Torsion point attacks on ‘SIDH-like’ cryptosystems

Kutas, Péter; Petit, Christophe

DOI:
10.1049/ise2.12076

Received: 27 January 2022 | Revised: 28 April 2022 | Accepted: 1 June 2022

IET Information Security

DOI: 10.1049/ise2.12076

The Institution of Engineering and Technology WILEY

REVIEW

Torsion point attacks on 'SIDH-like' cryptosystems

Péter Kutas[1,2] | **Christophe Petit**[1,3]

[1]School of Computer Science, University of Birmingham, Birmingham, UK

[2]Department of Computer Algebra, Eötvös Loránd University, Budapest, Hungary

[3]Laboratoire d'Informatique, Université libre de Bruxelles, Brussels, Belgium

Correspondence

Péter Kutas, School of Computer Science, University of Birmingham, Birmingham, UK; Department of Computer Algebra, Eötvös Loránd University, Pázmány Péter sétány 1/c, Budapest 1117, Hungary.
Email: P.Kutas@bham.ac.uk

Funding information

Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/S01361X/1, EP/V011324/1; Ministry of Innovation and Technology and the National Research Development and Innovation Office; Nemzeti Kutatási Fejlesztési és Innovációs Hivatal; Quantum Information National Laboratory

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, which was submitted to and remains in the third round of NIST's post-quantum standardisation process 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 focusses 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', summarises 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 Couveignes's seminal work [1] where he introduced the notion of hard homogeneous spaces and instantiated it with ordinary elliptic curves (this scheme was independently discovered by Rostovtsev and Stolbunov [2], and thus it is referred to as the CRS scheme), and Charles, Goren and Lauter's hash function [3] (CGL) based on isogenies of supersingular elliptic curves. Jao and de Feo introduced SIDH [4] in 2011, and the field has blossomed in recent years, for example, with the introduction of CSIDH [5] (the only post-quantum scheme which provides non-interactive key exchange), SQISign and many more isogeny-based schemes. SIKE [6], which is a key encapsulation mechanism derived from SIDH, is currently a third round alternate candidate in NIST's post-quantum standardisation project.

Most isogeny-based protocols today are based on the hardness of computing isogenies between supersingular elliptic curves. However, only CGL hash function [3] and the GPS signature scheme [7] rely solely on this 'pure' isogeny problem. In SIDH protocol, parties send over torsion point images, which motivate 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 $\phi: E_0 \to E$ and given the restriction of ϕ to the B-torsion of E_0, compute ϕ.

A more specific version of the SSI-T problem is called the CSSI problem in Ref. [4]. 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, quaternion algebras and the SIDH protocol. In Section 3, we discuss active attacks on SIDH, namely the GPST attack [8] and its extensions. In Section 4, we describe 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 originating from Ref. [9] and significantly improved in Ref. [10]. In Section 6, we discuss the quantum hidden-shift attack from Ref. [11]. Finally, in Section 7, we discuss open problems which could shape the future of torsion-point attacks and their impact on the field of isogeny-based cryptography.

2 | SUPERSINGULAR ISOGENY DIFFIE–HELLMAN AND ITS VARIANTS

We refer to Ref. [12, 13] for general background on elliptic curves and isogeny-based cryptography. The following high-level description of SIDH [4] and some of its variants relevant to Problem 1.1 are taken nearly verbatim from [10, 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 $\mathbb{F}_{p^2}$ 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 $(E_B, \phi_B(P_A), \phi_B(Q_A))$ 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 [6] 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 Ref. [6] uses $p = 2^{216} \cdot 3^{137} - 1$. Other constructions belonging to the SIDH 'family tree' of protocols use different types of parameters [14–16].

We may assume knowledge of End(E_0): The only known way to construct supersingular elliptic curves is by reduction of elliptic curves with complex multiplication by a small discriminant (which implies small-degree endomorphisms: see Ref. [17, 18]), 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 [6]. Several variants of SIDH exist in the literature.

In Ref. [14], 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 ith prime number, the ith party's secret isogeny has degree $\ell_i^{e_i}$, the ith 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 j invariant 1728, and 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, Problem 1.1 must be hard when $A = \ell_1^{e_1}$ and $B = f\prod_{i=2}^{n}\ell_i^{e_i}$.

Another example of an SIDH-like scheme is B-SIDH [15]. Here, 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 j invariant) 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 [8] 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 E, E_1, E_1' are supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ and 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 computes the curve

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

The key exchange is successful if both parties compute the same curve (up to isomorphism). For simplicity, we suppose that $A = 2^n$ but the attack generalises to arbitrary smooth degree isogeny.

Remark 3.2 There is a pretty simple attack if one is allowed to send 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.

More generally, the Weil pairing of the two malicious points sent in place of $\phi_B(P_A)$ and $\phi_B(Q_A)$ must have the same value as the Weil pairing of $\phi_B(P_A)$ and $\phi_B(Q_A)$ themselves, because this value can be determined from deg ϕ_B and the pairing of P_A and Q_A only. Unfortunately, there is no way to distinguish the malicious points sent by the attacker from the actual images of P_A and Q_A under ϕ_B, as long as they have the right order and pairing value.

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:

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

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 we can compute α_k from s_k using one oracle call. It is clear that after n calls to the oracle we retrieve the static secret key α.

3.1.1 | Countermeasures

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 [6] is obtained. However, for some applications this is not desirable, namely when both parties' keys should be static.

In 2017, Azarderaksh et al. [19] introduced a variant of SIDH called k-SIDH. The main idea is the following: Alice and Bob choose k different secret isogenies 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 determined by k. Public key sizes grow linearly in k and the number of SIDH key exchanges is a quadratic function of k. In the original paper in Ref. [19] 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 Ref. [20], Dobson et al. demonstrated an attack against 2-SIDH, which generalises 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 generalise 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 [21] proposed a way of lowering the number of key exchanges by using automorphisms of the starting curves. In this way, each secret corresponds to three curves, which lowers the size of the public keys and the communication cost. However, the attack from Ref. [20] can be extended to the Jao–Urbanik scheme [22] 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 Ref. [22] 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), but the Jao–Urbanik scheme has smaller keysizes. 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 | Another adaptive attack

Another adaptive attack against SIDH (different in nature from the GPST attack) is given in Ref. [23]. The attack works in two steps.

The second step of the attack is non adaptive; the attacker simply applies (a small modification of) the shifted endomorphism attacks of Section 5. These attacks require large torsion point information, which is not normally available in an SIDH instance; the first step of the attack will collect this information through adaptive queries. One can also note that the shifted endomorphism attacks of Section 5 typically assume knowledge of the exact images of points through the secret isogeny, but Ref. [24] showed that knowing these images up to a common scalar multiple is enough.

In this first step, the attacker starts with torsion point information available in a normal execution of the SIDH protocol, and they proceed to harvest additional information. To achieve this, they (adaptively) use isogenies of degrees larger than prescribed by the protocol and observe (using the same 'oracle' as in GPST attack) how the shared key computed by the other party is affected by the changes.

As an example, to increase the order of torsion point information available by a small multiple ℓ, the attacker first computes a public key as in a normal execution of the SIDH protocol. Instead of sending that public key (made of a curve and torsion point information) to the other (static) party, the attacker then sends a neighbour of the curve and correspondingly adapted points to the other party. By observing how the shared key is affected by these changes, the attacker deduces the image of a cyclic subgroup of order $B\ell$ through the static secret isogeny. Repeating this for three distinct subgroups gives the images of all points of order $B\ell$ through the secret isogeny up to a common scalar multiple. The same procedure can then be repeated with another (possibly equal) small prime ℓ', until enough torsion point information is available to run the second step of the attack.

3.3 | 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 Ref. [25], 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 Ref. [26], 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 Ref. [26] for details.

Recently, there have been further advances in side-channel attacks (and protection) against implementations of SIDH. The reader is referred to Ref. [27–30] for more information.

4 | REDUCTION TO THE ENDOMORPHISM RING COMPUTATION PROBLEM

Computing the endomorphism ring 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 [31] and was later improved by Delfs–Galbraith [32] to a running time of $\tilde{O}(p^{1/2})$. The most recent algorithm [33] 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 [34] and has a running time of $\tilde{O}(p^{1/4})$.

It is natural to ask how finding isogenies between supersingular elliptic curves relates to computing endomorphism rings. The KLPT algorithm [35] 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 $\mathrm{Hom}(E_2, E_1)$ 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 $\phi_I^* : \mathrm{Hom}(E_1, E_2) \to I$, $\psi \mapsto \psi \circ \phi_I$ 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 computation and computing isogenies of a specific degree is from Ref.

[8]. 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. LEt O_1 and O_2 be the endomorphism rings of E_1 and E_2, respectively. Compute a connecting ideal I between O_1 and O_2 (this can be accomplished with an algorithm of Kirschmer and Voight [37]). Then find the shortest element in I using the LLL algorithm and then translate it to an isogeny between E_1 and E_2 (every step is done on the quaternion side except this last one). 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 secret isogeny has degree roughly p); thus, the algorithm from Ref. [8] fails. It is a natural ask whether one can extend the algorithm from Ref. [8] to be applicable to B-SIDH as well. In the context of B-SIDH, this question is also more important in the following sense. Since the curves in SIDH are rather close, computing the secret isogeny between them using a meet-in-the-middle approach is more efficient, then computing the endomorphism rings of both curves and applying the previous attack. In other words, in the context of SIDH, this is purely a reduction between algorithmic problems. For B-SIDH, the situation is different. Computing the endomorphism ring of the curves involved is less expensive and then running a meet-in-the-middle attack (the gap is even larger on the quantum side, as the cost of a meet-in-the-middle quantum attack is $O(p^{1/2})$ as endomorphism ring computation runs in time $O(p^{1/4})$). Thus, if one can find a polynomial-time attack on B-SIDH if both curves' endomorphism ring is known; then one has an attack on B-SIDH that performs much faster than a meet-in-the-middle routine.

The main idea of Ref. [38] is that one can exploit the torsion information provided to generalise the attack from Ref. [8] to a wide variety of parameters. Note that the algorithm in Ref. [8] did not use the torsion information at all; it solely relied on the curves being close. We sketch the attack from Ref. [38]. Similar to Ref. [8], one computes an LLL-reduced basis of a connecting ideal I (in this setting, it is also advisable to compute a smooth norm connecting ideal with KLPT as it used later on), let these be ω_1, ω_2, ω_3, ω_4. Let the corresponding isogenies be ϕ_1, ϕ_2, ϕ_3, ϕ_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 x_i modulo B by solving a system of linear equations (we omit several technical difficulties here for which the reader is referred to Ref. [38]). Why is this information useful? The reason is that an LLL-reduced basis has the property that one can bound the x_i-s using the smallest degree element in 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 $A/B < 16\sqrt{p}$. 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.

Interestingly, Wesolowski [39] has recently shown that one can compute the secret isogeny in CSIDH in polynomial time if the endomorphism ring of both curves is known. Previously, only a subexponential reduction was given in Ref. [40].

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, all previously described algorithms use some extra information, namely closeness of the curves, torsion-point information or the curves are defined over $\mathbb{F}_p$.

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 [9], which was first improved in Ref. [41] and then further improved in Ref. [10].

5.1 | Petit's attack

The starting point is the following. Let ϕ: $E_1 \to E_2$ be an isogeny of degree A, and suppose we know the action of ϕ on the B-torsion. Let $\theta \in$ 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 Ref. [9] 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 exhaustive search (or a meet-in-the-middle algorithm if e is composite). Finally, one can obtain $\ker(\hat{\phi})$ 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 $\theta \in$ End (E_1) is then equivalent to finding an integer solution (a, b, c, d, e) with small e to the following equation:

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

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 $c^2 \equiv \frac{Be-d^2}{A^2} \pmod{p}$. 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 Ref. [9], 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.

5.2 | The dual isogeny and the Frobenius attack

Follow-up papers improve on Petit's original algorithm by relaxing the condition on θ and relating the algorithm to different equations. In Ref. [41], the authors use triangular decompositions and certain endomorphisms with many eigenvalues to derive the following equation:

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

In Ref. [10], 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 $\deg\left(\phi \circ \theta \circ \hat{\phi} + [d]\right) = B^2 e$, 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 a priori, 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 [15] because there the parameters are balanced.

The main idea of the Frobenius approach outlined in Ref. [10] 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 for the p component and using exhaustive search (or meet-in-the-middle) for the rest. This results in an alternative equation:

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

One can solve this equation by first setting $c = 0$ and $d = pd'$ and dividing by p, leading to

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

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 $\frac{B^2 e - pd'^2}{A^2}$ 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 only motivated by having the same security level for Alice and Bob. In many SIDH applications, the parameters are not balanced [42, 43] and future protocols may arise using unbalanced parameters.

5.3 | Exponential-time attacks

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 summarised in Figure 1. Even though Figure 1 is quite comprehensive we highlight the state of the art on SIDH-like parameters:

– Assuming the starting curve has j-invariant 1728, torsion-point attacks outperform classical meet-in-the-middle attacks whenever $B > A(3 + \epsilon)$ for any $\epsilon > 0$.

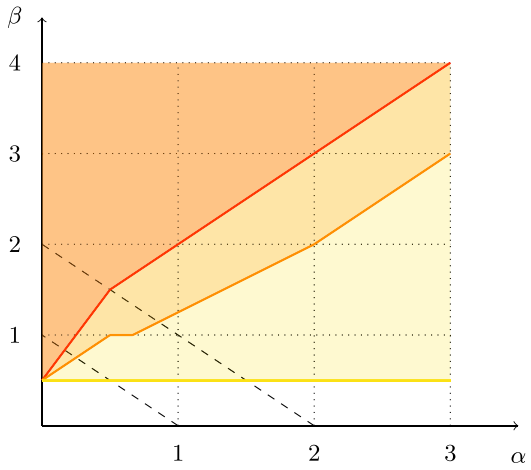

FIGURE 1 Performance of attacks from Ref. [10]. Here $A \approx p^{\alpha}$ and $B \approx p^{\beta}$. Parameters (α, β) above the red, orange and yellow curves are parameters admitting a polynomial-time attack, an improvement over the best classical attacks, and an improvement over the best quantum attacks, respectively. Parameters below the upper dashed line are those allowing $AB|(p^2 - 1)$ as in Ref. [15]. Parameters below the lower dashed line are those allowing $AB|(p - 1)$ as in Ref. [6, 43]

– Assuming the starting curve has j-invariant 1728, torsion-point attacks outperform quantum meet-in-the-middle attacks whenever $B > A(1 + \epsilon)$ for any $\epsilon > 0$.

5.4 | Backdoor attacks

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 with unknown endomorphism ring 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). 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 non-trivial 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 [44]. One has now found θ but not the curve. The curves 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 Ref. [10], the curves containing such a θ are called (A, B)-backdoor curves. The number of these curves is exponential in $\log p$.

The condition for the existence of such a curve is $B > A^2$, so in particular it is independent of p. The above *polynomial time* attack still requires unbalanced SIDH parameters, but non-polynomial time generalisations can be faster than meet-in-the-middle algorithms for balanced SIDH parameters [10]. An application of this idea can be found in Ref. [24] where they analyse the security of a recently proposed oblivious pseudo-random function [42]. They show that in order to avoid backdoor attacks (or failures of the security proof), it is advisable to use a trusted party to generate a random starting curve. Interestingly, there is also a constructive application of backdoor curves as they can be used as a trapdoor mechanism. In Ref. [45], the authors propose Séta, an isogeny-based PKE, which relies on the hardness of finding a certain quadratic order (corresponding to the previous backdoor) in the endomorphism ring of the starting curve.

This seems to suggest that against all intuition it is probably safer to instantiate SIDH starting from $y^2 = x^3 + x$, than from a random curve if there is no guarantee that the curve was honestly generated. 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, that is, 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 realises the Hard Homogeneous Space concept of Couveignes [1]. In the SIDH setting, one does not have a 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 the 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 $f: I \to O$ 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 Ref. [11], it is shown that if one has access to a malleability oracle, then one can invert f in quantum subexponential time. It is also shown that this framework applies to CSIDH and is essentially the same attack as the one proposed by Childs, Jao and Soukharev [46]. 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 $f: I \to O$ 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 $(O/\mathrm{AO})^*$ 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 technical problems that have an easy solution. 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 $\phi : E \to E_X$ 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 focussing 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 on 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.

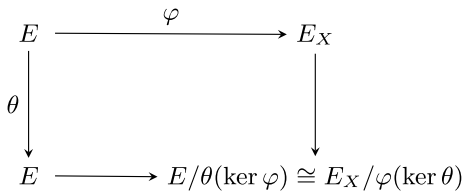

FIGURE 2 Supersingular isogeny Diffie–Hellman (SIDH) key exchange instance with isogenies φ and the endomorphism θ

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. A particular choice for which this is 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 than 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.

7 | OPEN PROBLEMS

There are various open problems that remain. Probably the more interesting questions are 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 Ref. [10], 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 utilises 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 generalise 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 [8], continuing with torsion point passive attacks [9, 10] 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 [11]. These attacks have improved over time: while Ref. [9] only worked for very unbalanced parameters, the latest improvements from Ref. [10] 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 Ref. [14] for any number of parties greater than 6. The future will tell whether these and other ideas will eventually affect the security of SIKE.

ACKNOWLEDGEMENTS

Péter Kutas and Christophe Petit's work were supported by EPSRC grants EP/S01361X/1 and EP/V011324/1. Péter Kutas was also supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary.

CONFLICT OF INTEREST

We do not have any conflicts of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Péter Kutas https://orcid.org/0000-0002-2043-9542

REFERENCES

1. Couveignes, JM.: Hard homogeneous spaces. IACR Cryptol Eprint Arch. 291, (2006)
2. Alexander, R., Stolbunov, A.: Public-key Cryptosystem Based on Isogenies. Cryptology ePrint Archive (2006)
3. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptol. 22(1), 93–113 (2009). https://doi.org/10.1007/s00145-007-9002-x
4. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: International Workshop on Post-Quantum Cryptography, pp. 19–34. Springer (2011)
5. Castryck, W., et al.: CSIDH: an efficient post-quantum commutative group action. In: Advances in Cryptology - ASIACRYPT 2018, pp. 395–427 (2018)
6. Jao, D., et al.: Supersingular isogeny key encapsulation. Updated version of [30] for round 3 of [37] (2020)
7. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. J. Cryptol. 33(1), 130–175 (2020). https://doi.org/10.1007/s00145-019-09316-0
8. Galbraith, S.D., et al.: On the security of supersingular isogeny cryptosystems. In: ASIACRYPT (1), Volume 10031 of LNCS, pp. 63–91 (2016)
9. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: ASIACRYPT (2), Volume 10625 of LNCS, pp. 330–353. Springer (2017)
10. de Quehen, V., et al.: Improved Torsion Point Attacks on SIDH Variants (2020). arXiv e-prints, page arXiv:2005.14681
11. Kutas, P., et al.: One-way functions and malleability oracles: hidden shift attacks on isogeny-based protocols. IACR Cryptol. ePrint Arch. 282 (2021)
12. Silverman, J.H.: The Arithmetic of Elliptic Curves, Volume 106. Springer Science & Business Media (2009)
13. De Feo, L.: Mathematics of isogeny based cryptography. CoRR (2017)
14. Azarderakhsh, R., et al.: Practical supersingular isogeny group key agreement. IACR Cryptol ePrint Arch 330 (2019)
15. Craig Costello, B.-SIDH: Supersingular isogeny diffie-hellman using twisted torsion. In: ASIACRYPT (2), Volume 12492 of Lecture Notes in Computer Science, pp. 440–463. Springer (2020). https://ia.cr/2019/1145
16. Anand Sahu, R., Gini, A., Pal, A.: Supersingular isogeny-based designated verifier blind signature. IACR Cryptol ePrint Arch 1498 (2019)
17. Castryck, W., Lorenz, P., Vercauteren, F.: Rational isogenies from irrational endomorphisms. In: EUROCRYPT (2), Volume 12106 of LNCS, pp. 523–548. Springer (2020)
18. Love, J., Boneh, D.: Supersingular Curves with Small Non-integer Endomorphisms (2019). arXiv preprint arXiv:1910.03180
19. Azarderakhsh, R., Jao, D., Leonardi, C.: Post-quantum static-static key agreement using multiple protocol instances. In: International Conference on Selected Areas in Cryptography, pp. 45–63. Springer (2017)
20. Dobson, S., et al.: An adaptive attack on 2-sidh. Int. J. Comput. Math.: Comput. Syst. Theory 5(4), 282–299 (2020). https://doi.org/10.1080/23799927.2020.1822446
21. Urbanik, D., Jao, D.: New techniques for sidh-based nike. J. Math. Cryptol. 14(1), 120–128 (2020). https://doi.org/10.1515/jmc-2015-0056
22. Basso, A., et al.: On adaptive attacks against jao-urbanik's isogeny-based protocol. In: International Conference on Cryptology in Africa, pp. 195–213. Springer (2020)
23. Boris Fouotsa, T., Petit, C.: A New Adaptive Attack on Sidh. Cryptology ePrint Archive (2021)
24. Basso, A., et al.: Cryptanalysis of an Oblivious Prf from Supersingular Isogenies
25. Gélin, A., Wesolowski, B.: Loop-abort faults on supersingular isogeny cryptosystems. In: Lange, T., Takagi, T. (eds.) Post-quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings, Volume 10346 of Lecture Notes in Computer Science, pp. 93–106. Springer (2017)
26. Ti, Y.Bo: Fault attack on supersingular isogeny cryptosystems. In: Lange, T., Takagi, T. (eds.) Post-quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings, Volume 10346 of Lecture Notes in Computer Science, pp. 107–122. Springer (2017)
27. Gora, A., et al.: Faulty isogenies: a new kind of leakage. arXiv preprint arXiv:2202.04896, (2022)
28. De Feo, L., et al.: Sike Channels. Cryptology ePrint Archive (2022)
29. Genêt, A., de Guertechin, N.L., Novak, K.: Full key recovery side-channel attack against ephemeral sike on the cortex-m4. In: International Workshop on Constructive Side-Channel Analysis and Secure Design, pp. 228–254. Springer (2021)
30. Tasso, É., et al.: Resistance of isogeny-based cryptographic implementations to a fault attack. In: International Workshop on Constructive Side-Channel Analysis and Secure Design, pp. 255–276. Springer (2021)
31. Russell Kohel, D.: Endomorphism rings of elliptic curves over finite fields. PhD thesis. University of California, Berkeley (1996)
32. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Des. Codes Cryptogr. 78(2), 425–440 (2016). https://doi.org/10.1007/s10623-014-0010-1
33. Eisentraeger, K., et al.: Computing Endomorphism Rings of Supersingular Elliptic Curves and Connections to Pathfinding in Isogeny Graphs (2020). arXiv preprint arXiv:2004.11495
34. Biasse, J.-F., Jao, D., Sankar, A.: A quantum algorithm for computing isogenies between supersingular elliptic curves. In: International Conference on Cryptology in India, pp. 428–442. Springer (2014)
35. Kohel, D., et al.: On the quaternion ℓ-isogeny path problem. LMS J. Comput. Math. 17A(A), 418–432 (2014). https://doi.org/10.1112/s1461157014000151

36. Voight, J.: Quaternion algebras. Preprint. 13, 23–24 (2018)
37. Kirschmer, M., Voight, J.: Algorithmic enumeration of ideal classes for quaternion orders. SIAM J. Comput. 39(5), 1714–1747 (2010). https://doi.org/10.1137/080734467
38. Boris Fouotsa, T., et al.: On the isogeny problem with torsion point information. In: IACR International Conference on Public-Key Cryptography, pp. 142–161. Springer (2022)
39. Wesolowski, B.: Orientations and the supersingular endomorphism ring problem. Cryptol ePrint Archi (2021)
40. Castryck, W., Lorenz, P., Vercauteren, F.: Rational isogenies from irrational endomorphisms. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 523–548. Springer (2020)
41. Paul, B., et al.: The dark SIDH of isogenies. IACR Cryptol ePrint Arch 1333 (2019)
42. Boneh, D., Kogan, D., Woo, K.: Oblivious pseudorandom functions from isogenies. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 520–550. Springer (2020)
43. Jao, D., et al.: Supersingular Isogeny Key Encapsulation (2017). Submission to [37] https://sike.org
44. Simon, D.: Quadratic Equations in Dimensions 4, 5 and More. Preprint (2005). https://simond.users.lmno.cnrs.fr/maths/Dim4.pdf
45. De Feo, L., et al.: Séta: supersingular encryption from torsion attacks. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 249–278. Springer (2021)
46. Childs, A., Jao, D., Soukharev, V.: Constructing elliptic curve isogenies in quantum subexponential time. J. Math. Cryptol. 8(1), 1–29 (2014). https://doi.org/10.1515/jmc-2012-0016

How to cite this article: Kutas, P., Petit, C.: Torsion point attacks on 'SIDH-like' cryptosystems. IET Inf. Secur. 1–10 (2022). https://doi.org/10.1049/ise2.12076