Generalising Fault Attacks to Genus Two Isogeny Cryptosystems

Ariana Goh, Chu-Wee Lim, and Yan Bo Ti

DSO National Laboratories, Singapore

ari.gzh@gmail.com, lchuwee@dso.org.sg, yanbo.ti@gmail.com

Abstract

In this paper, we generalise the SIDH fault attack and the SIDH loop-abort fault attacks on supersingular isogeny cryptosystems (genus-1) to genus-2. Genus-2 isogeny-based cryptosystems are generalisations of its genus-1 counterpart, as such, attacks on the latter are believed to generalise to the former.

The point perturbation attack on supersingular elliptic curve isogeny cryptography has been shown to be practical. We show in this paper that this fault attack continues to be practical in genus-2, albeit with a few additional traces required. We also show that the loop-abort attack carries over to the genus-2 setting seamlessly.

This article is a minor revision of the version accepted to the workshop Fault Diagnosis and Tolerance in Cryptography 2022 (FDTC 2022).

Isogeny-based cryptography was first proposed in 1997 by Couveignes in an unpublished manuscript [Cou06] on hard homogeneous spaces. This was re-discovered independently by Rostovtsev and Stolbunov in 2006 [RS06]. In 2011, Jao and De Feo introduced the supersingular isogeny Diffie–Hellman (SIDH) protocol [JF11]. This went on to form the basis of SIKE in 2017 which is a fourth-round candidate in NIST's post-quantum standardisation process. However, a flurry of recent results has since broken SIDH [**?**,**?**,**?**]. The isogenies of supersingular elliptic curves have also found use in creating a hash function in 2005 by Charles, Goren, and Lauter [CLG09]. These cryptosystems are dependent on the difficulty of finding isogenies between supersingular elliptic curves.

This hard problem of finding isogenies between elliptic curves is not unique to abelian varieties of dimension one, i.e. elliptic curves. One can generalise the SIDH protocol to abelian varieties of higher dimension, which is what was proposed by Flynn and Ti in [FT19]. They presented the G2SIDH protocol which generalises SIDH to genus-2 and bases G2SIDH on the difficulty of finding isogenies between principally polarised superspecial abelian surfaces. Furthermore, Takashima [Tak17] proposed a generalisation of the CGL hash function to genus-2 in 2018, and this was improved by Castryck, Decru, and Smith in 2019 which patches the vulnerability that causes collisions in the Takashima hash observed in [FT19]. Cryptanalysis of higher general cryptosystems has been conducted by Costello and Smith in [CS20], and Kunzweiler, Ti, and Weitkämper in [KTW21].

In this paper, we take a different approach and outline physical attacks that are able to recover the secret keys of G2SIDH. Physical attacks have a long history in cryptography and have also been proposed against isogeny-based attacks in the literature [GW17, KAJ17, Ti17, KPHS18, CKM+20, ZYD+20, CKM21, XIU+21, ACDMRH22, UXT+22].

The method that we employ is fault attacks. The analogous fault attacks on SIDH that we are interested in can be classified as loop-abortion techniques as well as point perturbation. The former method of Gélin and Wesolowski [GW17] relies on using a fault to disrupt loops in the SIDH algorithm to force a system to output intermediate values which correspond to intermediate curves along the secret isogeny. The latter attack of Ti [Ti17] aims to perturb one of the auxiliary points of the protocol. This forces the system to compute images of random points through the secret isogeny. The attack exploits this fact and uses the result of this unintended computation to recover the secret isogeny with several traces.

This paper describes the generalisation of both fault attack methods to the G2SIDH protocol. We begin by outlining the preliminaries in Section 1 and sketching out the G2SIDH protocol in Section 1.3. Next, the generalisation of Ti's fault attack to G2SIDH will be detailed in Section 2, and we provide an analysis of this attack in Section 3. And we will elucidate the loop-abort attack on G2SIDH in Section 4. Lastly, we will discuss the implications of this attack on current genus-2 cryptosystems in Section 5.

Acknowledgements

We would like to thank Daren Khu for helpful discussions. We would also like to thank anonymous reviewers of ANTS and FTDC for helpful feedback.

1 Preliminaries/Background

This section serves as a concise introduction to the central characters of genus-2 isogeny-based cryptography. We will only cover the necessary topics and will set out the notation to be used in the paper.

1.1 Abelian Surfaces

We consider *abelian surfaces*, i.e. abelian varieties A of dimension 2, over $\mathbb{F}_q$ where q is a power of a prime $p > 2$. Given A, we have the *dual abelian variety* $A^\vee$, together with an invertible sheaf $\mathcal{P}$ on the product $A \times A^\vee$. An isogeny $\lambda : A \to A^\vee$ is called a *polarisation* of A; the polarisation λ is *principal* if it is an isomorphism.

We consider the principally polarised abelian surfaces (PPAS), which are pairs of (A, λ) where A is an abelian surface and $\lambda : A \to A^\vee$ is a principal polarisation. As shown in [GGR05, Thm. 3.1], a principally polarised abelian surface over $\overline{\mathbb{F}}_p$ is isomorphic to one of the following:

(a) the Jacobian $J(C)$ of a smooth projective curve C of genus-2, or

(b) the product of two elliptic curves $E \times E'$,

where $J(C)$ is equipped with its canonical principal polarisation.

For each n prime to p, we have $A[n] \cong C_n^4$ (where C_n is the cyclic group of order n), and the *Weil pairing*

$$e_n : A[n] \times A^\vee[n] \longrightarrow \mu_n$$

which is bi-additive, alternating and Galois-invariant. Via the polarisation λ, we obtain a pairing $e_n : A[n] \times A[n] \to \mu_n$.

A principally polarised abelian surface is said to be *superspecial* if it is isomorphic over $\overline{\mathbb{F}}_p$ to a product of supersingular elliptic curves *as abstract abelian varieties*. In addition, all such products are isomorphic to each other. As an analogue of supersingular elliptic curves, our objects of interest are the principally polarised superspecial abelian surfaces (PPSSAS).

1.2 Mumford representation

We consider a PPSSAS which is the Jacobian of a smooth projective curve C of genus-2. Such a C is also said to be *superspecial*. It is hyperelliptic, so we can write it in the form $y^2 = f(x)$ for some $f(x) \in \mathbb{F}_q[x]$ of degree 5 or 6.

Lemma 1.1 ([Gal12, Lem. 10.3.3]). *Let $C' = C \cap \mathbb{A}^2$. Any non-zero divisor $D \in \mathrm{Pic}^0(C)$ is linearly equivalent to a divisor of one of the following forms:*

- *$P_1 - \infty$ with $P_1 \in C'$, or*
- *$P_1 + P_2 - 2\infty$ with $P_1, P_2 \in C'$ such that $\iota(P_1) \neq P_2$, where $\iota : C \to C$ is the hyperelliptic involution.*

Such a divisor is said to be semi-reduced.

Given a semi-reduced divisor D, consider the polynomial $u(x) = \prod_i (x - x_{P_i})$, which is of degree 1 or 2. Then there is a unique $v(x) \in \overline{\mathbb{F}}_p[x]$ satisfying $\deg v(x) < \deg u(x)$ and $v(x_{P_i}) = y_{P_i}$ for each i, such that

$$v(x)^2 \equiv f(x) \pmod{u(x)}.$$

The pair $(u(x), v(x))$ is called the *Mumford representation* of the semi-reduced divisor D. Furthermore, D is defined over $\mathbb{F}_q$ if and only if $u(x), v(x) \in \mathbb{F}_q$.

For a general $D \in \mathrm{Pic}^0(C)$, one first finds a semi-reduced $D' \in \mathrm{Pic}^0(C)$ which is linearly equivalent to D, then computes its Mumford representation. Although D' is not unique, its Mumford representation is. Given Mumford representations of divisors $D_1, D_2 \in \mathrm{Pic}^0(C)$, one can apply Cantor's algorithm [Gal12, Thm. 10.3.14] to compute the Mumford representation of $D_1 + D_2$.

1.3 G2SIDH

Let A be a PPSSAS. If $G \subseteq A$ is a finite subgroup of order prime to p, we obtain an abelian variety A/G with a canonical separable isogeny $A \to A/G$ of kernel G. We will only consider the case where G is *proper*, i.e. it does not contain $A[n]$ for any $n > 1$. Note that there is no loss of generality since $A/(A[n]) \cong A$ if n is coprime to p.

Definition 1.2. *Let $m \geq 1$ be prime to p and $G \subseteq A[m]$ be a proper subgroup. We say G is maximal m-isotropic if*

(a) the Weil pairing $e_m : A[m] \times A[m] \to \mu_m$ restricts trivially to G, and

(b) G is not properly contained in any subgroup of $A[m]$ satisfying (1).

The significance of such subgroups is given by the following.

Proposition 1.3 ([FT19]). *Let A be PPSSAS and $G \subseteq A$ be a finite proper subgroup of order prime to p. The polarisation for the abelian surface A/G is principal if and only if G is a maximal m-isotropic subgroup for some $m \geq 1$, in which case it is also superspecial (hence a PPSSAS).*

Let $\ell \neq p$ be a small prime. In the case where $m = \ell^n$, we have the following.

Lemma 1.4 ([FT19]). *The maximal ℓ^n-isotropic subgroups of $A[\ell^n]$ are isomorphic to either*

$$C_{\ell^n} \times C_{\ell^n}, \quad \text{or} \quad C_{\ell^n} \times C_{\ell^{n-k}} \times C_{\ell^k}$$

for some $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$. In particular, each $G \cong C_\ell \times C_\ell \subseteq A[\ell]$ is a maximal ℓ-isotropic subgroup. The resulting $A \to A/G$ is called an (ℓ, ℓ)-isogeny.

Now we may describe our isogeny graph for the protocol for G2SIDH. As in SIDH, we fix a small prime $\ell \neq p$, say $\ell \in \{2, 3\}$, and consider the graph $\mathcal{G}_{p,\ell}$ whose vertices are isomorphism classes of PPSSAS, and edges are (ℓ, ℓ)-isogenies. Note that if $A \to B$ is an (ℓ, ℓ)-isogeny, so is the dual $B^\vee \to A^\vee$ and composing with their respective polarisations gives an (ℓ, ℓ)-isogeny $B \to A$. Thus the graph is undirected.

As noted in [FT19], the graph $\mathcal{G}_{p,\ell}$ is not collision resistant due to the prevalence of "diamonds". In the same paper, the authors describe G2SIDH, a genus-2 variant of the SIDH key-exchange protocol based on $\mathcal{G}_{p,\ell}$ for $\ell = 2, 3$. The presence of diamonds does not affect the security of G2SIDH.

1. Pick a prime of the form $p = 2^{e_A} \cdot 3^{e_B} \cdot f - 1$, where 2^{e_A} and 3^{e_B} are of comparable length.

2. Pick a PPSSAS A over $\mathbb{F}_{p^2}$. One can, for example, start from a fixed superspecial genus-2 curve, then traverse randomly along $\mathcal{G}_{p,2}$. In [KTW21], the authors suggest taking $y^2 = x^6 + 1$ for the starting curve.

3. To traverse along $\mathcal{G}_{p,\ell}$, select random points $P_1, P_2, P_3 \in A[\ell^n]$ such that $G = \langle P_1, P_2, P_3 \rangle$ is a maximal ℓ^n-isotropic subgroup of $A[\ell^n]$. To achieve this, one can proceed as in [KTW21] with a symplectic basis of $A[\ell^n]$ with respect to the Weil pairing, then obtain G via a sequence of (ℓ, ℓ)-isogenies of length n.

4. Now the key exchange proceeds as in SIDH. Alice and Bob agree on a PPSSAS, as in step (2). Alice then chooses a secret maximal 2^{e_A}-isotropic subgroup of $A[2^{e_A}]$ and computes a symplectic basis (P_1, P_2, P_3, P_4) of $A[2^{e_A}]$. Bob chooses a secret maximal 3^{e_B}-isotropic subgroup of $A[3^{e_B}]$ and computes a symplectic basis (Q_1, Q_2, Q_3, Q_4) of $A[3^{e_B}]$. Each of them sends their respective basis to the other party.

5. From her basis, Alice proceeds as in step (3) to obtain a random maximal 2^{e_A}-isotropic subgroup G_A, with corresponding isogeny $\phi_A : A \to A/G_A =: J_A$. She then sends the tuple

$$(J_A, \phi_A(Q_1), \phi_A(Q_2), \phi_A(Q_3), \phi_A(Q_4))$$

   to Bob. One can show that the resulting basis $\{\phi_A(Q_i) : 1 \leq i \leq 4\}$ of $J_A[3^{e_B}]$ is symplectic.

6. Similarly, Bob obtains a random maximal 3^{e_B}-isotropic subgroup G_B, with corresponding isogeny $\phi_B : A \to A/G_B =: J_B$. He sends the tuple

$$(J_B, \phi_B(P_1), \phi_B(P_2), \phi_B(P_3), \phi_B(P_4))$$

   to Alice.

7. Now Alice computes $\phi_B(G_A)$ from the symplectic basis $\{\phi_B(P_i) : 1 \leq i \leq 4\}$ of $J_B[2^{e_A}]$ and obtains the PPSSAS $J_{AB} := J_B/\phi_B(G_A)$. Similarly, Bob obtains $J_{BA} := J_A/\phi_A(G_B)$. One then has $J_{AB} \cong J_{BA}$.

The resulting PPSSAS J_{AB} is, with overwhelming probability, the Jacobian of a genus-2 curve. Quantitatively, there are $O(p^3)$ such cases but only $O(p^2)$ products of two supersingular elliptic curves. Hence Alice and Bob can derive a shared key by computing the genus-2 curve C_{AB} such that $J_{AB} \cong J(C_{AB})$. Since such curves form a moduli space of dimension 3, one can describe C_{AB} by a set of three parameters. An explicit description of these invariants was computed by Igusa [Igu60] and by Cardona, Quer, Nart and Pujolàs [CNP02, CQ05].

1.4 Fault attack on SIDH

Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Recall that Alice computes a secret isogeny $\phi_A : E \to E_A$ whose kernel is a cyclic subgroup of order 2^{e_A}. Suppose the attacker Eve injects a fault into the system, causing it to perturb a point in E to some $X \in E(\mathbb{F}_{p^2})$. This is possible since SIDH computations only use the x-coordinates of points; with probability 50%, a random change of this coordinate results in a rational point on the curve. From the SIDH protocol, Eve gains access to the point $\phi_A(X)$. Multiplying X by a suitable scalar, we may assume without loss of generality that $X \in E[2^{e_A}]$.

First, consider the ideal case where X has order 2^{e_A} exactly; this occurs with a probability of 50%. If P' denotes a generator of $\ker \phi_A$, then $\langle P', X \rangle$ almost certainly generates the whole of $E[2^{e_A}]$. We obtain:

$$E_A/\langle \phi_A(X) \rangle \cong E/\langle P', X \rangle = E/E[2^{e_A}] \cong E$$

so the isogeny $E_A \to E_A/\langle \phi_A(X) \rangle$ is dual to ϕ_A. This enables Eve to obtain Alice's secret isogeny ϕ_A.

More generally, if X has order $2^{e_A - k}$ for a small k, then Eve computes $E_A \to E_A/\langle \phi_A(X) \rangle =: E'$ and $E'/\langle Y \rangle \cong E$ for some rational point $Y \in E'$ of order 2^k. This has only 2^{2k} possibilities and Eve can easily launch a brute-force attack to recover the dual isogeny $E_A \to E' \to E'/\langle Y \rangle$.

2 Fault attack

The idea of the fault attack is that it forces the protocol to output the image of a random divisor under the secret isogeny, and given enough of such random images, we can recover the secret isogeny. By using certain "reduced" representations of divisors (see next paragraph), a fault is likely to give us a valid divisor. Hence the adversary may assume after enough faults, they would obtain the image of a valid random divisor under the secret isogeny, and hence recover the secret isogeny by Lemma 2.4.

Recall that the Mumford representation represents a divisor $P+Q-2\infty$ with polynomials u, v where the roots of u are the x-coordinates of $P, Q \in A(\mathbb{F}_q)$ and $y_\star = v(x_\star)$, $\star \in \{P, Q\}$. However v carries a lot more information than necessary, instead, one could simply choose an agreed ordering of $\mathbb{F}_{q^2}$ and represent the y-coordinate via a 0 or 1, representing the two roots of the hyperelliptic curve equation. The added benefit of this representation is the reduction of message sizes in any protocol. Also, this means that whenever we apply a fault to a divisor, we would either flip the y-coordinate bit or alter the polynomial u. As u is monic, the fault can only affect the x^0 and x^1 coefficients.

Half the time after faulting, the roots of u lie in $\mathbb{F}_q$. In this case, there is a roughly $\frac{1}{2}$ chance each x-coordinate yields a y-coordinate in $\mathbb{F}_q$ via the hyperelliptic curve, which gives us a total probability of this case of $\frac{1}{8}$.

The other half, the roots lie in $\mathbb{F}_{q^2}$ and there's a $\frac{1}{2}$ chance that they yield a y-coordinate in $\mathbb{F}_{q^2}$. The divisor for such as case is of the form $P + P^{\mathrm{Frob}_{\mathbb{F}_{q^2}/\mathbb{F}_q}} - 2\infty$, which tells us that there is a $\frac{1}{2}$ chance in this case as the y-coordinate bit for both points gives us such a divisor, hence this case contributes a probability of $\frac{1}{8}$.

In total, this tells us that there is a $\frac{1}{4}$ chance that the faulted point is a valid divisor.

2.1 Technical results for the recovery of isogeny from image of random points

The first two results are well-known in the literature and are stated here for completeness. We will use this proposition and the next theorem for the other results to come.

Proposition 2.1 ([MRM74, Prop. II.6.3]). *Let A be an abelian variety of dimension g, we have*

$$A[n] \cong \begin{cases} C_n^{2g} & p \nmid n \\ C_n^{i} & n = p^k \end{cases} \quad 0 \le i \le 2g\,.$$

Theorem 2.2 ([MRM74, Thm. II.7.4]). *Let A be an abelian variety. Then there is a bijective correspondence between finite subgroups $K \subset A$ and separable isogenies $f : A \to B$ where isogenies $f_1 : A \to B_1$ and $f_2 : A \to B_2$ are considered equivalent if there exists an isomorphism $g : B_1 \to B_2$ such that $g \circ f_1 = f_2$.*

This correspondence is given by sending each finite subgroup $K \subset A$ to $f : A \to A/K$ and each separable isogeny $f : A \to B$ to $\ker f \subset A$.

The next result is the first of our technical lemmas. It shows that we are able to recover a secret isogeny if we are provided with images of a torsion subgroup that contains the kernel of the isogeny.

Lemma 2.3. *Suppose there exists a separable isogeny $\phi : A \to B$ of abelian varieties of dimension 2 over $\overline{\mathbb{F}}_p$ with $\ker\phi \subset A[\ell^e]$ where ℓ is a prime distinct from p and we are given $A, B, \phi\,(A[\ell^e])$. Then we can recover ϕ by brute-forcing at most $16e$ isogenies.*

Proof. By Theorem 2.2, we know that there exists a separable isogeny $\psi : B \to B/\left(\phi(A[\ell^e])\right) = A$ such that $\ker\psi = \phi(A[\ell^e])$ and $\psi \circ \phi = [\ell^e]$.

Define $A_i = B/\left([\ell^{e-i}]\phi(A[\ell^e])\right)$ the isogenies ψ_i by $\psi_i : A_{i-1} \to A_i$ via the quotient

$$A_{i-1} \to A_{i-1}/\left((\psi_{i-1} \circ \psi_{i-2} \cdots \circ \psi_1)\left([\ell^{e-i}]\phi(A[\ell^e])\right)\right) = A_i.$$

Next, find isogenies $\phi_i : A_i \to A_{i-1}$ such that $\phi_i \circ [\ell^{i-1}] \circ \psi_i = [\ell^i]$. Importantly, $\ker\phi_i \subset \ker[\ell]$, hence there are at most 16 isogenies to search through since there are 16 subgroups of $C_\ell^{2\times 2} = A_i[\ell]$. Finally define $\phi = \phi_1 \circ \cdots \circ \phi_{e-1} \circ \phi_e$.

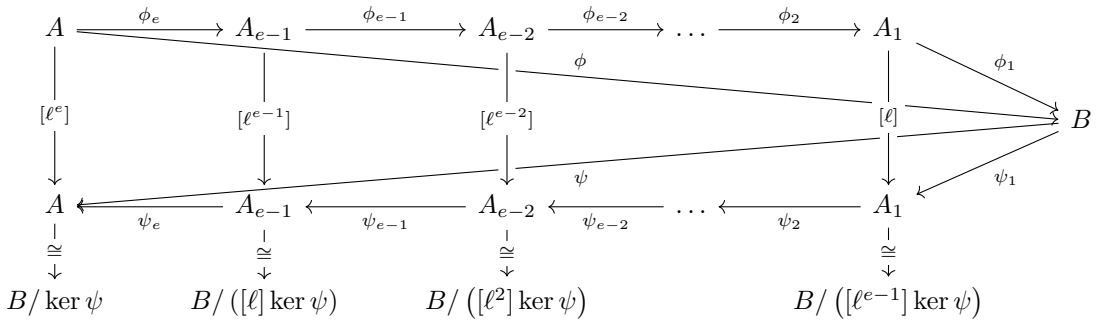

Figure 1: Isogeny ϕ from A to B is the sequence on the top. The isogeny ψ from B to A is the sequence at the bottom.

This procedure is illustrated in the diagram in Figure 1.

□

In the preceding lemma, we saw that knowledge of the images of the torsion subgroup is sufficient for us to recover the secret isogeny. However, this may not be practical. As such, this next result computes the amount of information we will need to recover the secret isogeny.

Lemma 2.4. *Suppose we have a separable isogeny $\phi : A \to B$ of abelian varieties of dimension 2 over $\overline{\mathbb{F}}_p$ such that $\ker\phi \subset A[\ell^e] \subset A(\mathbb{F}_{p^r})$ and $\phi(A[\ell^e])$ has m generators where ℓ is a prime distinct from p. Given the images of $n \geq m$ random points $P_i \in A(\mathbb{F}_{p^r})$ under ϕ, there is a probability of at least $\left(1-\ell^{-1}\right)^m$ that the group $\phi(A[\ell^e])$ is generated by $[h]\phi(P_i)$ where $h = \ell^{-2e}\left|A(\mathbb{F}_{p^r})\right|$.*

Proof. Note that h is an integer as $A[\ell^e]$ is a subgroup of $A(\mathbb{F}_{p^r})$ and $|A[\ell^e]| = \ell^{2e}$.

Since $[h]P_i \in A[\ell^e]$, $[h]\phi(P_i) = \phi([h]P_i)$ gives us a random point in $\phi(A[\ell^e])$. With sufficiently many random points, it is possible that we generate all of $\phi(A[\ell^e])$.

The probability that we generate $\phi(A[\ell^e])$ is given by Theorem 3.2 of the next section. □

Remark 2.5. *We note that Lemma 2.3 works in the more general case where $\ker\psi \subset \phi(A[\mathcal{N}])$ for $p \nmid \mathcal{N}$ by applying the lemma to each prime factor of $\mathcal{N}$. The lemma also generalizes to higher dimensions, however, the isogeny and quotient computations may be difficult.*

Lemma 2.4 also generalizes directly to higher dimensions by setting $h = \ell^{-ge}\left|A(\mathbb{F}_{p^r})\right|$ where g is the dimension of the abelian varieties.

Remark 2.6. *We can know precisely when we have sufficiently many points with the following procedure. Choose a minimal set of generators Q_j of $B[\ell^e]$, then we can express every point $[h]P_i$ by $\sum_{j=1}^{2g} M_{i,j}Q_j$, where $M_{i,j} \pmod{\ell^e}$ is unique. Let $UDV = M$ be a Smith normal form of M as a matrix over $\mathbb{Z}_\ell$.*

Since D is diagonal and U and V are invertible, we can easily read out the size of the subgroup generated by $[h]P_j$. Letting $\{a_i\}_{i=1}^{m'}$ be invariant factors of M that has ℓ-adic valuation less than e, then we get

$$\left|\langle [h]P_1, \dots, [h]P_n\rangle\right| = \ell^{m'e}\prod_{i=1}^{m'} p^{-v_\ell(a_i)}.$$

By comparing this value with $|\phi(A[\ell^e])|$, we can find the index of the subgroup generated by $[h]P_j$ in $\phi(A[\ell^e])$. If the index is 1, then we know that $[h]P_j$ generates the entire group.

Furthermore, if the random points do not generate $A[\ell^e]$ but $[\phi(A[\ell^e]) : \langle [h]P_1, \dots, [h]P_n\rangle]$ is small, we can brute-force for $\phi(A[\ell^e])$ as the number of groups containing $[h]P_n$ of a small index in $C_{\ell^e}^{2g}$ is at most $[\phi(A[\ell^e]) : \langle [h]P_1, \dots, [h]P_n\rangle]^{2g}$.

In the examples, we shall assume that each fault is successful and yields the image of a random divisor under the secret isogeny.

Example 2.7. *In the case of cryptosystems, we have $g = 2$ and the kernel is of the form $C^2_{\ell^e}$. In this case, Theorem 3.2 tells us that the probability that n faults are enough is $(1-\ell^{-n})\left(1-\ell^{1-n}\right)$. This tells us that on average, we need the following number of faults:*

$$\sum_{n=1}^{\infty} n\left(\left(1-\ell^{-n}\right)\left(1-\ell^{1-n}\right)-\left(1-\ell^{1-n}\right)\left(1-\ell^{2-n}\right)\right)=2+\frac{\ell+2}{\ell^2-1}$$

and if we used 2 faults, the probability that we generate the entire kernel is

$$\left(1-\ell^{-1}\right)\left(1-\ell^{-2}\right)=1-\ell^{-1}-\ell^{-2}+\ell^{-3}.$$

When $\ell = 2$, this value is $\frac{3}{8}$.

When only 2 faults are used, we can also give bounds on the average index of the subgroup, which gives us an estimate of how much brute-force is needed.

For an upper bound, note that quotienting by $\ell^e\mathbb{Z}_\ell \times \ell^e\mathbb{Z}_\ell$ reduces the size of the index, hence we obtain an upper bound with

$$\begin{aligned}
&\sum_{k=0}^{\infty} k\mathbb{P}(2,2,k)\\
=\ &\left(\ell^{-2};\ell\right)_2 \sum_{k=0}^{\infty} k l^{-2k}\binom{1+k}{k}_\ell\\
=\ &\left(1-l^{-2}\right)\left(1-l^{-1}\right)\frac{l^3(l+2)}{(l-1)(l^2-1)^2}\\
=\ &\frac{l+2}{l^2-1}
\end{aligned}$$

where the sum of the product is derived from the q binomial theorem by substituting $m = 2, t = l^{-2}$ into the following identity:

$$\begin{aligned}
\sum_{k=0}^{\infty} k t^k \binom{m+k-1}{k}_\ell &= t\frac{\partial}{\partial t}\sum_{k=0}^{\infty} t^k \binom{m+k-1}{k}_\ell\\
&= t\frac{\partial}{\partial t}\prod_{i=0}^{m-1}\frac{1}{1-q^i t}.
\end{aligned}$$

When $l = 2, 3$, the upper bound is $\frac{4}{3}$ and $\frac{5}{8}$ respectively.

A lower bound is given by the following expression:

$$\sum_{k=0}^{e-1} k\mathbb{P}(2,2,k) + e\left(1-\sum_{k=0}^{e-1}\mathbb{P}(2,2,k)\right).$$

As an example, when $e = 9$ and $l = 2$, the value is approximately 1.330.

Since there are at most k^4 isogenies to brute-force, we can also estimate the upper bound for the number of isogenies to brute-force with a similar technique:

$$\sum_{k=0}^{\infty} k^4\mathbb{P}(2,2,k) = \frac{l^7+16l^6+75l^5+176l^4+219l^3+176l^2+65l+16}{(l^2-1)^4}.$$

When $l = 2, 3$, the upper bound is $\frac{2990}{27} \approx 110.741$ and $\frac{6755}{512} \approx 13.193$ respectively.

Example 2.8. *In the case that the kernel has the form $C_{\ell^e} \oplus C_{\ell^{e-k}} \oplus C_{\ell^k}, k < e$, then the result from Theorem 3.2 tells us that the probability that n faults are enough is $\left(1-\ell^{-n}\right)\left(1-\ell^{1-n}\right)\left(1-\ell^{2-n}\right)$. This tells us that we need on average, we need the following number of faults:*

$$\sum_{n=1}^{\infty} n \begin{pmatrix} \left(1-\ell^{-n}\right)\left(1-\ell^{1-n}\right)\left(1-\ell^{2-n}\right) \\ -\left(1-\ell^{1-n}\right)\left(1-\ell^{2-n}\right)\left(1-\ell^{3-n}\right) \end{pmatrix} = 3 + \frac{\ell^3+3\ell^2+4\ell+3}{\ell^4+\ell^3-\ell-1}$$

and if we used 3 faults, the probability that we generate the entire kernel is

$$\left(1-\ell^{-1}\right)\left(1-\ell^{-2}\right)\left(1-\ell^{-3}\right) = 1-\ell^{-1}-\ell^{-2}+\ell^{-4}+\ell^{-5}-\ell^{-6}\,.$$

When $\ell = 2$, this value is $\frac{21}{64} \approx 0.33$.

If only 3 faults are used, we can give bounds on the average index of the subgroup, which gives us an estimate of how much brute-force is needed.

Similar to the previous case, the lower bound is given by

$$\begin{aligned} & \sum_{k=0}^{\infty} k\mathbb{P}(3,3,k) \\ = \; & \left(\ell^{-3};\ell\right)_3 \sum_{k=0}^{\infty} k l^{-3k} \binom{2+k}{k}_{\ell} \\ = \; & \left(1-l^{-3}\right)\left(1-l^{-2}\right)\left(1-l^{-1}\right)\frac{l^3(l+2)}{(l-1)(l^2-1)^2} \\ = \; & \frac{l^3+3l^2+4l+3}{(l^3-1)(l+1)}\,. \end{aligned}$$

When $l = 2, 3$, the upper bound is $\frac{31}{21} \approx 1.476$ and $\frac{69}{104} \approx 0.663$ respectively.

A lower bound is given by the following expression:

$$\sum_{k=0}^{\min(k,e-k)-1} k\mathbb{P}(3,3,k) + e\left(1 - \sum_{k=0}^{\min(k,e-k)-1} \mathbb{P}(3,3,k)\right)\,.$$

As an example, when $\min(k, e-k) = 9$ and $l = 2$, the value is approximately 1.472.

Since there are at most k^4 isogenies to brute-force, we can also estimate the upper bound for the number of isogenies to brute-force with a similar technique:

$$\sum_{k=0}^{\infty} k^4\mathbb{P}(3,3,k) = \frac{l^{15}+20l^{14}+\ldots}{\left(l^3-1\right)^4\left(l+1\right)^4}\,.$$

When $l = 2, 3$, the upper bound is approximately 128.239 and 14.244 respectively.

Example 2.9. *As a final example, suppose that the kernel is of the form $C_{\ell^e} \oplus C_{\ell^{e-k}} \oplus C_{\ell^k}, k < e$ as before but we want the probability that with n faults such that*

$$\left[\phi\left(A\left[\ell^e\right]\right) : \langle [h]P_1, \ldots, [h]P_n\rangle\right] = \ell^{\delta}$$

with $\delta < \min(k, e-k)$. This is again given by Theorem 3.2 which gives us

$$\mathbb{P}(3,n,\delta) = \ell^{-n\delta}\left(\ell^{-n};\ell\right)_3 \binom{2+\delta}{\delta}_{\ell} = \ell^{-n\delta}\left(\prod_{i=0}^{2} 1-\ell^{i-n}\right)\left(\prod_{i=1}^{\delta}\frac{1-\ell^{3+k-i}}{1-\ell^i}\right) \approx \ell^{-\delta(n-2)}\,.$$

2.2 Pseudocode and software simulation of the attack

Suppose we have abelian varieties A, B of dimension 2 and an isogeny $\phi : A \to B$ and the images of random points $P_i \in A(\mathbb{F}_q)$ under ϕ. Further suppose that $\ker \phi \subset A[\ell^e] \subset A(\mathbb{F}_q)$ and suppose that we have $|\ker \phi| = \ell^k$. We provide the pseudocode to obtain the isogeny ϕ:

Algorithm 1: Recovery of isogeny after fault injection

Data: $\phi(P_i)$
Result: $\phi : A \to B$
$h \leftarrow \ell^{-ge} |A(\mathbb{F}_q)|$;
$P_i' \leftarrow [h]\phi(P_i)$; /* Now we have $P_i' \in \phi(A[\ell^e])$*/
Determining how much brute-force is needed;
Set Q_i as generators of $B[\ell^e]$;
Compute $M_{i,j} \in \mathbb{Z}_\ell$ with $P_i' = \sum_j M_{i,j} Q_j$;
$UDV \leftarrow M$; /* Smith normal form over $\mathbb{Z}_\ell$*/
$a_i \leftarrow D_{i,i}$;
$m \leftarrow |\{a_i | v_\ell(a_i) < e\}|$;
$\ell^\epsilon \leftarrow \ell^k \ell^{-me} \prod_{i=1}^{m'} p^{v_\ell(a_i)}$;
Add more row vectors into D with brute-force to get $B/\langle (DV)_1, (DV)_2, \ldots, (DV)_n \rangle \cong A$; /* We only need to brute-force vectors in $\phi(A[\ell^e])/\langle P_1', P_2', \ldots, P_n' \rangle$ See Lemma 2.4 and Example 2.9 for the details*/
$G \leftarrow \langle P_1', P_2', \ldots, P_n' \rangle = \phi(A[\ell^e]) := \ker \psi$;
Computing ϕ;
for $i \leftarrow 1$ **to** e **do**
    $A_i \leftarrow B/([\ell^{e-i}]G)$;
    $\psi_i : A_{i-1} \to A_i$;
    Brute-force $\phi_i : A_i \to A_{i-1}$ to get $[\ell^i] = \psi_i[\ell^{i-1}]\phi_i$;
end
$\phi \leftarrow \phi_1 \ldots \phi_{e-1}\phi_e$;
return ϕ;

An implementation that simulates fault injection and secret key recovery can be found in `https://github.com/yanboti/Genus2FaultAttack`. Note that the implementation simplifies pseudocode by omitting the Smith normal form computation and does not require brute-forcing.

3 Analysis of attack

In the attack above, we are able to generate a subgroup of $\phi(A[\ell^e])$. If the subgroup has a small enough index, then brute-forcing the remaining parts of the group becomes feasible. In this section, we compute the probability that n random points generate a subgroup of a fixed index in $\phi(A[\ell^e])$. In the case the index of the subgroup has index 1, then no brute-force is needed. Since $\phi(A[\ell^e])$ is a quotient of $A[\ell^e] \cong C_{\ell^e}^{2g}$, it can be written in the form $\bigoplus_{i=1}^m C_{\ell^{q_i}}$.

To proof a statement about the index, it is most natural to consider the Smith normal form of some matrix since the index of the subgroup generated can be read directly, see Remark 2.6 for more information. Although $A[\ell^e]$ could be viewed as a module over $\mathbb{Z}$ or $\mathbb{Z}/\ell^e\mathbb{Z}$, integers coprime to ℓ are not invertible in $\mathbb{Z}$ and $\mathbb{Z}/\ell^e\mathbb{Z}$ is not a domain, which means the Smith normal form is not well-defined. Since $\ell^{>e} A[\ell^e] = 0$, we can view $A[\ell^e]$ as a $\mathbb{Z}_\ell$-module where $\mathbb{Z}_\ell$ is the ℓ-adic integers and this resolves both problems earlier since integers coprime to ℓ are invertible and $\mathbb{Z}_\ell$ is a domain. Hence we can first study random points in $\mathbb{Z}_\ell^m$ first, then take a quotient to get the result we are interested in.

For convenience, we introduce the notations for the ℓ-deformed Pochhammer symbols and binomial coefficients

Definition 3.1.

$$(a;\ell)_n = \prod_{k=0}^{n-1} 1 - a\ell^k$$

$$\binom{m}{n}_\ell = \prod_{k=0}^{n-1} \frac{1-\ell^{m-k}}{1-\ell^{k+1}}$$

We have the following theorem for the probability that n random points in $\mathbb{Z}_\ell^m$ generate a submodule of index k.

Theorem 3.2. *Let P_i be n random elements in $\mathbb{Z}_\ell^m$ and assume $m \leq n$. Let $M = \mathbb{Z}_\ell^m / \langle P_1, \ldots, P_n \rangle$ where $\langle P_1, \ldots, P_n \rangle$ is the $\mathbb{Z}_\ell$-module generated by ℓ. Denote the probability that $|M| = \ell^k$ as $\mathbb{P}(m,n,k)$. Then*

$$\mathbb{P}(m,n,k) = \ell^{-nk} \left(\ell^{-n};\ell\right)_m \binom{m+k-1}{k}_\ell .$$

Before proving the theorem, we provide the following corollary in the case that we are interested in.

Corollary 3.2.1. *The probability that n random elements in $\bigoplus_{i=1}^m \frac{\mathbb{Z}_\ell}{\ell^{q_i}\mathbb{Z}_\ell}$ generates a subgroup of index ℓ^k where $k < \min_i q_i$ is $\mathbb{P}(m,n,k)$. In the case that $k = \min_i q_i$, then $\mathbb{P}(m,n,k)$ gives a lower bound.*

Proof. For any submodule $M \subset \mathbb{Z}_\ell^m$ of finite index ℓ^k, it can be written as $M = \bigoplus_{i=1}^m \ell^{m_i}\mathbb{Z}_\ell$ with $\sum_i m_i = k$. This submodule projected to the quotient module $\bigoplus_{i=1}^m \frac{\mathbb{Z}_\ell}{\ell^{q_i}\mathbb{Z}_\ell}$ would then have the form

$$\bigoplus_{i=1}^m \frac{\ell^{m_i}\mathbb{Z}_\ell + \ell^{q_i}\mathbb{Z}_\ell}{\ell^{q_i}\mathbb{Z}_\ell}$$

which has index $\ell^{\sum_i \min(m_i,q_i)}$. This sum is precisely $\ell^{\sum_i m_i} = \ell^k$ if $m_i < q_i$ for all i, which is true if $k \leq q_i$. Otherwise, it is at least $\ell^{\min_i q_i}$. This shows that the probability is equal for $k < \min_i q_i$ and $\mathbb{P}(m,n,k)$ provides a lower bound when $k = \min_i q_i$ □

In the theorem above, $\mathbb{P}(m,n,0)$ gives the probability that we generate the full kernel and the k in $\mathbb{P}(m,n,k)$ is a measure of how much brute-force is needed. To understand the terms, we provide the following approximation:

Remark 3.3. *We have that*

$$\begin{aligned}
\mathbb{P}(m,n,k) &= \ell^{-nk} \left(\ell^{-n};\ell\right)_m \binom{m+k-1}{k}_\ell \\
&= \ell^{-nk} \left(\prod_{i=0}^{m-1} 1 - \ell^{i-n} \right) \left(\prod_{i=1}^{k} \frac{1-\ell^{m+k-i}}{1-\ell^i} \right) \\
&= \ell^{-nk} \left(1 - \ell^{-(n+1-m)} + O\left(\ell^{-(n+2-m)}\right) \right) \left(\ell^{k(m-1)} \underbrace{+\ell^{k(m-1)-1}}_{\text{if } k \neq 0} + O\left(\ell^{k(m-1)-2}\right) \right) \\
&= \ell^{-k(n+1-m)} + O\left(\ell^{-k(n+1-m)-1}\right) .
\end{aligned}$$

Thus, in the case of $k = 0$ we have

$$\mathbb{P}(m,n,0) = \left(\ell^{-n};\ell\right)_m = \prod_{i=0}^{m-1} 1 - \ell^{i-n} \geq \left(1 - \ell^{-(n-m+1)}\right)^m .$$

Before proving the theorem, we use the following lemmas to prove the case for $k = 0$, before generalizing it by induction.

Lemma 3.4. *Let $\mathcal{P}$ be a $m \times n$ matrix with $\mathcal{P}_{j,i} = (P_i)_j$. Suppose its invariant factors are $a_1, a_2, \ldots, a_m$, then $|M| = \prod_{i=1}^{m} a_i$.*

Remark 3.5. *This allows us to consider the rows of matrix $\mathcal{P}$ instead of the columns P_i, giving us m vectors in $\mathbb{Z}_\ell^n$, making counting a lot easier.*

Lemma 3.6.

$$\mathbb{P}(m, n, 0) = \prod_{i=0}^{m-1} 1 - \ell^{i-n} = \left(\ell^{-n}; \ell\right)_m .$$

Proof. We can work over $\mathbb{F}_\ell$ by reduction mod ℓ. Furthermore, by Remark 3.5, the problem reduces to finding the number of ordered linearly independent m-tuples $(\mathcal{P}_i)_{i=1}^m$ over $\mathbb{F}_\ell^n$. We require

$$\mathcal{P}_{i+1} \in \mathbb{F}_\ell^n - \langle \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_i \rangle$$

and since

$$|\mathbb{F}_\ell^n - \langle \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_i \rangle| = \ell^n - |\langle \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_i \rangle| = \ell^n - \ell^i$$

we get

$$\mathbb{P}(m, n, 0) = \prod_{i=0}^{m-1} 1 - \ell^{i-n} = \left(\ell^{-n}; \ell\right)_m .$$

□

Now we can prove Theorem 3.2.

Proof. We proceed by induction on a recurrence relation. Suppose that $\mathcal{P}_1 \in \ell\mathbb{Z}_\ell^n$, then by dividing ℓ from each component, we obtain a new matrix with one of the invariant factors reduced by a factor of ℓ. Otherwise, by swapping the columns appropriately, we can get a unit at $\mathcal{P}_{1,1}$, and by using elementary row operations, we can get $\mathcal{P}_{i,1} = \delta_{i,1}$. This reduces to the case of the smaller matrix, hence we have the recursion relation

$$\mathbb{P}(m, n, k) = \frac{1}{\ell^n}\mathbb{P}(m, n, k-1) + \frac{\ell^n - 1}{\ell^n}\mathbb{P}(m-1, n-1, k)$$

and the boundary conditions $\mathbb{P}(m, n, 0) = (\ell^{-n}; \ell)_m$ and evidently $\mathbb{P}(1, n, k) = \ell^{-nk} - \ell^{-n(k+1)}$. The second boundary condition and the recursion relation tell us we can extend the function to $\mathbb{P}(0, n, k) = 0$.

From this, we get the following recursion relation

$$\mathbb{P}\left(m, n, k\right) = \sum_{i=0}^{m-1} \ell^{-n+i} \left(\ell^{-n}; \ell\right)_i \mathbb{P}\left(m-i, n-i, k-1\right) .$$

With this, we can prove the theorem by induction.

By the previous lemma, we have $\ell^0 \left(\ell^{-n}; \ell\right)_m \binom{m-1}{0}_\ell = \left(\ell^{-n}; \ell\right)_m = \mathbb{P}(m, n, 0)$ which proves the base case of $k = 0$. For the inductive step, we have

$$\begin{aligned}
&\mathbb{P}\left(m, n, k\right) \\
&= \sum_{i=0}^{m-1} \ell^{-n+i} \cdot \left(\ell^{-n}; \ell\right)_i \cdot \mathbb{P}\left(m-i, n-i, k-1\right) \\
&= \sum_{i=0}^{m-1} \left(\ell^{-n+i} \cdot \left(\ell^{-n}; \ell\right)_i \cdot \ell^{-(n-i)(k-1)} \cdot \left(\ell^{-n+i}; \ell\right)_{m-i} \cdot \binom{m-i+k-2}{k-1}_\ell \right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\ell^{-n};\ell\right)_m \cdot \ell^{-nk} \cdot \sum_{i=0}^{m-1} \ell^{ik} \binom{m+k-i-2}{k-1}_\ell \\
&= \left(\ell^{-n};\ell\right)_m \cdot \ell^{-nk} \cdot \sum_{i=0}^{m-1} \left(\ell^{ik} \cdot \binom{m+k-i-1}{k}_\ell - \ell^{(i+1)k} \binom{m+k-i-2}{k}_\ell\right) \\
&= \left(\ell^{-n};\ell\right)_m \cdot \ell^{-nk} \cdot \left(\binom{m+k-1}{k}_\ell - \ell^{mk} \binom{k-2}{k}_\ell\right) \\
&= \left(\ell^{-n};\ell\right)_m \cdot \ell^{-nk} \cdot \binom{m+k-1}{k}_\ell .
\end{aligned}
$$

□

4 Fault attack via loop-abort

The premise of the loop-abort attack is simple: terminate the isogeny-chain computation prematurely so that secret information regarding the full isogeny can be extracted. At its simplest description, the loop-abort attack is able to interrupt the computation of Alice's secret isogeny and forces the device to output intermediate values that can be used by an adversary to recover Alice's secret isogeny.

4.1 The attack

Consider the isogeny computation of a $(2^n, 2^{n-k}, 2^k)$-isogeny ϕ. Since isogeny computations have complexities in the order of the degree of the isogeny, the complexity of ϕ is $O(2^{2n})$. One can employ a computational trick to reduce the complexity to $O(4n)$ by factoring the isogeny into n consecutive $(2,2)$-isogenies. The isogeny computation would then iterate through all n $(2,2)$-isogeny computations before outputting the result. In fact, this sort of computation is done in all implementations of isogeny-based cryptography that we know of.

The isogeny computation in G2SIDH-like schemes is partitioned into small degree isogenies for computation. Alice's isogeny computation can be represented as

$$
J_0 \xrightarrow{\phi_0} J_1 \xrightarrow{\phi_1} J_2 \xrightarrow{\phi_2} \ldots \xrightarrow{\phi_{n-1}} J_n \,,
$$

where the ϕ_i's are computed sequentially in a loop.

The adversary's goal is to recover all the ϕ_i's, which is equivalent to knowing the secret isogeny. If the adversary can interrupt the iterative computation of the isogeny to output J_i for $i = 1, \ldots, n-1$, they will be able to recover all the ϕ_i's. To see this, observe that the adversary can recover ϕ_1 given J_1 by checking all the (2,2)-isogenous varieties from J_0 and finding a match. Knowing the isogeny ϕ_{i-1}, the adversary can work out J_i (or, they could have stored this from the previous iteration), and in the next iteration of the attack, when given J_{i+1}, they can identify ϕ_i. Performing this attack iteratively, the adversary will be able to recover all ϕ_i, and hence the secret isogeny.

4.2 Trade-off between number of faults and computations

Recall now that there are generally 15 $(2,2)$-isogenies from a given PPAS. Hence each iteration of the isogeny would increase the search complexity fourteen-fold (after excluding the dual isogeny). The aim of the loop-abort attack is to interrupt the isogeny computation midway through the iteration. This allows the adversary to reduce the search space of the isogeny problem to something manageable. Indeed, with precise control of such an attack, one only needs to perform $O(14n)$ computations to recover the secret isogeny in its entirety at a cost of n precise fault injections. Attackers can also elect to increase the computational complexity in order to reduce the number successful of fault injections; in general, m successful perturbations can result in $O(14^{n/m}m)$ complexity in the best case.

5 Applicability to genus-2 isogeny-based cryptography

This section describes the impact of the fault attack on existing genus-2 isogeny-based cryptosystems. In particular, we will state how the fault attack may be used against G2SIDH of Flynn and Ti [FT19], the generalisation of the identification scheme based on SIDH [JF11].

5.1 Fault attack models

The crux of a fault attack is to introduce faults in the implementation through external means. Faults can be brought about via an electromagnetic pulse, a clock glitch, or a voltage glitch. The impact of such perturbations on a target device and to the adversary's ability to recover secret material can differ greatly depending on the following characteristics of fault attacks (see [BBB+22]).

- *Precision.* The precision of the fault describes the ability of an adversary to select bits in certain variables to be perturbed at a particular point in time. The more precise the attack, the more unrealistic it will be.
- *Deterministic.* A deterministic attack model hinges on the ability of an adversary to set the value of the parameter under attack. A random fault model can only randomly choose a value of the parameter under attack and is more practical than a deterministic one.
- *Repeatability.* This indicates the adversary's ability to repeat identical faults in either the temporal or spatial dimensions through different traces. Requiring repeatability in an attack would make the attack more difficult to pull off.

Point perturbation attack

The point perturbation attack can be classified as an imprecise and random attack that does not require repeatability. The attack does not need to be precise because there are multiple locations in the implementation where auxiliary points are called for computation. This is because the fault only needs to be directed on any one of the auxiliary points in the protocol, and random faults on the point would allow an adversary to recover the secret. Furthermore, according to [TFMP21, §4.2], the margin of the location and timing of the fault is relatively large. This is particularly so for genus-2 cryptosystems as the target (the x^1- and x^0-coordinates) have a combined footprint of $4 \log_2 p$-bits, this allows for less precision in the fault injection.

Loop-abort attack

The loop-abort attack requires that a fault is inserted into a loop counter. Targeting a loop counter would require precision in time and space of where the fault should be injected. However, in most implementations, two observations can make the loop-abort attack more feasible:

(a) Most possible values of the loop-counter would cause the loop to abort, so there is no need to target a precise bit as randomisations of the value would be sufficient for the loop to be aborted.

(b) To cause the i-th loop to be aborted, the perturbation on the loop counter can be made at any time during the i-th loop. This increases the temporal margin of a fault attack.

5.2 Attack models on protocols

In the following, we will examine how the two different fault attacks can be used against different protocols. In particular, we focus on the applicability of the two fault attacks on the protocols.

Ideally, the fault attack would be able to recover the secret isogeny without the knowledge of either party, and the attack should be successful with a single trace. However, this is not possible in both fault attacks, because correct images of auxiliary points are required by SIDH and the genus-2 protocols, and premature

termination of the isogeny computation would result in the wrong public keys. By introducing faults, these protocols are unable to be completed as specified and errors may be raised. The upshot is that fault injection attacks can be detected.

5.2.1 G2SIDH

Recall that the G2SIDH protocol is a key-exchange protocol between two parties seeking to establish a secret key. This protocol has been described in §1.3. To break the protocol, an adversary has to learn the secret key of either one of the parties.

Point perturbation attack. Suppose an adversary is trying to learn Alice's static secret isogeny and has the ability to cause a fault in Alice's computation. After introducing a fault in Q_i just prior to Alice's computation of $\phi_A(Q_i)$ for $i = 1, 2, 3, 4$, Alice would then proceed to publish the public key tuple

$$(J_A, \phi_A(X_1), \phi_A(X_2), \phi_A(X_3)), \phi_A(X_4)).$$

The adversary will then be able to recover ϕ_A using the point perturbation attack.

But it should be noted that this attack has to be targeted at the key generation phase of G2SIDH since the derivation of the shared key step of the procedure does not require the computation of auxiliary points.

As was noted in [TFMP21], a normal implementation of G2SIDH should not require that Alice recomputes her static public key in the key generation step. The presence of the point perturbation attack makes such an implementation mistake even more devastating. Also, the fault attack model is more realistic in a multiparty scenario where a central server has a different static key, that will be re-computed during each session, for each user.

A multiparty scenario can be one of the two cases:

(a) Alice trying to communicate with Bob and Charlie using different static public keys for each.

(b) Alice, Bob, and Charlie trying to establish a shared secret using the multiparty key exchange scheme proposed by Azarderakhsh, Jalali, Jao, and Soukharev [AJJS19].

In the former case, Alice uses different static keys to communicate with Bob and Charlie. In this setting, it is not unreasonable to assume that Alice would store only the static secrets with Bob and Charlie to save on storage[1]. This would require that Alice recomputes her public key (using the secret isogeny) each time Bob or Charlie initiates an interaction.

In the latter case, each time the multiparty protocol is invoked, Alice would be compelled to compute using her static secret in the fourth pass of the protocol. Refer to Step 4 of Section 4.3 of [AJJS19]. Notice that Alice does not need to recompute the first step of the protocol since that is always fixed.

Loop-abort attack. In this attack, the adversary is able to recover the static secret isogeny of G2SIDH by iteratively computing the key generation or shared key derivation phase of the protocol. Having learnt the isogeny ϕ_i, the adversary injects a fault in the $(i + 1)$st round to recover ϕ_{i+1}. Note that at most a total of e_A traces are needed to recover the entirety of the secret key. The adversary can reduce the number of traces needed to recover the secret isogeny at the expense of more computation by following the trade-offs suggested in §4.2.

5.2.2 Identification protocol

Although an identification protocol has not formally been defined for genus-2 isogenies, it can be derived from the ingredients used to construct G2SIDH. We will not be detailing the exact construction and will refer readers to [JF11] to note the similarities between the key exchange protocol and the identification protocol. In such an identification scheme, the adversary's task is to recover the prover's long-term secret isogeny.

[1]Since the storage requirement can increase rapidly with the increase in the number of parties she needs to communicate with.

For the point perturbation attack, the fault has to be injected into the key generation phase to recover the static long-term secret. However, attacking this protocol with the point perturbation attack faces the same difficulties as in the G2SIDH protocol. Indeed, the static secret isogeny will not be used to compute images of auxiliary points outside of the key generation. However, our observation regarding the multiparty scenarios in G2SIDH carries over to the identification scheme.

The loop-abort attack is suitable in this scenario. This is because identification protocols require multiple passes to achieve the desired level of security. Loop-abort attacks can then target the computations that use the secret isogeny to recover it.

5.3 Feasibility of attack models

The feasibility of the fault attack on various SIDH-like protocols has been discussed in by Ti [Ti17], Gélin and Wesolowski, and Tasso, De Feo, El Mrabet and Pontié [TFMP21].

The point perturbation attacks will cause a failure in genus one key exchange protocols since the auxiliary points that are needed to complete the protocol will be altered. We have also shown that with 2-4 traces (as given by the examples in §2) of the point perturbation attack are required to fully recover the secret key with high probability. The point perturbation also benefits from allowing for more margin in the fault injection in both temporal and spatial dimensions. The disadvantage of the point perturbation attack is that it can only be launched against implementations in the key generation phase. As mentioned, it can also work on implementations that recompute the key generation phase. However, this is not guaranteed in every implementation.

In contrast, the loop-abort attack can be launched against all the protocols that we have looked at in Section 5.2. This is because the loop-abort attack targets the sequential computation of secret isogenies which forms the basis of all of the cryptosystems looked at in this paper. The downside of this approach is that inducing a fault in the loop counter is comparatively more difficult than injecting a fault in a point. Furthermore, significantly more traces are needed to recover the entire secret key (up to e_A traces).

The experimental results of [TFMP21] showed that fault attacks of [Ti17] are "exploitable in practice though electromagnetic injection on a SoC." We believe that the genus-2 version of this fault attack will retain this property of practicality. Experimental results on loop-abort attacks by Page and Vercauteren [PV06], Espitau, Fouque, Gérard and Ticouchi [EFGT18], and Blömer, da Silva, Günther, Krämer and Seifert [BdSG$^+$14], have demonstrated the feasibility of loop-abort attacks.

5.4 Countermeasures

The countermeasure to thwart the point perturbation attack is to implement order checking before the publication of the auxiliary points. The purpose of such a check is to ensure that faulted points are not mapped through the secret isogeny which would leak information about the secret. This is the same as with the SIDH case and can be implemented with minimal overheads. The point order checking will be able to detect perturbations in the points since perturbed points are random points and will have random order in the group. Performing a point order check would then identify these points, thus halting outputs from being produced, thus thwarting the attack.

Loop-abort attacks can be countered by introducing (one or more) parallel counters that check against the primary loop counter after each iteration of the isogeny computation. This simple countermeasure will be able to detect faults in the loop counter and halt the computation if such a perturbation is observed, thus thwarting the loop-abort attack.

The countermeasures against these attacks are simple and cheap to implement.

References

[ACDMRH22] Gora Adj, Jesús-Javier Chi-Domínguez, Víctor Mateu, and Francisco Rodríguez-Henríquez, *Faulty isogenies: a new kind of leakage*, Cryptology ePrint Archive, Report 2022/153, 2022,

https://ia.cr/2022/153.

[AJJS19] Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev, *Practical supersingular isogeny group key agreement*, IACR Cryptol. ePrint Arch. (2019), 330.

[BBB+22] Anubhab Baksi, Shivam Bhasin, Jakub Breier, Dirmanto Jap, and Dhiman Saha, *A survey on fault attacks on symmetric key cryptosystems*, ACM Comput. Surv. (2022), Just Accepted.

[BdSG+14] Johannes Blömer, Ricardo Gomes da Silva, Peter Günther, Juliane Krämer, and Jean-Pierre Seifert, *A practical second-order fault attack against a real-world pairing implementation*, 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014 (Assia Tria and Dooho Choi, eds.), IEEE Computer Society, 2014, pp. 123–136.

[CKM+20] Fabio Campos, Matthias J. Kannwischer, Michael Meyer, Hiroshi Onuki, and Marc Stöttinger, *Trouble at the CSIDH: protecting CSIDH with dummy-operations against fault injection attacks*, 17th Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2020, Milan, Italy, September 13, 2020, IEEE, 2020, pp. 57–65.

[CKM21] Fabio Campos, Juliane Krämer, and Marcel Müller, *Safe-error attacks on SIKE and CSIDH*, Security, Privacy, and Applied Cryptography Engineering - 11th International Conference, SPACE 2021, Kolkata, India, December 10-13, 2021, Proceedings (Lejla Batina, Stjepan Picek, and Mainack Mondal, eds.), Lecture Notes in Computer Science, vol. 13162, Springer, 2021, pp. 104–125.

[CLG09] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren, *Cryptographic hash functions from expander graphs*, J. Cryptol. **22** (2009), no. 1, 93–113.

[CNP02] Gabriel Cardona, Enric Nart, and Jordi Pujolàs, *Curves of genus two over fields of even characteristic*, Mathematische Zeitschrift **250** (2002), 177–201.

[Cou06] Jean Marc Couveignes, *Hard homogeneous spaces*, IACR Cryptol. ePrint Arch. (2006), 291.

[CQ05] Gabriel Cardona and Jordi Quer, *Field of moduli and field of definition for curves of genus 2*, Lecture Notes Ser. Comput. **13** (2005), 71–83.

[CS20] Craig Costello and Benjamin Smith, *The supersingular isogeny problem in genus 2 and beyond*, Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings (Jintai Ding and Jean-Pierre Tillich, eds.), Lecture Notes in Computer Science, vol. 12100, Springer, 2020, pp. 151–168.

[EFGT18] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi, *Loop-abort faults on lattice-based signature schemes and key exchange protocols*, IEEE Trans. Computers **67** (2018), no. 11, 1535–1549.

[FT19] E. Victor Flynn and Yan Bo Ti, *Genus two isogeny cryptography*, Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers (Jintai Ding and Rainer Steinwandt, eds.), Lecture Notes in Computer Science, vol. 11505, Springer, 2019, pp. 286–306.

[Gal12] Steven D Galbraith, *Mathematics of public key cryptography*, Cambridge University Press, 2012.

[GGR05] Josep Gonzalez, Jordi Guardiaand, and Victor Rotger, *Abelian surfaces of GL_2-type as jacobians of curves*, Acta Arithmetica 116 (2005), 263–287.

[GW17] Alexandre Gélin and Benjamin Wesolowski, *Loop-abort faults on supersingular isogeny cryptosystems*, Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings (Tanja Lange and Tsuyoshi Takagi, eds.), Lecture Notes in Computer Science, vol. 10346, Springer, 2017, pp. 93–106.

[Igu60] Jun-Ichi Igusa, *Arithmetic variety of moduli for genus two*, Annals of Mathematics **72** (1960), no. 3, 612–649.

[JF11] David Jao and Luca De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings (Bo-Yin Yang, ed.), Lecture Notes in Computer Science, vol. 7071, Springer, 2011, pp. 19–34.

[KAJ17] Brian Koziel, Reza Azarderakhsh, and David Jao, *Side-channel attacks on quantum-resistant supersingular isogeny diffie-hellman*, Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers (Carlisle Adams and Jan Camenisch, eds.), Lecture Notes in Computer Science, vol. 10719, Springer, 2017, pp. 64–81.

[KPHS18] Philipp Koppermann, Eduard Pop, Johann Heyszl, and Georg Sigl, *18 seconds to key exchange: Limitations of supersingular isogeny diffie-hellman on embedded devices*, IACR Cryptol. ePrint Arch. (2018), 932.

[KTW21] Sabrina Kunzweiler, Yan Bo Ti, and Charlotte Weitkämper, *Secret keys in genus-2 SIDH*, IACR Cryptol. ePrint Arch. (2021), 990.

[MRM74] David Mumford, Chidambaran Padmanabhan Ramanujam, and Yuri Ivanovich Manin, *Abelian varieties*, vol. 3, Oxford university press Oxford, 1974.

[PV06] Dan Page and Frederik Vercauteren, *A fault attack on pairing-based cryptography*, IEEE Trans. Computers **55** (2006), no. 9, 1075–1080.

[RS06] Alexander Rostovtsev and Anton Stolbunov, *Public-key cryptosystem based on isogenies*, IACR Cryptol. ePrint Arch. (2006), 145.

[Tak17] Katsuyuki Takashima, *Efficient algorithms for isogeny sequences and their cryptographic applications*, Mathematical Modelling for Next-Generation Cryptography: CREST Crypto-Math Project (Tsuyoshi Takagi, Masato Wakayama, Keisuke Tanaka, Noboru Kunihiro, Kazufumi Kimoto, and Dung Hoang Duong, eds.), Mathematics for Industry, Springer Singapore, 2017, pp. 97–114.

[TFMP21] Élise Tasso, Luca De Feo, Nadia El Mrabet, and Simon Pontié, *Resistance of isogeny-based cryptographic implementations to a fault attack*, Constructive Side-Channel Analysis and Secure Design - 12th International Workshop, COSADE 2021, Lugano, Switzerland, October 25-27, 2021, Proceedings (Shivam Bhasin and Fabrizio De Santis, eds.), Lecture Notes in Computer Science, vol. 12910, Springer, 2021, pp. 255–276.

[Ti17] Yan Bo Ti, *Fault attack on supersingular isogeny cryptosystems*, Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings (Tanja Lange and Tsuyoshi Takagi, eds.), Lecture Notes in Computer Science, vol. 10346, Springer, 2017, pp. 107–122.

[UXT+22] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma, *Curse of re-encryption: A generic power/em analysis on post-quantum kems*, IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022** (2022), no. 1, 296–322.

[XIU+21] Keita Xagawa, Akira Ito, Rei Ueno, Junko Takahashi, and Naofumi Homma, *Fault-injection attacks against nist's post-quantum cryptography round 3 KEM candidates*, Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II (Mehdi Tibouchi and Huaxiong Wang, eds.), Lecture Notes in Computer Science, vol. 13091, Springer, 2021, pp. 33–61.

[ZYD+20] Fan Zhang, Bolin Yang, Xiaofei Dong, Sylvain Guilley, Zhe Liu, Wei He, Fangguo Zhang, and Kui Ren, *Side-channel analysis and countermeasure design on arm-based quantum-resistant SIKE*, IEEE Trans. Computers **69** (2020), no. 11, 1681–1693.