Generalising Fault Attacks to Genus Two Isogeny Cryptosystems

Ariana Goh[1], Chu-Wee Lim[1], and Yan Bo Ti[1]

DSO National Laboratories, Singapore
ari.gzh@gmail.com, lchuwee@dso.org.sg, yanbo.ti@gmail.com

Abstract. In this paper we generalise Ti's fault attack and the loop abort fault attacks on supersingular isogeny cryptosystems (genus one) to genus two. Genus two isogeny based cryptosystems are generalisations of its genus one counterpart, as such, attacks on the the latter are believed to generalise to the former.
Fault attacks on supersingular elliptic curve isogeny cryptography has been shown to be practical. We show in this paper that fault attacks continue to be practical in genus two, albeit with a few additional traces required.

Isogeny-based cryptography was proposed in 1997 by Couveignes in an unpublished manuscript [Cou06] on hard homogeneous spaces.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 third-round alternative candidate in NIST's post-quantum standardisation process. 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 which generalises SIDH to genus two 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 two in 2018, and this was improved by Castryck, Decru, and Smith in 2019. Cryptanalysis of higher genera cryptosystems have 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 are 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 discuss the implications of this attack on current genus two cryptosystems in Section 4. Lastly, we will sketch the loop-abort attack on G2SIDH in Section 5.

1 Preliminaries/Background

This section serves as a concise introduction to the central characters of genus two 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:

1. the Jacobian $J(C)$ of a smooth projective curve C of genus 2, or
2. 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 11 ([Gal12]) *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, Chap. 10] 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 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 12 *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*

1. *the Weil pairing $e_m : A[m] \times A[m] \to \mu_m$ restricts trivially to G, and*
2. *G is not properly contained in any subgroup of $A[m]$ satisfying (1).*

The significance of such subgroups is given by the following.

Proposition 13 *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 $l \neq p$ be a small prime, say $l \in \{2, 3\}$. In the case where $m = l^n$, we have the following.

Lemma 14 *The maximal l^n-isotropic subgroups of $A[l^n]$ are isomorphic to either*

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

for some $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$. In particular, each $G \cong C_l \times C_l \subseteq A[l]$ 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 $l \neq p$, say $l \in \{2, 3\}$, and consider the graph $\mathcal{G}_{p,l}$ 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 $\mathcal{G}_{p,l}$ is not collision resistant due to the prevalence of "diamonds". In the same paper, the authors describe G2SIDH, a genus two variant of the SIDH key-exchange protocol based on $\mathcal{G}_{p,l}$ for $l = 2, 3$. 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 two 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 $\mathcal{G}_{p,l}$, select random points $P_1, P_2, P_3 \in A[l^n]$ such that $G = \langle P_1, P_2, P_3 \rangle$ is a maximal l^n-isotropic subgroup of $A[l^n]$. To achieve this, one can proceed as in [KTW21] with a symplectic basis of $A[l^n]$ which 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 two 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 two 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

We summarise the fault attack on SIDH in [Ti17].

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 point 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, we obtain the image of a valid random divisor under the secret isogeny, and hence recovery the secret isogeny by Lemma 23.

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, x^1 coefficients.

Half the time after faulting, the roots of u lies 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 lies 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 $P + P^{\text{Frob}_{\mathbb{F}_{q^2}/\mathbb{F}_q}} - 2\infty$, 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 result is well-known in the literature and is stated here for completeness. We will use this proposition for the other results to come.

Proposition 21 ([MRM74, Prop II.6.3]) *Let A be an abelian variety of dimension g, we have*

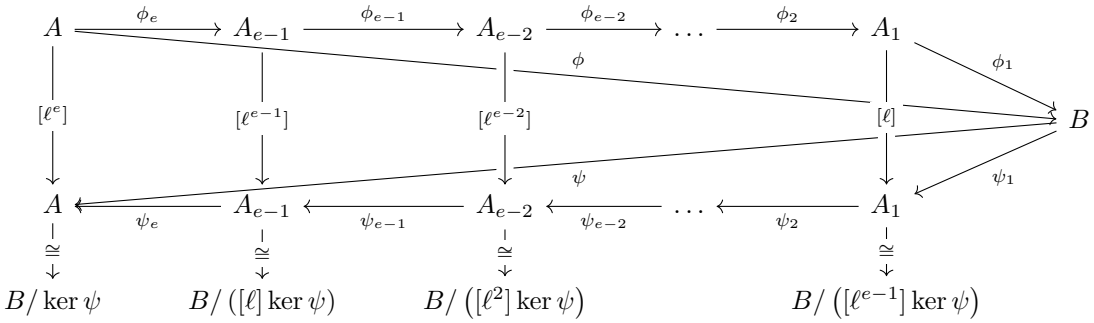
$$A[n] \cong \begin{cases} C_n^{2g} & p \nmid n \\ C_n^i & n = p^k \end{cases} \quad 0 \leq i \leq 2g$$

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 22 *Suppose we have a separable isogeny $\phi : A \to B$ of abelian varieties over $\overline{\mathbb{F}}_p$ with $\ker \phi \subset A[n]$ and we are given $\phi(A[n])$. We can recover ϕ efficiently whenever n has small prime factors.*

Proof. By [MRM74, Thm II.7.4], we know that there exists a separable isogeny $\psi : B \to B/(\phi(A[n])) \cong A$ such that $\ker \psi = \phi(A[n])$ and $\psi \circ \phi = [n]$.

In the case that $n = \ell^e$ for a small prime ℓ, it is computationally feasible to recover ϕ given $\ker \psi$. Consider the following commutative diagram:



As there are only a small number of possible separable isogenies f where $\ker f \subset \ker[\ell]$, given by the number of subgroups of $\ker[\ell]$ which is quite small by Proposition 21. Hence each ϕ_i is easy to compute and ϕ is easy to compute

Now in the case that $n = \prod_i \ell_i^{e_i}$, repeat the above procedure but for each prime and compose the isogenies.

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 23 *Suppose furthermore that $A[\ell^e] \subset A(\mathbb{F}_{p^r})$ and $\phi(A[\ell^e])$ has m generators. Given the images of at least 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 we can find $\phi(A[\ell^e])$.*

Proof. Let $n = \ell^{-ge} \left|A\left(\mathbb{F}_{p^r}\right)\right|$. n is a integer as $A\left[\ell^e\right]$ is a subgroup of $A\left(\mathbb{F}_{p^r}\right)$ $\left|A\left[\ell^e\right]\right| = \ell^{ge}$.

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

We can know precisely when we have sufficiently many points with the following procedure. Find some minimal set of generators of $B\left[\ell^e\right]$, say Q_j. Then we know that $\sum_{i=j}^{2g} \mathbb{Z}Q_j = C_{\ell^e}^{2g}$. Express every point $[n]P_i = \sum_{j=1}^{2g} M_{i,j}Q_j$ and let $UDV = M$ be a smith form of M as a matrix over $\mathbb{Z}_\ell$.

Since D is diagonal and U, V are invertible, we can easily read out the size of the subgroup generated by $[n]P_j$, and hence decide how much brute force is needed to reach $\phi\left(A\left[\ell^e\right]\right)$. Letting $\{a_i\}_{i=1}^{m'}$ be invariant factors of M that has ℓ-adic valuation more than e, then we get

$$\left|\sum \mathbb{Z}\left([n]P_i\right)\right| = \ell^{m'e} \prod_{i=1}^{m'} p^{-v_\ell(a_i)}$$

and by comparing with $\left|\phi\left(A\left[\ell^e\right]\right)\right|$, we can find how much brute force is needed.

The probability that we generate $\phi\left(A\left[\ell^e\right]\right)$ is given by Theorem 32 of the next section.

Remark 24 *Note that if the random points dont generate $A\left[\ell^e\right]$ but $\left[\phi\left(A\left[\ell^e\right]\right) : \sum_i \mathbb{Z}\left([n]P_i\right)\right]$ is small, we can brute force for the remaining part of the kernel.*

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

Example 25 *In the case of cryptosystems, we have $g = 2$ and the kernel is of the form $C_{\ell^e}^2$. In this case, Theorem 32 tells us that the probability that n faults is enough is $\left(1 - \ell^{-n}\right)\left(1 - \ell^{1-n}\right)$. This tells us that we need 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 $\frac{3}{8}$.

Example 26 *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 32 tells us that the probability that n faults is 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:*

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

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 $\frac{21}{64} \approx 0.33$.

Example 27 *As a final example, suppose that the kernel 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*

$$\left[\phi\left(A\left[\ell^e\right]\right) : \sum_i \mathbb{Z}\left([n]P_i\right)\right] = \ell^\delta$$

This is again given by Theorem 32 which gives us

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

assuming $\delta < k, e-k, e$.

2.2 Pseudocode and software simulation of the attack

Suppose we have abelian varieties A, B of dimension g 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: $\phi : A \to B$
$n \leftarrow \ell^{-ge}\,|A(\mathbb{F}_q)|$;
$P_i' \leftarrow [n]\phi(P_i)$; `/* Now we have` $P_i' \in \phi(A[\ell^e])$ `*/`
Determining how much bruteforce is needed;
Set Q_i as generators of $B[\ell^e]$;
Compute $M_{i,j} \in \mathbb{Z}_\ell$ with $P_i' = \sum_j M_{i,j}Q_j$; `/* Possible when` ℓ `is small */`
$UDV \leftarrow M$; `/* Smith normal 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 $B/\mathbb{Z}(DV)_i \cong A$ where the sum is over the rows ; `/* We only need to brute force vectors in`

$$B\left[\ \left[\phi\left(A\left[\ell^e\right]\right) : \sum_i \mathbb{Z}\left([n]P_i\right)\right]\ \right]$$

`See Lemma 23 and Example 27 for the details */`
$G \leftarrow \sum \mathbb{Z}P_i' = \phi\left(A\left[\ell^e\right]\right) := \ker\psi$;
Computing ϕ;
for $i \leftarrow 1$ **to** e **do**
  $A_i \leftarrow B/([\ell^{e-i}]G)$ $\psi_i : A_{i-1} \to A_i$;
  Bruteforce $\phi_i : A_i \to A_{i-1}$ to get $[\ell^i] = \psi_i[\ell^{i-1}]\phi_i$; `/* Possible when` ℓ `is small */`
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 bruteforcing.

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 bruteforcing the remaining parts of the group becomes feasible. Hence in this section, we compute the probability that given n random points, how likely are we to 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.

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

Definition 31

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

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

Suppose that $\phi(A[\ell^e])$ has m generators as a $\mathbb{Z}_\ell$-module. This now reduces the problem to computing the probability that with n random elements in $(\mathbb{Z}_\ell)^m$ generates the entire $\mathbb{Z}_\ell$-module, and if not, how 'much' is left. This is given by the following theorem (for convenience we use p instead of ℓ):

Theorem 32 *Let P_i be n random elements in $\mathbb{Z}_p^m$, distributed with respect to the normalized Haar measure and assume $m \leq n$. Let $M = \mathbb{Z}_p^m / \sum_{i=1}^n \mathbb{Z}P_i$ and denote the probability that $|M| = p^k$ is as $\mathbb{P}(m,n,k)$. Then*

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

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 bruteforce is needed. To understand the terms, we provide the following approximation:

Remark 33

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

In the case of $k = 0$, we have

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

Lemma 34 *Let $\mathcal{P}$ be a $m \times n$ matrix with $\mathcal{P}_{j,i} = (P_i)_j$. Suppose it's invariant factors are $a_1, a_2, \ldots, a_m$, then $|M| = \prod_{i=1}^m a_i$.*

Remark 35 *This allows us to consider the rows of matrix $\mathcal{P}$ instead of the columns P_i, giving us m vectors in $\mathbb{Z}_p^n$, making counting a lot easier.*

Lemma 36

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

Proof. We can work over $\mathbb{F}_p$ by reduction mod p and by Remark 35, the problem reduces to finding the number of ordered linearly independent m-tuples $(\mathcal{P}_i)_{i=1}^m$ over $\mathbb{F}_p^n$. We require

$$\mathcal{P}_{i+1} \in \mathbb{F}_p^n - \sum_{j=1}^{i} \mathbb{F}_p \mathcal{P}_i$$

and since

$$\begin{aligned}\left|\mathbb{F}_p^n - \sum_{j=1}^{i} \mathbb{F}_p \mathcal{P}_i\right| &= p^n - \left|\sum_{j=1}^{i} \mathbb{F}_p \mathcal{P}_i\right| \\ &= p^n - p^i\end{aligned}$$

Hence we get

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

Now we can prove Theorem 32.

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

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

and the boundary conditions $\mathbb{P}(m,n,0) = \left(p^{-n};p\right)_m$ and evidently $\mathbb{P}(1,n,k) = p^{-nk} - p^{-n(k+1)}$. The last one tells us that we can reasonably extend the function to $\mathbb{P}(0,n,k) = 0$.

From this we get the following recursions immediately

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

With this, we can prove the theorem by induction.

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

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

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

4 Application to genus two isogeny-based cryptography

This section describes the impact of the fault attack on existing genus two 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], and the Genus 2 Oblivious Transfer protocol of Fernàndez-València [Fer19].

4.1 Attack models on protocols

The fault attack model on the various protocols are similar. 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 because correct images of auxiliary points are required by SIDH and the genus two protocols. By faulting these points, these protocols are unable to be completed as specified and errors may be raised.

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.

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 fault attack.

In a static key exchange set-up, the long-term static secret key is usually protected against active attacks by a Fujisaki–Okamoto transform. But it should be noted that FO transforms will not be able to prevent the fault attack. This is because, throughout the validation process the static public key of Alice is only computed once and is never checked during validation. Hence the fault attack would not be detected by this validation and an adversary would be able to recover the secret isogeny. In an ephemeral key exchange set-up, the attack can still be mounted.

As noted in [TFMP21] in the fault attack model is 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.

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. After the long-term secret is computed, the images of the auxiliary points under the secret isogeny must also be computed. It is at this stage that the fault attack can be carried out in an attempt to recover the secret key.

Genus 2 Oblivious Transfer The genus two oblivious transfer has a long term secret key that can be attacked using the fault attack delineated in this paper.

4.2 Feasibility of attack models

The feasibility of the fault attack on various SIDH-like protocols have been discussed in by Ti [Ti17] and Tasso, De Feo, El Mrabet and Pontié [TFMP21]. In short, fault 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 fault attack are required to fully recover the secret key.

For the identification and oblivious transfer protocols, the presence of a long-term secret allows this fault attack to work well.

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 two version of this fault attack will retain this property of practicality.

4.3 Countermeasures

The countermeasure to thwart this attack is to implement order checking before the publication of the auxiliary points. This is the same as with the SIDH case.

5 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.

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.

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 for 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.

This loop abort technique is applicable to all the cryptographic protocols described above.

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.

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.

Fer19. Ramsès Fernàndez-València, *Genus 2 supersingular isogeny oblivious transfer*, IACR Cryptol. ePrint Arch. (2019), 758.

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.

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.