of some torsion points through the isogeny.

Several attacks have exploited similar information, starting from the GPST active attacks [18], continuing with torsion point passive attacks [13, 29] and most recently an attack contradicting the folklore intuition that hidden shift attacks cannot be applied to SIDH-like protocols because of their non commutative nature [26]. These attacks have improved over time: while [29] only worked for very unbalanced parameters, the latest improvements from [13] lead to a quantum attack with complexity similar (up to polylogarithmic factors) to previously known (non torsion point) attacks for SIKE parameters and a polynomial attack on a group key exchange from [1] for any number of parties greater than 6.

Future will tell whether these and other ideas will eventually affect the security of SIKE.

Bibliography

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

[2] Reza Azarderakhsh, David Jao, and Christopher Leonardi. Post-quantum static-static key agreement using multiple protocol instances. In *International Conference on Selected Areas in Cryptography*, pages 45–63. Springer, 2017.

[3] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Charlotte Weitkämper. On adaptive attacks against jao-urbanik's isogeny-based protocol. In *International Conference on Cryptology in Africa*, pages 195–213. Springer, 2020.

[4] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *International Conference on Cryptology in India*, pages 428–442. Springer, 2014.

[5] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 520–550. Springer, 2020.

[6] Paul Bottinelli, Victoria de Quehen, Chris Leonardi, Anton Mosunov, Filip Pawlega, and Milap Sheth. The dark SIDH of isogenies. *IACR Cryptology ePrint Archive*, 2019:1333, 2019.

[7] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology - ASIACRYPT 2018*, pages 395–427, 2018.

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

[9] Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of CRYPTOLOGY*, 22(1):93–113, 2009.

[10] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[11] Craig Costello. B-SIDH: Supersingular isogeny diffie-hellman using twisted torsion. In *ASIACRYPT (2)*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020. `https://ia.cr/2019/1145`.

[12] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006:291, 2006.

[13] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion point attacks on SIDH variants. *arXiv e-prints*, page arXiv:2005.14681, May 2020.

[14] Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016.

[15] Samuel Dobson, Steven D Galbraith, Jason LeGrow, Yan Bo Ti, and Lukas Zobernig. An adaptive attack on 2-sidh. *International Journal of Computer Mathematics: Computer Systems Theory*, 5(4):282–299, 2020.

[16] Kirsten Eisentraeger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs. *arXiv preprint arXiv:2004.11495*, 2020.

[17] Luca De Feo. Mathematics of isogeny based cryptography. *CoRR*, abs/1711.04062, 2017.

[18] 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.

[19] Steven D Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1):130–175, 2020.

[20] Alexandre Gélin and Benjamin Wesolowski. Loop-abort faults on supersingular isogeny cryptosystems. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 93–106. Springer, 2017.

[21] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation. *Updated version of [22] for round 3 of [28]*, 2020.

[22] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation. *Submission to [28]*, 2017. `https://sike.org`.

[23] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

[24] 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.

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

[26] Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Charlotte Weitkämper. One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols. *IACR Cryptol. ePrint Arch.*, 2021:282, 2021.

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

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

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

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

[31] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[32] Denis Simon. Quadratic equations in dimensions 4, 5 and more. *Preprint*, 2005. `https://simond.users.lmno.cnrs.fr/maths/Dim4.pdf`.

[33] Boris Fouotsa Tako, Péter Kutas, and Simon-Philipp Merz. On the isogeny problem with torsion point information. IACR Cryptology ePrint Archive 2021/153, `https://ia.cr/2021/153`.

[34] Yan Bo Ti. Fault attack on supersingular isogeny cryptosystems. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2017.

[35] David Urbanik and David Jao. New techniques for sidh-based nike. *Journal of Mathematical Cryptology*, 14(1):120–128, 2020.

[36] John Voight. Quaternion algebras. *preprint*, 13:23–24, 2018.