Breaking SIDH in polynomial time

DAMIEN ROBERT

Abstract. We show that we can break SIDH in (classical deterministic) polynomial time, even with a random starting curve E_0.

1. Introduction

1.1. **Result.** We extend the recent attacks by [CD22; MM22] and prove that there exists a proven deterministic polynomial time attack on SIDH [JD11; DJP14] / SIKE [JAC+17], even with a random starting curve E_0.

Both papers had the independent beautiful idea to use isogenies between abelian surfaces (using [Kan97, § 2]) to break a large class of parameters on SIDH. Namely, on a random starting curve E_0, if the degree of the secret isogenies are $N_A > N_B$, their attack essentially apply whenever $a := N_A - N_B$ is smooth. This is highly unlikely, however they use the fact that it is possible to tweak the parameters N_A and N_B to augment the probability of success (or reduce the smoothness bound on a), see Section 6. In the case where $\mathrm{End}(E_0)$ is known, [CD22] also have a (heuristic) polynomial time attack, essentially because one can use the endomorphism ring to compute an a-isogeny on E_0 even if a is not smooth, see Section 5.

A natural idea is to go in even higher dimension to extend the range of parameters on which an attack is possible, even on a random curve E_0. We show in Section 2 that by going to dimension 8, it is possible to break in polynomial time all parameters for SIDH.

Theorem 1.1. *We suppose that we are given the following input: we are given a secret N_B-isogeny over a finite field $\phi_B : E_0 \to E_B$ along with its images on (a basis of) the N_A-torsion points of E_0, where N_A and N_B are smooth coprime integers and $N_A > N_B$. We also assume that we are given the factorisations of N_A and N_B and (for simplicity) that we are given a basis of $E_B[N_B]$ and a decomposition of $N_A - N_B$ as a sum of four squares. Let $\mathbb{F}_q$ be the smallest field such that ϕ_B, and the points of $E_0[N_A]$ and $E_B[N_B]$ are defined*[1].

Then there is an explicit N_A-endomorphism $F : E_0^4 \times E_B^4$ in dimension $g = 8$ such that evaluating F at (P, P, P, P, Q, Q, Q, Q), for any $P \in E_0(\mathbb{F}_q), Q \in E_B(\mathbb{F}_q)$ allows to recover $\phi_B(P)$ and $\widetilde{\phi_B}(Q)$. Furthermore the kernel of F is described by 8 explicit rational generators which can be computed in time $O(\log N_A)$.

This reduces recovering ϕ_B to evaluating the isogeny F in dimension 8 given generators of its kernel. Using the algorithm of [LR22], such an isogeny can be evaluated, via the naive algorithm to compute smooth isogenies, in time $O(\ell_A^8 \log N_A + \log^2 N_A)$ where ℓ_A is the largest prime divisor of N_A. This cost can even be reduced to $\widetilde{O}(\ell_A^8 \log N_A)$ using the optimised computation of smooth isogenies of [DJP14, § 4.2.2].

In particular, we can find a basis for the kernel of ϕ_B in at most 2-evaluations of F on the basis of $E_B[N_B]$, for a total cost of $\widetilde{O}(\ell_A^8 \log N_A)$.

Date: September 11, 2022.

[1] We make no further assumptions on E_0 and E_B: we do not require them to be supersingular. In the context of SIDH, $\mathbb{F}_q$ will be the base field $\mathbb{F}_{p^2}$.

Remark 1.2.

- The decomposition of a as a sum of four squares is a precomputation step that only depends on N_A and N_B. It can be done in random polynomial time $O(\log^2 a)$ binary operations by [RS86].
- The attack is thus "quasi-linear", ie in $\widetilde{O}(\log N_A)$ arithmetic operations in $\mathbb{F}_q$, when $\ell_A = O(1)$, or even $\ell_A = O(\log\log N_A)$.
- In the context of SIDH, if $N_B > N_A$ we will simply try to recover Alice's secret isogeny Φ_A instead. By considering the dual isogeny $\tilde{F}$, we will also see in Section 6.4 that as in [QKL+21], in Theorem 1.1 it is also possible to directly reconstruct ϕ_B (with the same complexity) as long as $N_A^2 > N_B$.
- Another contribution of this paper is to give a precise (but heuristic, see Heuristic 4.4) complexity bound for a dimension 4 attack: $\widetilde{O}(\log N_A \ell_A^4)$ arithmetic operations (after a precomputation), see Section 4. This precise complexity bounds uses the fact mentioned above that we can also explicitly build an N_A^2-isogeny F rather than just a N_A-isogeny. This gives more freedom for the tweaking of parameters needed for the dimension 4 attack.
- The method of Sections 2 and 3 shows that the following powerful embedding lemma holds: for any N-isogeny $f : A \to B$ between abelian varieties of dimension g, and any $N' > N$, it is possible to efficiently embed f into an N'-isogeny F in dimension $8g$ (or $4g$ or $2g$ in certain cases). This provides considerable flexibility at the cost of going up in dimension, and was used in [Rob22b] to show that an isogeny over a finite field always admits an efficient representation.

1.2. **Outline.** We prove Theorem 1.1 in Section 2. This Section is written to be short and self contained, and since it applies in all cases, without requiring any parameter tweaks, the complexity analysis is straightforward. We recommend the reader, unless interested in the gory details of the dimension 2 and 4 attacks, to skip directly to this section.

For reasons stated in Remark 2.1, for practical attacks it would be more convenient to go in lower dimension. We first describe a common framework encapsulating possible dimension $2g$ attacks in Section 3, before describing our dimension 4 attack in Section 4. We explain how the dimension 2 attacks of [CD22; MM22] fit into this common framework in Section 5. Parameter tweaks, needed for the dimensions 2 and 4 attacks, are described in Section 6.

For this introduction, we give more context in Section 1.3 explain how our attacks fit into the broad class of "torsion point attacks" in Section 1.4, and summarize in Section 1.5 the different complexities of the different dim 2, 4 and 8 attacks of [CD22; MM22; Rob22a].

1.3. **Context.** Supersingular Isogeny Diffie-Hellman (SIDH) is a post-quantum key exchange protocol initially proposed in [JD11] with further ameliorations (among many other papers) in [DJP14; CLN16]. A standard transform gives a key encapsulation method SIKE (supersingular isogeny key encapsulation) [JAC+17] which was submitted to the NIST post-quantum competition, and recently selected as an alternative candidate in the fourth round of the competition.

The key hardness problem of many isogeny based protocols is based on the difficulty of recovering a large degree isogeny $f : E \to E'$ between two ordinary or supersingular elliptic curves, the so-called *isogeny path problem*. To the best of our knowledge, without more information on E and E' (like an explicit representation of part of their endomorphism rings) this problem still has *exponential quantum security for supersingular curves*.

However, for the SIDH key exchange, Bob will reveal not only the codomain E_B of his secret N_B-isogeny $\phi_B : E_0 \to E_B$ (N_B a large smooth number) but also the action of ϕ_B on the N_A*-torsion* $E_0[N_A]$ for an integer N_A prime to N_B, typically by revealing the image $Q_1 = \phi_B(P_1), Q_2 = \phi_B(P_2)$ of a basis (P_1, P_2) of $E_0[N_A]$. This added information then allows Alice to pushforward her secret N_A isogeny $\phi_A : E_0 \to E_A$ to $\phi'_A : E_B \to E_{AB}$, via $\operatorname{Ker} \phi'_A = \phi_B(\operatorname{Ker} \phi_A)$. Alice also reveals the action of her secret isogeny ϕ_A on $E_0[N_B]$, and then Bob can pushforward his secret N_B isogeny to $\phi'_B : E_A \to E_{AB}$ via $\operatorname{Ker} \phi'_B = \phi_A(\operatorname{Ker} \phi_B)$. The codomain is the same since the maps $\phi'_B \circ \phi_A : E_0 \to E_A \to E_{AB}$ and $\phi'_A \circ \phi_B : E_0 \to E_B \to E_{AB}$ have the same kernel $\operatorname{Ker} \phi_A + \operatorname{Ker} \phi_B$:

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] \arrow[d, "\phi_A"] & E_B \arrow[d, "\phi'_A"] \\
E_A \arrow[r, "\phi'_B"] & E_{AB}
\end{tikzcd}$$

The supersingular curve E_{AB} is then the common secret of Alice and Bob.

But as we will see, this is a *key weakness* that allows to break the SIDH key exchange. This is worth emphasizing once more: the work of [CD22; MM22; Rob22a] only breaks *SSI-T*, the supersingular isogeny with torsion problem, not the more general supersingular isogeny path problem. In particular, it does not apply to protocols like [CLM+18; DKL+20].

1.4. **Torsion points attacks.** Let us recall the setup. Eve wants to recover the secret N_B-isogeny ϕ_B, and she knows the image of ϕ_B on a basis of $E_0[N_A]$. It has been well known that the publication of these so called torsion points could, for some parameters, reduce the security of the supersingular isogeny problem.

Petit in [Pet17] had the first key idea of the following "torsion points" attack: assume that the attacker Eve could somehow combine Bob's secret N_B-isogeny ϕ_B and/or its dual $\widetilde{\phi_B}$ with an isogeny α she controls into a N_A-isogeny $F : E_0 \to E'$. Eve knows the action of ϕ_B on $E_0[N_A]$ because Bob published it, and she also knows the action of the dual isogeny $\widetilde{\phi_B} : E_B \to E_0$ on $E_B[N_A]$. Indeed, if (P_1, P_2) is a basis of $E_0[N_A]$, and $Q_1 = \phi_B(P_1)$, $Q_2 = \phi_B(P_2)$, then $\widetilde{\phi_B}(Q_1) = N_B P_1, \widetilde{\phi_B}(Q_2) = N_B P_2$. Notice that Q_1, Q_2 is a basis of $E_B[N_A]$ since N_A is prime to N_B.

Since she knows the action of α too because she controls it, she can recover the action of F on (a basis of) $E_0[N_A]$. It is then easy for Eve to compute the kernel of F using some linear algebra and discrete logarithms, see Lemma 3.3. These discrete logarithms are inexpensive because N_A is assumed to be smooth.

From this kernel $\operatorname{Ker} F$, she can then evaluate F on any point of E_0 via an isogeny algorithm, from which she can try to recover ϕ_B if extracting ϕ_B from F is possible.

In his attack, Petit considers for F an endomorphism of E_0 of the form $F = \widetilde{\phi_B} \circ \gamma \circ \phi_B + [d]$, where γ is a trace 0 endomorphism (meaning that $\tilde{\gamma} = -\gamma$) of degree e. Then it is easy to check that F is a $N_B^2 e + d^2$-isogeny, so it remains to find parameters such that $N_B^2 e + d^2 = N_A$, and to construct a γ of degree e. From the knowledge of F, it is not too hard to extract ϕ_B.

Remark 1.3. A variant is to "tweak" the parameters, in order to increase the range of susceptible parameters. For instance if we can find parameters such that $N_B^2 e + d^2 = u N_A$ with u smooth, then F will be an $u N_A$-isogeny. We only know its action on $E_0[N_A]$, so we cannot recover it directly. However F is a composition $F_2 \circ F_1$ of a N_A-isogeny F_1 followed by a u-isogeny F_2, so we can at least recover F_1 and then try to brute force F_2. A similar strategy holds for higher dimensional attacks, we will describe more possible tweaks in Section 6.

This attack, while powerful, can only apply to unbalanced parameters (here $N_A > N_B^2$), and requires the knowledge of a non trivial endomorphism of E_0. Further work, like [QKL+21], improves the range of parameters susceptible to these attacks, but still requires a non trivial endomorphism.

For SIKE's NIST submission, such an endomorphism is easy to find because the starting curve $E_0 = E_{\text{NIST}}$ is defined over $\mathbb{F}_p$. So in [Cos21], Costello argues that if this line of "torsion points" attacks is improved to reach the SIKE's parameters submitted to the NIST, a preventive measure would be to switch the starting elliptic curve E_0 to a "random" one, so that Eve has no prior informations on its endomorphism ring. (This was not considered for SIKE's submission because it would involve either a trusted multipartite setup to build E_0 or for Alice's to first walk a random path and publish a "random" E_0, hence adding some complexity to the key exchange.)

The second key breakthrough was in the recent attacks by [CD22; MM22] by Castryck–Decru and Maino–Martindale respectively (we refer to Sections 1.5 and 5 for more details on these two articles). They both, independently, had the beautiful idea that it is possible to extend the range of parameters susceptible to "torsion points" attack by constructing a N_A-isogeny F in dimension 2, on a product of two supersingular curves. Indeed, going up in dimension largely opens up the range of isogeny we can construct explicitly.

They exploit the following (easy) lemma, due to Kani in [Kan97] as part of his deep work on classifying covers $C \to E$ of elliptic curves by genus 2 curves: given a N_B-isogeny $\phi_B : E_0 \to E_B$ and a a-isogeny $\alpha : E_0 \to E'$, with a prime to N_B, it is possible to build an explicit $a + N_B$-isogeny $F : E_0 \times E'' \to E_B \times E'$ in dimension 2 (see Section 3 for a generalisation to dimension g). This means, assuming $N_A > N_B$, that Eve can break SIDH as long as she can find a $a = N_A - N_B$ isogeny from E_0.

This is in particular the case whenever a is smooth, and is the focus of Maino and Martindale's article (Castryck and Decru also consider this case briefly). While the probability to get a smooth a is small, tweaking the parameters can increase it, and subsequent analysis by De Feo showed that this gives a (heuristic) subexponential $L(1/2)$ attack. In particular, torsion points attacks can apply even to "random curves"!

Castryck and Decru furthermore exploit the fact that for the NIST submission, the curve $E_0 = E_{\text{NIST}}$ is either $y^2 = x^3 + x$ or $y^2 = x^3 + 6x^2 + x$. It has an explicit endomorphism $2i$, hence it is easy to construct an a-isogeny α (which can be evaluated efficiently) whenever $a = a_1^2 + 4a_2^2$. In particular, they obtain a (heuristic) polynomial time attack for this specific E_0 (assuming the factorisation of a is precomputed).

Our current work stems from the fact that it is easy to extend Kani's lemma to dimension g abelian varieties (see Section 3). Namely, from a a-isogeny and a N_B-isogeny in dimension g (with a prime to N_B), we can build an explicit $a + N_B$-isogeny in dimension $2g$. We will apply this to the diagonal embedding of ϕ_B to $E_0^g \to E_B^g$, this is still an N_B-isogeny, so it remains to find an a-isogeny on E_0^g, where $a = N_A - N_B$. We then exploit that even if we do not know $\text{End}(E_0)$, on E_0^2 we can always build endomorphisms of the form $\alpha = \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$, which give $a_1^2 + a_2^2$-endomorphisms. Hence we get a dimension $2g$ attack, $g = 2$, whenever $a = a_1^2 + a_2^2$ (eventually after parameter tweaks).

And of course the general case stems from the fact that an integer is always a sum of four squares: $a = a_1^2 + a_2^2 + a_3^2 + a_4^2$ [Δ1650; Lag70], from which we can then build a a-endomorphism α on E_0^4 in dimension $g = 4$, hence get a dimension $2g = 8$ attack. The fact that there always exist a-endomorphisms on A^4 for any abelian variety A and any integer a was first used by Zarhin in [Zar74] and is known as "Zarhin's trick" or the "quaternion trick".

We remark also that unlike the decomposition of a as a sum of two squares, which requires its factorisation, the decomposition as a sum of four squares can be done in (random) polynomial time, see Remark 1.2. It is then easy to build by hand a $N_B + a$-endomorphism on $E_0^4 \times E_B^4$, we will see in Section 2 that $F = \begin{pmatrix} \alpha & \widetilde{\phi_B} \\ -\phi_B & \tilde{\alpha} \end{pmatrix}$ fits.

As mentioned above, this endomorphism F can be seen as a special case of the dimension g generalisation in Section 3 of Kani's lemma to build isogenies on product of abelian varieties. But it can also be seen as a variant of Petit's endomorphism to higher dimension. Indeed, if F_1 is a d_1-endomorphism and F_2 is a d_2-endomorphism, then $F_1 + F_2$ is a $d_1 + d_2$-endomorphism whenever $\widetilde{F_1}F_2 = -\widetilde{F_2}F_1$. Our dimension 8 endomorphism is the case $F = F_1 + F_2$ with $F_1 = \begin{pmatrix} \alpha & 0 \\ 0 & \tilde{\alpha} \end{pmatrix}$ a a-endomorphism and $F_2 = \begin{pmatrix} 0 & \widetilde{\phi_B} \\ -\phi_B & 0 \end{pmatrix}$, a N_B-endomorphism. Petit's endomorphism $F = \widetilde{\phi_B} \circ \gamma \circ \phi_B + [d]$ is the case where $F_1 = \widetilde{\phi_B} \circ \gamma \circ \phi_B$ is antisymmetric (ie of trace 0, ie $\widetilde{F_1} = -F_1$) and $F_2 = [d]$ is symmetric (ie $\widetilde{F_2} = F_2$), with $F_1 F_2 = F_2 F_1$.

1.5. Complexities of the different attacks.

The article by Castryck and Decru was published first in 2022-07-30, with only minor revisions since. As mentioned above, this article mainly focuses on the dimension 2 attack when $E_0 = E_{\text{NIST}}$ is NIST's starting curve, ie contains the endomorphism $2i$. In this case they obtain a heuristic polynomial time algorithm (with no explicit bound).

The heuristic is due to two reasons. First in [CD22], Castryck and Decru guess a starting path for ϕ_B and use F as an oracle to know if the guess was correct or not, then they iterate the process. The heuristic is then that if a wrong path is guessed, the codomain of F will be a Jacobian of a superspecial curve rather than a product of two supersingular elliptic curves. Assuming heuristically that the codomain of F for a wrong guess is uniform among all superspecial surfaces, the probability of a mistake is $\approx 1/p$, hence negligible. But, as first noticed by Maino and Martindale in [MM22], and also independently by Oudompheng [Oud22], Petit, and Wesolowski [Wes22b], the isogeny F allows to directly recover ϕ_B. This gives a more direct attack (no need to guess many isogenies), and removes the first heuristic.

The second reason is that for their attack to work on the starting curve $E_0 = E_{\text{NIST}}$, they need $a = N_A - N_B$ to be of the form $a = a_1^2 + 4a_2^2$. In this case they can build an a-isogeny α which can be evaluated in $O(\log a)$ arithmetic operations. For a uniform integer less than x, the probability to be decomposed in this form is roughly $1/\sqrt{\log x}$ (see Remark 4.2), so assuming that parameter tweaks behave like uniform integers, we may assume that we can tweak the parameters without increasing their size too much in such a way that the attack can apply. Also this decomposition (which is a precomputation) supposes access to a factorisation oracle; hence is in polynomial time only in the quantum model.

This second heuristic (and the need for factorisation) can be removed using work by Wesolowski [Wes22b] explaining how to directly build a $N_A - N_B$-isogeny α when $\operatorname{End}(E_0)$ is known. More precisely, Wesolowski builds an ideal I_α of norm a which represents α, and evaluating α on a point is done by using [FKM+22, Lemma 3.3]. Constructing this isogeny and then evaluating it on a point can be done in polynomial time, but there is no clear complexity bound as of yet. But the evaluation of α on a basis of $E_0[N_A]$ can be seen as a polynomial time precomputation, depending on E_0. Via this precomputation, the attack then reduces to evaluating a N_A-isogeny F in dimension 2.

We mention also that Castryck and Decru implemented their attack in Magma (so far this is the only publicly available implemented attack), which showed that it was practical, breaking Microsoft's and the NIST parameters. The timings were then considerably improved in an open source reimplementation in Sage [POP+22], where Oudompheng implemented the direct isogeny recovery of [MM22] and the extended parameter tweaks of [Rob22a] (see Section 5).

The article by Maino and Martindale was published in 2022-08-08, with a second major revision in 2022-08-25, fixing an error where their initial endomorphism candidate did not respect the product polarisations. They focus on the case where $\mathrm{End}(E_0)$ is not known, case which is also briefly investigated by Castryck and Decru. The first version does not contain a complexity estimate, but in the second version they use an analysis due to De Feo which shows that, using slightly more general parameter tweaks, they have an heuristic subexponential $L(1/2)$ attack.

This current article [Rob22a] was first published in 2022-08-11 (it's better to forget about the 2022-08-10 version which contained typos in the definition of the matrix F…) focusing mainly on the polynomial time dimension 8 attack (and explaining very briefly the dimension 4 attack). There was a revision on 2022-08-23 expanding on the dimension 4 attack and another revision on 2022-08-25 giving a general dimension $2g$ attack framework that shows how the dimension 2 attacks of Castryck–Decru and Maino–Martindale and our dimension 4 and 8 attacks all fit together. A further revision was published in 2022-09-02 to expand the introduction and mention the complexity result of the second version of [MM22]. The current version was published in 2022-09-11 to give a precise heuristic and complexity bound for the dimension 4 (and 2) attacks. We expect a last revision once the dimension 4 and 8 are finished to be implemented in order to give explicit timings.

At the time of its publication, [Rob22a] was the only one containing a precise complexity estimate, and the only available polynomial time attack (with or without random starting curve) with no heuristics. Due to the work of Wesolowski and De Feo mentioned above, and the improved parameters tweaks of Section 6, the current situation (as far as I am aware) is now as follow:

- When $E_0 = E_{\mathrm{NIST}}$ is NIST's starting curve, the attack of Castryck-Decru using the endomorphism $2i$ (as implemented in [POP+22]) is in heuristic polynomial time. We refer to Proposition 5.1 for a complexity analysis: We can find a decomposition $N_A = (b_1 + 4b_2^2)N_B/D + (a_1 + 4a_2^2)$ where D is a divisor of N_B heuristically of magnitude $\Theta(\log N_B)$ in $O(\log^3 N_A)$ binary operations for this precomputation step. The attack is then in $\widetilde{O}(D \log N_A \ell_A^2) = \widetilde{O}(\log^2 N_A \ell_A^2)$ arithmetic operations. We can reduce the magnitude of D to $\Theta(\sqrt{(}\log N_B))$ (heuristically) at the price of doing $O(\sqrt{\log N_B})$ factorisation calls in the precomputation. The attack is then in $\widetilde{O}(\log^{1.5} N_A \ell_A^2)$ arithmetic operations.

  Using [Wes22b], the dimension 2 attack can also apply to any elliptic curve with known endomorphism ring in proven polynomial time (but the exact degree has not been bounded yet). More precisely, after a polynomial time precomputation to construct the a-isogeny α and its action on a basis of $E_0[N_A]$, the attack is the same as in Theorem 1.1 except that F is computed in dimension 2, hence its evaluation costs $\widetilde{O}(\log N_A \ell_A^2)$ arithmetic operations in $\mathbb{F}_q$, see Proposition 5.2.
- When E_0 is a "random" curve, the dimension 2 attack of Maino and Martindale (and also Castryck and Decru) is in (heuristic) subexponential time $L(1/2)$ [MM22].

The dimension 4 attack of Section 4 is in heuristic polynomial time (because it needs parameter tweaks). The precomputation is very similar to the precomputation done for Castryck-Decru using the endomorphism $2i$ (because both attacks rely on decomposing an integer as a sum of two squares), except that in this case we can also build a N_A^2-isogeny with no added (asymptomatic) cost by Section 6.4. Under Heuristic 4.4, the precomputation costs $O(\log^3 N_A)$ binary operations to find a decomposition $N_A^2 = (b_1^2 + 2b_2)^2 N_B + (a_1^2 + a_2^2)$, and then the attack is in $\widetilde{O}(\log N_A \ell_A^4)$ arithmetic operations by Proposition 4.6. We stress that for the dimension 4 attack the heuristic only concerns the average complexity of finding this decomposition of N_A^2 (provided it exists), not the attack itself.

The dimension 8 attack of Section 2 is in proven polynomial time, and is in $\widetilde{O}(\log N_A \ell_A^8)$ arithmetic operations by Theorem 1.1. The precomputation step is the decomposition of $N_A - N_B$ as a sum of four squares and can be done in randomized $O(\log^2 N_A)$ binary operations.

The dimension 8 (resp. 4) attack remains the only proven (resp. heuristic) polynomial time attacks for a random curve E_0.

- When $\ell_A = O(1)$ (or even $O(\log\log N_A)$), the dimension 8, dimension 4, and if $\mathrm{End}(E)$ is known, the dimension 2 attacks, all have quasi-linear complexity of $\widetilde{O}(\log N_A)$ arithmetic operations.

  The constants involved will be larger for the higher dimensional attack, however the precomputation of the dimension 8 attack is faster than the precomputation of the dimension 2 attack. Furthermore, in dimension 2, when E has known endomorphisms but is not E_{NIST}, the precomputation step also depends on the starting curve E_0. An implementation is ongoing to compare timings.

1.6. **Thanks.** Many thanks are due to the persons who commented on the prior versions. Special thanks to Benjamin Wesolowski and Marco Streng, for suggesting to simply use $b = 1$ in the dimension 8 attack. This significantly simplify the description of the attack in this case. (Although as noted above the general $b > 0$ case is still useful for the dimension 4 attack).

This work was supported by the ANR `ANR-19-CE48-0008` project `Ciao`.

2. Dimension 8 attack

Since $N_A > N_B$, write $N_A = N_B + a$ for a positive integer $a > 0$. It is harmless to suppose that N_A is prime to N_B, otherwise if $d = \gcd(N_A, N_B)$, we could recover the kernel of a d-isogeny through which ϕ_B factors (since we know its action on $E_0[d] \subset E_0[N_A]$), so we could reduce to solving the problem with new coprime parameters $N'_A = N_A/d$, $N'_B = N_B/d$.

As N_A is prime to N_B, $\gcd(N_A, a) = 1$. Let $M \in M_4(\mathbb{Z})$ be a 4×4 matrix such that $M^T M = a\,\mathrm{Id}$. Explicitly we write $a = a_1^2 + a_2^2 + a_3^2 + a_4^2$ and take

$$M = \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix},$$

the matrix of the multiplication of $a_1 + a_2 i + a_3 j + a_4 k$ in the standard quaternion algebra $\mathbb{Z}[i, j, k]$ [Ham44]. Let α_0 be the endomorphism on E_0^4 given matricially by M, The dual (with respect to the product principal polarisation) $\tilde{\alpha}_0$ of α_0 is given matricially by M^T (since integer multiplications are their own dual), so $\tilde{\alpha}_0\alpha_0 = a\,\mathrm{Id}$, hence α_0 is an a-isogeny, which can be evaluated in $O(\log a)$ arithmetic operations. We let α_B be the endomorphism of E_B^4 given by the same matrix M, and by abuse of notation we denote by $\phi_B\,\mathrm{Id} : E_0^4 \to E_B^4$ the diagonal embedding of $\phi_B : E_0 \to E_B$. We remark that since α_0 is given by an integral matrix, it commutes with ϕ_B in the sense that we have the equation: $\phi_B\alpha_0 = \alpha_B\phi_B$:

$$\begin{tikzcd}
E_0^4 \arrow[r, "\phi_B\,\mathrm{Id}"] \arrow[d, "\alpha_0"] & E_B^4 \arrow[d, "\alpha_B"] \\
E_0^4 \arrow[r, "\phi_B\,\mathrm{Id}"] & E_B^4
\end{tikzcd}$$

Let $F = \begin{pmatrix} \alpha_0 & \widetilde{\phi_B}\,\mathrm{Id} \\ -\phi_B\,\mathrm{Id} & \widetilde{\alpha_B} \end{pmatrix}$, where $\widetilde{\phi_B}$ is the dual isogeny $E_B \to E_0$ of ϕ_B. F is an endomorphism on the 8-dimensional abelian variety $X = E_0^4 \times E_B^4$. Since the dual $\tilde{F}$ of F is given by $\tilde{F} = \begin{pmatrix} \widetilde{\alpha_0} & -\widetilde{\phi_B}\,\mathrm{Id} \\ \phi_B\,\mathrm{Id} & \alpha_B \end{pmatrix}$ by Lemma 3.2, we compute

$$\tilde{F}F = F\tilde{F} = \begin{pmatrix} N_B + a & 0 \\ 0 & N_B + a \end{pmatrix} = N_A\,\mathrm{Id}\,.$$

Hence F is an N_A-isogeny on X (with respect to the product polarisations).[2]

As in Section 1.4, the action of F on the N_A-torsion is explicit, hence we can recover its kernel. But in this case we can directly recover $\operatorname{Ker} F$ as follow: it is given by the image of $\tilde{F}$ on $X[N_A]$. Furthermore, since a is prime to N_A, the kernel of F is exactly the image of $\tilde{F}$ on $E_0^4[N_A] \times 0$, so we immediately get the 8 generators $(g_1, \dots, g_8)$ of the kernel $\operatorname{Ker} F = \{(\widetilde{\alpha_0}(P), (\phi_B\,\mathrm{Id})(P)) \mid P \in E_0^4[N_A]\}$. This step costs $O(\log a)$ arithmetic operations in $E_0(\mathbb{F}_q)$.

We can then compute F (on any point $P \in X(\mathbb{F}_q)$) using an isogeny algorithm in dimension 8, decomposing the N_A-endomorphism F as a chain of ℓ-isogeny for ℓ the prime factors of N_A. If ℓ_A is the largest prime divisor of N_A, the complexity of the first ℓ_A-isogeny computation will first be $\widetilde{O}(\log N_A)$ arithmetic operations in $A(\mathbb{F}_q)$ to compute the multiples $\frac{N_A}{\ell_A} g_i$, followed by the individual ℓ_A-isogeny computations on P and the g_i. These isogenies computations cost $O(\ell_A^8)$ operations over $\mathbb{F}_q$ using [LR22]. Since we compute a composition of at most $O(\log N_A)$ isogenies, the total cost of evaluating F on P is $O(\log^2 N_A + \log N_A \ell_A^8 \log \ell_A)$. This naive method uses $O(\log N_A)$ ℓ-isogeny calls where $\ell \mid N_A$, and multiplications which cost $O(\log^2 N_A)$ in total. The optimised method of [DJP14, § 4.2.2] shows that by increasing the number isogeny calls to $\widetilde{O}(\log N_A)$, the multiplications cost can be reduced to $\widetilde{O}(\log N_A)$ multiplications by $\ell \mid N_A$. This optimised version thus costs $\widetilde{O}(\ell_A^8 \log N_A + \ell_A \log N_A) = \widetilde{O}(\ell_A^8 \log N_A)$. (Note that since a ℓ-isogeny in dimension 8 is going to be much more expansive than a multiplication by ℓ, for practical attacks it will be important to apply the optimised *weighted* strategy of [DJP14, § 4.2.2] rather than their *balanced* strategy.)

Remark 2.1. The isogenies computations in [LR22; BCR10; Som21] use a (level $m = 4$ or $m = 2$) theta model of X, which we can compute as the (fourfold) product theta structure of the theta models of E_0 and E_B. It is also well known how to switch between the theta model

[2] We refer to Section 3 for the definition of an N-isogeny between principally polarised abelian varieties in dimension g.

and the Weierstrass model on an elliptic curve, and it is not hard to extend the conversion to the product of elliptic curves, since the product theta structure is given by the Segre embedding. The arithmetic on the theta models can be done in $O(1)$ arithmetic operations in a $O(1)$-extension of $\mathbb{F}_q$ (if $8 \mid N_A N_B$ the theta model will already be rational). However the big $O()$ notation hides an exponential complexity in the dimension g. In dimension 8 and level $m = 4$, the theta model uses 2^{16} coordinates, so we would need in practice to switch to the *Kummer* model by working in level $m = 2$ which "only" requires 2^8 coordinates. This is another reason why we would prefer to compute an endomorphism in dimension $g = 4$ rather than $g = 8$: in dimension 4 we would only need 2^8 coordinates in level $m = 4$, or 2^4 coordinates in level $m = 2$.

Thus we can evaluate F on any point of X, so we can evaluate ϕ_B or $\widetilde{\phi}_B$ on any point of E_0 (resp. E_B). We can now recover the kernel of ϕ_B on E_0 as the image of $\widetilde{\phi}_B$ on $E_B[N_B]$. If (Q_1, Q_2) is a basis of $E_B[N_B]$, we compute $Q'_i = \widetilde{\phi}_B(Q_i)$ by evaluating F on the point $(0, 0, 0, Q_i, 0, 0, 0)$, and the kernel of ϕ_B is generated by whichever Q'_i has order N_B. If $\omega(N_B)$ is the number of distinct prime divisors of N_B, this step costs $O(\omega(N_B) \log N_B)$ operations in $E_0(\mathbb{F}_q)$ (which can be improved to $O(\log N_B \log\log N_B)$ using a binary product tree) along with two calls to the evaluation of F.

This concludes the complexity analysis of Theorem 1.1.

Remark 2.2.

- It is immediate to generalize Theorem 1.1 to recover an N_B-isogeny ϕ_B between abelian varieties E_0, E_B of dimension g. The attack reduces to computing one N_A-isogeny in dimension $8g$ (or eventually $4g$ or even $2g$ if the parameters allow for it).

  The same proof as above holds; the complexity of evaluating the dimension $8g$ N_A-isogeny will be $\widetilde{O}(\log N_A \ell^{8g})$ arithmetic operations using [LR22] and the fast smooth isogeny computation of [DJP14, § 4.2.2].

  We recover Ker ϕ_B as the image of $\widetilde{\phi}_B$ on a $2g$-dimensional basis of $E_B[N_B]$, hence we get $2g$ generators. To extract a g dimensional basis of the kernel from these generators, we can take any g points and check if the Weil pairing matrix with a basis of $E_0[N_B]$ has full rank (we expect this will be the case with high probability). This can be done by computing the determinant of $g \times g$ submatrices and testing if it is of primitive N_B-order. Hence, since the dimension g is fixed, this still costs $O(\log N_B)$.An alternative to reduce the complexity in g is to compute discrete logarithms using Pohlig-Hellman's algorithm in $\widetilde{O}(\log N_B \ell_B^{1/2})$ (see the proof of Lemma 3.3) so that we may use linear algebra to extract a full rank submatrix.
- When $\ell_A = O(1)$, or even $\ell_A = O(\log\log N_A)$, we can use a SIDH style fast evaluation of the N_A-isogeny F as in [DJP14, § 4.2.2]. The attack to recover generators of Ker ϕ_B then becomes "quasi-linear": $\widetilde{O}(\log N_A)$ arithmetic operations, hence as efficient asymptotically as the key exchange itself (with a higher constant of course).
- The attack also breaks the TCSSI-security assumption of [DDF+21, Problem 3.2].

3. Dimension $2g$ attack

We first generalize the construction of Section 2, and then show how it can be applied (in certain cases) to mount an attack in dimension 4 or 2.

3.1. N-isogenies.

Definition 3.1. An N-isogeny $f : (A, \lambda_A) \rightarrow (B, \lambda_B)$ of principally polarised abelian varieties is an isogeny such that $f^* \lambda_B := \hat{f} \circ \lambda_B \circ f = N \lambda_A$, where $\hat{f} : \hat{B} \rightarrow \hat{A}$ is the dual isogeny. Letting $\tilde{f} = \lambda_A^{-1} \hat{f} \lambda_B$ be the dual isogeny $\tilde{f} : B \rightarrow A$ of f with respect to the principal polarisations, this condition is equivalent to $\tilde{f} f = N$.

If Θ_A is a divisor associated to λ_A, sections of $m\Theta_A$ gives coordinates on A (if $m \geq 3$ we get a projective embedding by Lefschetz' theorem). Given a suitable model of $(A, m\Theta_A)$, a representation of the kernel $K = \mathrm{Ker} f$ of an N-isogeny f (for instance coordinates for its generators), and the coordinates of a point $P \in A$, an N-isogeny algorithm will output a suitable model of $(B, m\Theta_B)$ and the coordinates of the image $f(P)$ in this model. For instance, the N-isogeny algorithm from [LR22] uses a theta model of level $m = 2$ or $m = 4$, and in dimension g can compute the image of an N-isogeny in $O(N^g)$ arithmetic operations over the base field (where the theta model is defined).

Note that in general, for an N-isogeny algorithm, we only have the kernel K and the source polarised abelian variety (A, Θ_A). We first need to check that the divisor $N\Theta_A$ descends through the isogeny $f : A \rightarrow B = A/K$. This implies that K must be a subgroup of $K(N\Theta_A)$, the kernel of the polarisation $N\lambda_A : A \rightarrow \hat{A}$ associated to $N\Theta_A$. And by descent theory [Mum66, Proposition 1 p.291; Mum70, Theorem 2 p. 231], the descents of $N\Theta_A$ correspond exactly to level subgroups $\widetilde{K}$ of K in Mumford's theta group $G(N\Theta_A)$. Hence $N\Theta_A$ descends if and only if K is isotropic for the commutator pairing of $G(N\Theta_A)$ (and the descent Θ_B will be of degree one if and only if K is maximal isotropic by a standard degree computation). Mumford proves in [Mum70, (5) p.229] that this commutator pairing is yet another incarnation of the Weil pairing. So the descent condition is thus equivalent to K being maximal isotropic for e_{N, Θ_A} in $A[N]$, as is well known (see eg [Kan97, Proposition 1.1]). Such a K is usually the entry point of an N-isogeny algorithm.

Our current situation is different: we already have a target codomain B with a polarisation λ_B, and we want $N\Theta_A$ to descend to λ_B, not just any other principal polarisation λ'_B (on which there will be many, see Remark 3.6). So it does not suffice to check that $\mathrm{Ker} f$ is maximal isotropic for the Weil pairing, we want $f^* \Theta_B \simeq N\Theta_A$ (isomorphism up to algebraic equivalence), ie $\tilde{f} \circ f = N$.

If this condition is satisfied, we know that $N\Theta_A$ descend, hence by the above discussion we automatically know that $\mathrm{Ker} f$ is maximal isotropic. Another way to see that without invoking descent theory is to use the fact that $\mathrm{Ker} f = \mathrm{Im} \tilde{f} \mid B[N]$, and that since $\hat{f}$ is the dual of f for the Weil pairings $e_{A,N}$ on $(A \times \hat{A})[N]$ and $e_{B,N}$ on $(B \times \hat{B})[N]$, then $\tilde{f}$ is the dual of f for the Weil pairings $e_{\lambda_A, N}$ on $(A \times A)[N]$ and $e_{\lambda_B, N}$ on $(B \times B)[N]$. In particular, if $x, y \in \mathrm{Ker} f$, $x = \tilde{f}(x'), y = \tilde{f}(y')$ for $x', y' \in B[N]$, so $e_{\lambda_A, N}(x, y) = e_{\lambda_A, N}(\tilde{f}(x'), \tilde{f}(y')) = e_{\lambda_B, N}(x', f \circ \tilde{f}(y')) = e_{\lambda_B, N}(x', Ny') = 1$.

We need the following standard Lemma:

Lemma 3.2. *If* $F = \begin{pmatrix} a & b \\ c & d \end{pmatrix} : (A, \lambda_A) \times (B, \lambda_B) \rightarrow (C, \lambda_C) \times (D, \lambda_D)$, *then for the product polarisations on* $A \times B$ *and* $C \times D$, $\tilde{F} = \begin{pmatrix} \tilde{a} & \tilde{c} \\ \tilde{b} & \tilde{d} \end{pmatrix}$.

Proof. Recall that we have a canonical isomorphism $\hat{A} \simeq \mathrm{Pic}^0(A)$, and that under this isomorphism the dual of f is given by $\hat{f} = f^*$. This shows that $\hat{F} : \hat{C} \times \hat{D} \rightarrow \hat{A} \times \hat{B}$ is given by $\hat{F} = \begin{pmatrix} \hat{a} & \hat{c} \\ \hat{b} & \hat{d} \end{pmatrix}$ (see eg [MGE12, Proposition 11.28]). Since the product polarisations act component by component by definition (see eg the proof of [BL04, Corollary 5.3.6] or the proof of [Kan16, Proposition 61]), we then get that $\tilde{F} = \begin{pmatrix} \tilde{a} & \tilde{c} \\ \tilde{b} & \tilde{d} \end{pmatrix}$. □

We will also use the fact that once we have evaluated an isogeny on a basis of the N-torsion it is easy to evaluate it on any other N-torsion point:

Lemma 3.3. *Let $f : A \to B$ be an isogeny between abelian varieties. Assume that the N-torsion of A is rational and that we are given a basis $(P_1, \dots P_{2g})$ of it. Then given the evaluation $f(P_i)$ of all P_i, it is possible to evaluate f on a point $P \in A[N]$ in time $\widetilde{O}(\log N \ell_N^{1/2})$ arithmetic operations.*

Furthermore, if f is an N-isogeny and we are given a rational basis of $B[N]$, it is possible to recover generators for its kernel Ker$f$ *in $\widetilde{O}(\log N \ell_N^{1/2})$ arithmetic operations.*

Proof. Given a point $P \in A[N]$, we can evaluate the Weil pairing $e_{W,N}(P, P_i)$ in $O(\log N)$ arithmetic operations (this assumes we work over a model which can compute the Weil pairing; this will be the case in the theta model by [LR10; LR15]).

From the Weil pairing matrix of the $e_{W,N}(P_i, P_j)$, we can first do $O(g^2)$ discrete logarithm computations from a N-th root of unity ζ to get a matrix with coefficients in $\mathbb{Z}/N\mathbb{Z}$. By linear algebra over $\mathbb{Z}/N\mathbb{Z}$, it is easy to compute a symplectic basis $(a_1, \dots, a_g, a'_1, \dots a'_{2g})$ of the N-torsion, along with the values of f on this basis. Using a naive linear algebra algorithm, this can be done in $O(g^3 \log N)$. The dominant cost will be the discrete logarithms.

The Pohlig-Hellman algorithm [PH78] has complexity $O(E \log N \ell_N^{1/2})$ operations in A, where if $N = \prod \ell_i^{e_i}$, $E = \sum e_i$. The iterative version of Pohlig-Helmman's algorithm which increases the current exponent e in the ℓ_i-discrete logarithm by 1 at each step, can be replaced by a Newton like version which double the precision. This faster variant, described in [Sho09, §11.2.3], has complexity $\widetilde{O}(\log N_A \ell_A^{1/2})$.

Given the symplectic basis, one can decompose a point P in this basis by $O(g)$ calls to the Weil pairing and discrete logarithms. Evaluating $f(P)$ can thus be done in $\widetilde{O}(\log N_A \ell_A^{1/2})$. If $P = \sum_{i=1}^{g} \lambda_i a_i + \lambda'_i a'_i$, $f(P) = \sum_{i=1}^{g} (\lambda_i f(a_i) + \lambda'_i f(a'_i))$.

If Ker$f \subset A[N]$, and we are given a rational basis of $B[N]$, we can first transform this into a symplectic basis $(b_1, \dots, b_g, b'_1, \dots b'_g)$ as above. We can express $f(P_i)$ in this basis using the Weil pairing and discrete logarithms, and solve a linear system over $\mathbb{Z}/N\mathbb{Z}$. Once again the discrete logarithms will dominate the complexity analysis. □

3.2. **Isogeny diamonds.** The endomorphism F of Section 2 is a particular case of a construction due to Kani for $g = 1$ [Kan97, § 2], which generalizes immediately to $g > 1$.

We define a (d_1, d_2)-isogeny diamond as a decomposition of a $d_1 d_2$-isogeny $f : A \to B$ between principally polarised abelian varieties of dimension g into two different decompositions $f = f'_1 \circ f_1 = f'_2 \circ f_2$ where f_1 is a d_1-isogeny and f_2 is a d_2-isogeny. Then f'_1 will be a d_2-isogeny and f'_2 a d_1-isogeny:

$$\begin{tikzcd}
A \arrow[r, "f_1"] \arrow[d, "f_2"] & A_1 \arrow[d, "f'_1"] \\
A_2 \arrow[r, "f'_2"] & B
\end{tikzcd}$$

Lemma 3.4 (Kani). *Let $f = f'_1 \circ f_1 = f'_2 \circ f_2$ be a (d_1, d_2)-isogeny diamond as above. Then $F = \begin{pmatrix} f_1 & \widetilde{f'_1} \\ -f_2 & \widetilde{f'_2} \end{pmatrix}$ is a d-isogeny $F : A \times B \to A_1 \times A_2$ where $d = d_1 + d_2$.*

Its kernel is given by the image of $\tilde{F} = \begin{pmatrix} \widetilde{f_1} & -\widetilde{f_2} \\ f_2' & \widetilde{f_2'} \end{pmatrix}$ *on* $(A_1 \times A_2)[d]$*. If* d_1 *is prime to* d_2*, we also have* $\operatorname{Ker} F = \{(\widetilde{f_1}(P), f_2'(P)) \mid P \in A_1[d]\}$*, the kernel is thus of rank* $2g$*.*

Proof. We check, using Lemma 3.2, that $\tilde{F}F = d\,\mathrm{Id}$. Furthermore if d_1 is prime to d_2, then the restriction of $\tilde{F}$ to $A_1[d] \times \{0\}$ is injective, hence its image spans the full kernel since $\# A_1[d] = d^{2g}$. □

The matrix F from Section 2 is a special case of Lemma 3.4 where $A = E_0^g, B = E_B^g$ and F is actually an endomorphism.

3.3. **Description of the attack.** Write $N_A = N_B + a, a > 0$. Suppose that we can find an explicit a-isogeny $\alpha_0 : E_0^g \to X_0$. Then we can consider the following pushout:

$$\begin{tikzcd}
E_0^g \arrow[r, "\phi_B"] \arrow[d, "\alpha_0"] & E_B^g \arrow[d, "\alpha_B"] \\
X_0 \arrow[r, "\phi_B'"] & X_B
\end{tikzcd}$$

Hence we have the following isogeny diamond

$$\begin{tikzcd}
X_0 \arrow[r, "\widetilde{\alpha_0}"] \arrow[d, "\phi_B'"] & E_0^g \arrow[d, "\phi_B"] \\
X_B \arrow[r, "\widetilde{\alpha_B}"] & E_B^g
\end{tikzcd}$$

so by Lemma 3.4, $F = \begin{pmatrix} \widetilde{\alpha_0} & \widetilde{\phi_B} \\ -\phi_B' & \alpha_B \end{pmatrix}$ is a N_A-isogeny $F : X_0 \times E_B^g \to E_0^g \times X_B$. In particular, $\operatorname{Ker} F$ is the image of $\tilde{F}$ on $(E_0^g \times X_B)[N_A]$. Since a is prime to N_b, it is also the image of $\tilde{F}$ on $E_0^g[N_A] \times 0$: $\operatorname{Ker} F = \{(\alpha_0(P), \phi_B(P)) \mid P \in E_0^g[N_A]\}$. In particular, we don't need to build X_B, we will recover it when evaluating F.

Notice that if $\alpha_0 : E_0 \to E'$ is an a-isogeny, then $\operatorname{diag}(\alpha_0) : E_0^g \to X_0 := E'^g$ is also an a-isogeny. So on our product of elliptic curves, we can always compose or precompose with smooth isogenies, see Section 6.2.

To increase the parameters susceptible to this attack, we can also postcompose and precompose $\phi_B : E_0^g \to E_B^g$ by isogenies β_1, β_2. Write $N_A = bN_B + a, a, b > 0$, eventually applying the parameter tweaks of Section 6. Note that since N_A is coprime to N_B, then dividing by $\gcd(N_A, a, b)$ if necessary, we may assume that N_A, a, b are coprime. Write $b = b_1 b_2$, and suppose that we can find an explicit b_1-isogeny $\beta_1 : E_0^g \to Y_0$, a b_2-isogeny $\beta_2 : E_B^g \to Y_B$, and a a-isogeny $\alpha_0 : E_0^g \to X_0$. Let $\gamma = \beta_2 \circ \phi_B \circ \widetilde{\beta_1} : Y_0 \to Y_B$, it is a bN_B-isogeny. Consider the following pushouts,

$$\begin{tikzcd}
Y_0 \arrow[d, "\alpha_0'"] & E_0^g \arrow[l, "\beta_1"'] \arrow[r, "\phi_B"] \arrow[d, "\alpha_0"] & E_B^g \arrow[r, "\beta_2"] \arrow[d, "\alpha_B"] & Y_B \arrow[d, "\alpha_B'"] \\
Z_0 & X_0 \arrow[l, "\beta_1'"'] \arrow[r, "\phi_B'"] & X_B \arrow[r, "\beta_2'"] & Z_B
\end{tikzcd}$$

since a is prime to bN_B, $\gamma' = \beta_2' \circ \phi_B' \circ \widetilde{\beta_1'} : Z_0 \to Z_B$ is a $N_B b$-isogeny and α', α'' are a-isogenies.

We thus have the following isogeny diamond

$$\begin{tikzcd}
Z_0 \arrow[r, "\widetilde{\alpha'_0}"] \arrow[d, "\gamma'"] & Y_0 \arrow[d, "\gamma"] \\
Z_B \arrow[r, "\widetilde{\alpha'_B}"] & Y_B
\end{tikzcd}$$

so by Lemma 3.4, $F = \begin{pmatrix} \widetilde{\alpha'_0} & \tilde{\gamma} \\ -\gamma' & \alpha'_B \end{pmatrix}$ is a N_A-isogeny $F : Z_0 \times Y_B \to Y_0 \times Z_B$. In particular, $\operatorname{Ker} F$ is the image of $\tilde{F}$ on $(Y_0 \times Z_B)[N_A]$. Since a is prime to bN_b, it is also the image of $\tilde{F}$ on $Y_0 \times 0$: $\operatorname{Ker} F = \{(\alpha'_0(P), \gamma(P)) \mid P \in Y_0\}$. Note that as before, this means that we don't need to construct Z_B explicitly, however in this case we need to construct the pushout Z_0.

This allows to compute F as a smooth N_A-isogeny of dimension $2g$ in time $O(\log^2 N_A + \log N_A \ell_A^{2g})$ or even $\widetilde{O}(\log N_A \ell_A^{2g})$ by [LR22], hence evaluate $\gamma = \beta_2 \circ \phi_B \circ \widetilde{\beta_1}$ on any point of Y_0. It remains to recover ϕ_B from γ. Applying $\widetilde{\beta_2}$ and β_1, we can always recover $b\phi_B$, hence we may recover ϕ_B whenever b is prime to N_B. Otherwise, we at least recover a $N_B/\gcd(b, N_B)$-isogeny through which ϕ_B factors, and we iterate, which is possible as long as $\gcd(b, N_B) < N_B$.

We leave to the reader the case where α is constructed from E_B. Note that, using discrete logarithms if needed, we only need to evaluate $\alpha_0, \beta_1, \beta_2$ on a basis of the N_A-torsion of their respective domain. It is thus better to build the isogenies from E_0^g rather than from E_B^g, these evaluations can then be seen as a precomputation (involving the parameters and E_0).

In summary we have reduced recovering ϕ_B to evaluating the isogeny F in dimension $2g$:

Theorem 3.5. *In the situation of Theorem 1.1, suppose that we can find $a, b > 0$ such that $N_A = bN_B + a$ (eventually tweaking the parameters N_A, N_B), with a, b, N_a coprime, $b = b_1 b_2$, and a b_1-isogeny $\beta_1 : E_0^g \to Y_0$, a b_2-isogeny $\beta_2 : E_B^g \to Y_B$, and a a-isogeny $\alpha_0 : E_0^g \to X_0$. Assume furthermore for simplicity that* $\gcd(b, N_B) = 1$ *(or is small). Let T be a bound on the arithmetic operations required to evaluate β_1, β_2 and the pushout α' of α and β_1 on a basis of the N_A-torsion of E_0^g, E_B^g, Y_0 respectively. (By the discussion above, for α' and β_1, this can be seen as a precomputation depending on E_0).Then, we can recover generators of* $\operatorname{Ker} \phi_B$ *in $O(\ell_A^{2g} \log N_A + \log^2 N_A + T)$ arithmetic operations in $\mathbb{F}_q$, or even in $\widetilde{O}(\ell_A^{2g} \log N_A + T)$ via the fast isogeny decomposition of [DJP14, § 4.2.2].*

Remark 3.6. In dimension 8, the domain (and codomain) of F is a product of supersingular elliptic curves, so is a superspecial abelian variety. The same is true for the isogeny F in dimension $2g$ by the argument below. Since F is an N_A-isogeny with N_A prime to the characteristic of the base field, F, or its decomposition into a product of ℓ-isogenies, preserve the a-number of the intermediate abelian varieties. Hence they have a-number equal to $2g$, so they are still superspecial. By a theorem due to Deligne, Ogus and Shioda [Shi79, Theorem 3.5], they are all isomorphic (without the polarisation!) to E_0^{2g}. So in the decomposition of F we always stay on the same abelian variety E_0^{2g}, except that we gradually change its polarisation. For instance in the dimension 2 attack, we start with a product polarisation but the intermediate polarisations will generically be indecomposable, hence correspond to Jacobians of genus 2 hyperelliptic superspecial curves.

4. Dimension 4 attack

In dimension 2, we can always write an a-endomorphism on E_0^2 whenever $a = a_1^2 + a_2^2$. So using Section 3, we can do a dimension 4 attack whenever we can find $a, b > 0$ such that

$N_A = bN_B + a$ and both a and b are a sum of two squares. To increase our probability of success, we can also tweaks the parameters N_A and N_B as explained in Section 6.

Remark 4.1. Since we can always prolong α and β by isogenies of smooth degree using Section 6.2, we can consider the more general decompositions: $N_A = (b_1^2 + b_2^2)eN_B + (a_1^2 + a_2^2)e$ with e, f sufficiently smooth. But smooth integers are of negligible density compared to sum of two squares, so for simplicity we focus only in this case here.

Write $a = a_1^2 + a_2^2$, $b = b_1^2 + b_2^2$. Note that unlike the decomposition of a as a sum of four squares from Section 2, these decompositions into a sum of two squares requires the factorisation of a, b.

Write $\alpha = \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix}$, $\beta = \begin{pmatrix} b_1 & -b_2 \\ b_2 & b_1 \end{pmatrix}$. These matrices can be interpreted as endomorphisms of E_0^2 or E_B^2 and commute with ϕ_B Id: $\beta_B \phi_B \operatorname{Id} = \phi_B \operatorname{Id} \beta_0$, $\alpha_B \phi_B \operatorname{Id} = \phi_B \operatorname{Id} \alpha_0$. Furthermore, $\tilde{\alpha}\alpha = (a_1^2 + a_2^2) \operatorname{Id}$, so α is an a-endomorphism, and similarly β is a b-endomorphism:

$$\begin{tikzcd}
E_0^2 \arrow[r, "\phi_B \beta"] \arrow[d, "\alpha_0"] & E_B^2 \arrow[d, "\alpha_B"] \\
E_0^2 \arrow[r, "\phi_B \beta"] & E_B^2
\end{tikzcd}$$

Lemma 3.4, or a direct computation, shows that $F = \begin{pmatrix} \alpha_0 & \widetilde{\phi_B \operatorname{Id}} \widetilde{\beta_B} \\ -\beta_B \phi_B \operatorname{Id} & \widetilde{\alpha_B} \end{pmatrix}$ is a $N_A = a + bN_B$-endomorphism of $E_0^2 \times E_B^2$. Its kernel is given by $\operatorname{Ker} F = \{(\widetilde{\alpha_0}(P), \beta_B \phi_B \operatorname{Id}(P)) \mid P \in E_0^2[N_A]\}$.

We can thus evaluate F, hence evaluate $\beta_B \phi_B \operatorname{Id} = \phi_B \operatorname{Id} \beta_0$ on any point in $E_0^2(\mathbb{F}_q)$ in $O(\log^2 N_A + \log N_A \ell_A^4)$ arithmetic operations over $\mathbb{F}_q$ by [LR22]. In this situation we can recover more than just $b\phi_B$. Indeed from the matrix $\beta_B \phi_B \operatorname{Id}$ we can directly recover $b_1 \phi_B$ and $b_2 \phi_B$; so if $b' = \gcd(b_1, b_2)$, we can recover $b' \phi_B$ in $O(\log b)$ arithmetic operations on E_B. This means that we can recover the kernel of a $N_B / \gcd(N_B, b')$-isogeny $E_0 \to E'_B$ through which ϕ_B factors. If $\gcd(N_B, b') = 1$ we have directly recovered ϕ_B, otherwise we iterate the process, which is possible as long as $\gcd(N_B, b') < N_B$.

Remark 4.2 (Sum of two squares)**.** To decompose a number b as a sum of two squares $b = b_1^2 + b_2^2$ is the same as finding a factorisation $b = (b_1 + ib_2)(b_1 - ib_2) = \beta\overline{\beta}$ in the Gaussian integers $\mathbb{Z}[i]$. The order $\mathbb{Z}[i] \subset \mathbb{Q}(i)$ is of discriminant -4, so it is the maximal order, and it is euclidean by [Gau32], hence is principal. The prime $(2) = ((1+i)(1-i)) = \big((1+i)^2\big)$ is ramified, and the other integer primes are unramified. By the quadratic reciprocity law [Gau01], when p is an odd prime, -1 is a square modulo p if and only if $p \equiv 1 \pmod 4$. Hence when $p \equiv 1 \pmod 4$ it splits in $\mathbb{Z}[i]$, otherwise when $p \equiv 3 \pmod 4$ it stays inert. In particular, p is a sum of two squares if and only if $p = 2$ or $p \equiv 1 \pmod 4$ [Ste25, p.622; Fer40; DD94, Supplement XI].

We deduce that b is a sum of two squares if and only if all odd primes $p \equiv 3 \pmod 4$ dividing b have even exponent $v_p(b)$. Also, $\gcd(b_1, b_2) \mid \gcd(\beta, \overline{\beta}) \mid 2\gcd(b_1, b_2)$. Therefore, if $b = b_1^2 + b_2^2$, $\gcd(b_1, b_2) = 2^{\lfloor v_2(b)/2 \rfloor} \times \prod_{p \mid b, p \equiv 3 \pmod 4} p^{v_p(b)/2}$. In particular, b admits a primitive representation as a sum of two squares if and only if the odd prime divisors of b are all congruent to 1 modulo 4 and $4 \nmid b$. More generally, if the odd prime divisors of $\gcd(b, N_B)$ are congruent to 1 modulo 4, and either $2 \nmid N_B$ or $4 \nmid b$, we can find a decomposition $b = b_1^2 + b_2^2$ such that $\gcd(b_1, b_2, N_B) = 1$.

In Section 5, we will need decompositions of the form $b = b_1^2 + 4b_2^2$. Such a decomposition exists if $\beta \in \mathbb{Z}[2i]$, which is a suborder of $\mathbb{Z}[i]$ of index 2. So b admits such a decomposition if and only if it can be written a sum of two squares and $v_2(b)$ is even.

Furthermore, the number of integers less than x that can be written as a sum of two squares is given by the asymptotic behaviour of the L function $L(s) = (1 - \frac{1}{2^s})^{-1} \prod_{p \equiv 1 \pmod 4} (1 - \frac{1}{p^s})^{-1} \prod_{p \equiv 3 \pmod 4} (1 - \frac{1}{p^{2s}})^{-1}$ at $s = 1$. By Perron's formula, it is asymptotically equivalent to $C/\sqrt{\log x}$ [LeV12, Volume 2, p. 260–263], where $C \approx 0.7642$ is the Landau-Ramanujan constant. Adapting the proof, the same asymptotic bound holds for the number of integers that are a primitive sum of two squares (resp. of the form $b_1^2 + 4b_2^2$) via the L function $L(s) = (1 + \frac{1}{2^s}) \prod_{p \equiv 1 \pmod 4} (1 - \frac{1}{p^s})^{-1}$ (resp. $L(s) = (1 - \frac{1}{2^{2s}})^{-1} \prod_{p \equiv 1 \pmod 4} (1 - \frac{1}{p^s})^{-1} \prod_{p \equiv 3 \pmod 4} (1 - \frac{1}{p^{2s}})^{-1}$), except with a different constant $C \approx 0.49$ (resp. $C \approx 0.51$).

Summing up this discussion, we get for the dimension 4 attack:

Theorem 4.3. *In the situation of Theorem 1.1, suppose that we can find $a, b > 0$ such that $N_A = bN_B + a$ (eventually tweaking the parameters N_A, N_B) with N_A, a, b coprime and a, b can be written as a sum of two squares: $a = a_1^2 + a_2^2$, $b = b_1^2 + b_2^2$. Assume furthermore for simplicity that $\gcd(b, N_B)$ has its odd prime divisors congruent to 1 modulo 4, and if $2 \mid \gcd(b, N_B)$ then $4 \nmid b$.*

Then, given the decomposition of a and b as a sum of two square (eg given their factorisation), we can recover generators for $\operatorname{Ker} \phi_B$ in classical deterministic time $O(\ell_A^4 \log \ell_A \log N_A + \log^2 N_A)$ arithmetic operations in $\mathbb{F}_q$, or even $\widetilde{O}(\log N_A \ell_A^4)$ with the fast variant of smooth isogeny computation.

As mentioned in Remark 4.1 and Section 6, we can more generally look at $N_A = e(b_1^2 + b_2^2)N_B + f(a_1^2 + a_2^2)$ with e, f sufficiently smooth.

4.1. **Parameters selection.** In order to find parameters such that we may apply Theorem 4.3, a first idea is the following. We search, using Section 6, parameters a, b such that $eN_A = bN_B/D + a$, where e is an integer, D is some divisor of N_B (that we will want as small as possible), and a, b sum of two (primitive) squares. Since $N_A > N_B$, there are $O(eD)$ possible choices for b, among whose $\Omega(eD/\sqrt{\log eD})$ will be a primitive sum of two squares by Remark 4.2. We thus have $\Omega(eD/\sqrt{\log eD})$ candidates for a. If we make the *heuristic assumption* that these a behave like a random integer between 0 and N_A, the probability to find a a that is a sum of two squares is $\Omega(1/\sqrt{\log N_A})$ by the same Remark. Hence we need to take $eD = \widetilde{O}(\sqrt{\log N_A})$. There are $O(D)$ candidate D-isogenies through which ϕ_B may factorize, and we need to apply Theorem 4.3 to each of these candidates. Likewise, there are $O(e^3)$ possibilities to guess the image of ϕ_B on the $N_A e$-torsion (and this does not even take into account the cost of finding the eN_A-torsion which possibly lives in an extension of $\mathbb{F}_q$). Thus it appears that for the tweaking of parameters, it is preferable to use $e = 1$, $D = \widetilde{O}(\sqrt{\log N_A})$. So these parameter tweaks will lose a factor $O(D)$ in the final arithmetic complexity of the attack.

However, for the dimension 4 attack, we will see that by using Section 6.4 we can actually set $e = N_A$ without extra cost (asymptotically).

The question remains of the cost of the precomputation of the parameters a, b. We can directly iterate through sum of two squares for b, but checking if a is a sum of two squares requires its factorisation. Here we can use a trick from [Wes22a]: we restrict to the case a

a prime congruent to 1 modulo 4. This only requires a primality test, hence is much less expensive. However the probability that a is a prime (congruent to 1 modulo 4) will only be (heuristically) $\Omega(1/\log N_A)$, so this strategy will require larger parameters eD. Luckily, for the dimension 4 attack we can take $e = N_A$ as we have seen, which is more than large enough.

Reframing the above discussion, we need the following heuristic:

Heuristic 4.4.

- *Let $N_1 > N_2$ be two coprime integers, with N_2 sufficiently large and $N_1/N_2 > \Omega(\sqrt{\log N_1})$. Then if b is uniform amongst the numbers $x < N_1/N_2$ that are sum of two squares (resp. a sum of two primitive squares, resp. of the form $u^2 + 4v^2$), the probability that $a = N_1 - bN_2$ is a sum of two squares (resp. a sum of two primitive squares, resp. of the form $u^2 + 4v^2$) is $\Omega(1/\sqrt{\log N_1})$.*
- *Let $N_1 > N_2$ be two coprime integers, with N_2 sufficiently large and $N_1/N_2 > \Omega(\log N_1)$. Then if b is uniform amongst the numbers $x < N_1/N_2$ that are sum of two squares (resp. a sum of two primitive squares, resp. of the form $u^2 + 4v^2$), the probability that $a = N_1 - bN_2$ is* prime *and a sum of two squares is $\Omega(1/\log N_1)$.*

Motivation. The motivation behind this heuristic is that the a we get will behave like a uniform integer between 1 and N_1. The density of sum of two squares (resp. a sum of two primitive squares, resp. of the form $u^2 + 4v^2$) less than N_1 is equivalent asymptotically to $C/\sqrt{\log N_1}$, where C depends on the exact form we want. Likewise, the density of primes congruent to 1 less than N_1 is equivalent asymptotically to $C/\log N_1$ by the prime number theorem [Had96; Val96] and Dirichlet's theorem on arithmetic progressions [Dir37]. □

This heuristic allows us to derive the following complexity cost of the precomputation step.

Corollary 4.5. *Let $N_1 > N_2$ be two coprime integers, with N_2 sufficiently large. Then for $\epsilon > 0$, there is a constant C_ϵ such that under Heuristic 4.4, if $N_1/N_2 > C_\epsilon \log^{1/2} N_1$, we can find with probability $> 1 - \varepsilon$ a decomposition $N_1 = bN_2 + a$ where a, b are sum of two squares (resp. a sum of two primitive squares, resp. of the form $u^2 + 4v^2$). This decomposition requires in average $O(\sqrt{\log N_1})$ factorisation calls and $O(\log^{2.5} N_A)$ binary operations.*

If $N_1/N_2 > C_\epsilon \log N_1$, still under Heuristic 4.4 we can find such a decomposition in average $O(\log N_1)$ tests of primality. It will cost on average $O(\log^3 N_1)$ binary operations.

Proof. By Heuristic 4.4, we need to sample $\Omega(\log^{1/2} N_1)$ b of the form $b_1^2 + b_2^2$ to find an a which is also a sum of two squares, or $\Omega(\log N_1)$ if we also want a prime. The same also holds for the other decomposition, only the constant in the Ω changes.

We first look at the complexity analysis of the second case. Testing the primality of a via the Miller-Rabin pseudo-primality test [Mil76; Rab80] costs $O(\log^2 a)$, and we have the same average complexity to find an integer z such that $z^2 = -1 \pmod a$ (this is more or less equivalent to the Miller-Rabin pseudo-primality test). From z and a, a continued fraction expansion allows to decompose a as a sum of two squares [Her48], so given z, the decomposition $a = a_1^2 + a_2^2$ can be done in time $O(\log^2 a)$ by Euclide's algorithm [Εὐκ00] (it is well known that the complexity can be improved to $\widetilde{O}(\log a)$, see eg [BCG+17, § 6.3]) for a total complexity of $O(\log^2 a)$ on average to test the primality of a and write it as a sum of two squares.

For the first case, we need to factorize a to see if it can be written as sum of two squares. Given the prime factors of a, we can use the method above to find the decomposition of a into irreducible factors in the Gaussian integers $\mathbb{Z}[i]$,so we can also decompose a as a sum of two squares in time $O(\log^2 a)$. □

Proposition 4.6. *Under Heuristic 4.4, the precomputation step of the dimension* 4 *attack takes average time* $O(\log^3 N_A)$ *binary operations to find a decomposition* $N_A^2 = (b_1^2 + b_2^2)N_B + a_1^2 + a_2^2$. *Once this decomposition is found, the dimension* 4 *attack can be done in* $\widetilde{O}(\log N_A \ell_A^4)$ *arithmetic operations.*

Proof. By Heuristic 4.4, we can find $e \mid N_A$ such that $eN_A = (b_1^2 + b_2^2)N_B + (a_1^2 + a_2^2)$ which b_1, b_2 coprime. This precomputation costs $\widetilde{O}(\log^3 N_A)$ by Corollary 4.5. We can now construct a eN_A-endomorphism $F : X \to X$ where $X = E_0^2 \times E_B^2$ as in Theorem 4.3. We only know its action on $X[N_A]$, but by considering $\tilde{F}$, we can explicitly decompose F as $F = F_2 \circ F_1$ where F_1 is a N_A-isogeny and F_2 a e-isogeny, see Section 6.4. This decomposition costs $\widetilde{O}(\log N_A + \log e\ell_A^4)$ to compute (more precisely: to recover the domain of F_2 and its kernel), and evaluating F via this decomposition costs $\widetilde{O}(\log N_A \ell_A^4)$. □

5. Dimension 2 attack

We briefly describe how the dimension 2 attacks, due to [CD22; MM22], fit into the general framework of Section 3.

Write $N_A = bN_B + a$, to apply Section 3 for $g = 1$, we need to construct a a-isogeny $\alpha = \alpha_0 : E_0 \to X_0$ and a b-isogeny $\beta : E_0 \to Y_0$ (or $\beta : E_B \to Y_B$) to get the push-out square:

$$\begin{tikzcd}
Y_0 \arrow[d, "\alpha'_0"] & E_0 \arrow[l, "\beta"'] \arrow[r, "\phi_B"] \arrow[d, "\alpha_0"] & E_B \arrow[d, "\alpha_B"] \\
Z_0 & X_0 \arrow[l, "\beta'"] \arrow[r, "\phi'_B"] & X_B
\end{tikzcd}$$

The corresponding isogeny diamond

$$\begin{tikzcd}
Z_0 \arrow[r, "\widetilde{\alpha'_0}"] \arrow[d, "\phi'_B\circ\widetilde{\beta'}"] & Y_0 \arrow[d, "\phi_B\circ\widetilde{\beta}"] \\
X_B \arrow[r, "\widetilde{\alpha_B}"] & E_B
\end{tikzcd}$$

shows that $F = \begin{pmatrix} \widetilde{\alpha'_0} & \beta \circ \widetilde{\phi_B} \\ -\phi'_B \circ \widetilde{\beta'} & \alpha_B \end{pmatrix}$ is a N_A-isogeny $F : Z_0 \times E_B \to Y_0 \times X_B$ by Lemma 3.4.

If we don't assume that $\mathrm{End}(E_0)$ is known, we can only construct a a-endomorphism whenever a is square: if $a = a_1^2$ we take the a-endomorphism $[a_1]$. More generally, since it is also easy to construct isogenies of smooth degree starting from E_0 or E_B (see Section 6.2), the framework of Section 3 shows that the attack applies whenever $N_A = b_1^2 eN_B + a_1^2 f$ where e, f are sufficiently smooth. This is essentially the attack of [MM22]; in the first version they only looked at $N_A - N_B$ smooth (and tweaking of parameters), but to get a subexponential complexity they needed to look at the more general $N_A = eN_B + f$ case, which was already considered in [CD22] (squares are of negligible density compared to smooth numbers, so we can forget about them).

As mentioned in Section 1.5, in [CD22], the authors use the matrix F as an oracle attack, which requires many isogeny guesses, compared to the direct isogeny recovery of [MM22].

However, they also use the fact that for the parameters of SIKE submitted to NIST (or the Microsoft challenge [Cos21]), E_0 has a know endomorphism $\gamma = 2i$, so $\mathrm{End}(E_0) \supset \mathbb{Z}[2i]$. Hence we can construct an explicit a-endomorphism α on E_0 whenever $a = a_1^2 + 4a_2^2$, which is possible whenever all primes p such that $p \equiv 3 \mod 4$ or $p = 2$ are of even exponent in a by Remark 4.2. By Section 3, prolonging by isogenies of smooth degrees if necessary, for this starting curve E_0 the attack holds whenever $N_A = (b_1^2 + 4b_2^2)eN_B + (a_1^2 + 4a_2^2)f$. Otherwise, one needs to do some guesses, as in Section 6. In [CD22], the authors only look at $N_A = N_B + (a_1^2 + 4a_2^2)f$, but in [POP+22], Oudompheng, inspired by an earlier version of this paper describing the dimension 4 attack, implemented the more general formula above. This bumped down the time to solve the SIKEp217 challenge from 9 to 2 seconds and SIKEp964 instances from 1+h to 30 seconds.

The discussion of Section 4.1 shows:

Proposition 5.1. *Under Heuristic 4.4, when E_0 has known endomorphism $\gamma = 2i$, the dimension 2 attack has, after a precomputation step involving $O(\sqrt{\log N_A})$ factorisations and $O(1)$ calls to γ, complexity $\widetilde{O}(\log^{1.5} N_A \ell_A^2)$ arithmetic operations.*

Alternatively, we can dispense with factorisations in the precomputation step at the cost of increasing the complexity of the attack: still under Heuristic 4.4, after a precomputation step costing $O(\log^3 N_A)$ binary operations and $O(1)$ calls to γ, the dimension 2 attack has complexity $\widetilde{O}(\log^2 N_A \ell_A^2)$ arithmetic operations.

Proof. We proceed as in the proof of Proposition 4.6. In Corollary 4.5, we require a, b to decompose as $a = a_1^2 + 4a_2^2$ and $b = b_1^2 + 4b_2^2$. To find such a and b, we look for relations $N_A = bN_B/D + a$ where D is a divisor of N_B. When we look for a a sum of two squares in Corollary 4.5, we can take $D = \Theta(\sqrt{\log N_A})$, if we require furthermore that a is prime to decrease the precomputation cost, then we need $D = \Theta(\log N_A))$. We assume implicitly that it is possible to find a divisor of N_B of this magnitude.

Also, since the endomorphisms α and β are built from γ, the evaluation cost of these endomorphisms will depend on the cost of evaluating γ. But we only need to evaluate α, β on points of N_A-torsion, so we may consider that the computation of γ on a basis of $E_0[N_A]$ is a precomputation (depending on E_0). Evaluating α and β then takes $\widetilde{O}(\log N_A \ell_A^{1/2})$ by Lemma 3.3.When $E_0 = E_{\mathrm{NIST}}$, the evaluation of γ is done in $O(1)$, so evaluating α and β can be done directly in $O(\log N_A)$.

Once these precomputations are done, the evaluation of F takes time $\widetilde{O}(\log N_A \ell_A^2)$ arithmetic operations. We need to multiply this complexity by $O(D)$, the number of isogenies we need to guess. □

When $E_0 \neq E_{\mathrm{NIST}}$ has known endomorphisms, Castryck and Decru use [KLP+14; LB20] to build a path from E_{NIST} to E_0. This allows them to pushforward the a-isogeny α_{NIST} from E_{NIST} to an a-isogeny α on E_0 using the methods of [GPS17; DKL+20]. This time, evaluating α on rational points can only be done in polynomial time. But since the attack only needs the action of α on the N_A-torsion, it is sufficient to evaluate α on a basis of $E_0[N_A]$. This can be seen as a precomputation, which in this case involves not only the parameters N_A, N_B but also the starting curve E_0. The remaining evaluations on points of N_A-torsion can then be done in $\widetilde{O}(\log N_A \ell_A^{1/2})$ by Lemma 3.3.

Recall also from Section 1.5 that [Wes22b] gives a method to construct an a-isogeny in proven polynomial time on any supersingular elliptic curve with known endomorphism ring. This isogeny can also be evaluated in polynomial time. Applying this to $a = N_A - N_B$, computing this a-endomorphism α and its evaluation on a basis $E_0[N_A]$ can be seen as a precomputation, and then we have a direct isogeny recovery without parameter tweaks as in Section 2, except we only need to compute isogenies in dimension 2 rather than 8.

Proposition 5.2 (Wesolowski). *If* $\mathrm{End}(E_0)$ *is known, after a polynomial time precomputation to compute an* a*-isogeny* α *and its action on the* N_A*-torsion, the dimension* 2 *attack has complexity* $\widetilde{O}(\log N_A \ell_A^2)$ *arithmetic operations.*

Unfortunately, it is not clear what is the exact bound on the precomputation step of Wesolowski's approach.

Finally, we mention that for the isogeny computations in dimension 2, since any principally polarised surface is either a Jacobian or an elliptic curve, one can also use the Jacobian model of [CE14] (which can be extended to the case of product of elliptic curves), rather than the theta model of [LR22].

6. Parameter tweaks

We recall the decomposition of the parameters we need for the different attacks from the generic framework of Section 3:

- In dimension 8, or in dimension 2 when $\mathrm{End}(E_0)$ has known endomorphism ring (using [Wes22b]), no tweaks!
- In dimension 4, we need a decomposition $N_A = e(b_1^2 + b_2^2)N_B + f(a_1^2 + a_2^2)$, e, f sufficiently smooth. For the dimension 2 attack of [CD22] where $\mathrm{End}(E_0)$ has endomorphism $2i$, we need the very similar decomposition $N_A = (b_1^2 + 4b_2^2)eN_B + (a_1^2 + 4a_2^2)f$.
- For [MM22], in dimension 2 when $\mathrm{End}(E_0)$ is not known, we need $N_A = eN_B + f$ with e, f sufficiently smooth.

These decompositions rely on the fact that we can build isogenies of smooth degree on E_0 and E_B, we detail that complexity in Section 6.2.

We can furthermore tweak the parameters N_A and N_B as follow, as in the strategies of [CD22; MM22]. In the following, we assume that we are in the context of SIDH, so E_0, E_B are supersingular elliptic curves defined over $\mathbb{F}_q$ with $q = p^2$.

(1) We can replace N_A by $N'_A = N_A/d_A$ where d_A any divisor of N_A.
(2) We can replace N_B by N_B/d_B, where d_B is a small divisor of N_B. This requires guessing the first d_B-isogeny step of Φ_B, and we have $O(d_B)$ guesses.
(3) We can replace N_A by $N'_A = eN_A$ where e is a small integer prime to N_B. This means that we will build F a $N'_A = eN_A$ isogeny in dimension $2g$, where we only know its action on the N_A-torsion, and we want to recover F (eg its kernel). For a general e, we explain possible strategies in Section 6.3, strategies which can be much improved when $e \mid N_A$, see Section 6.4.

The rest of this section is devoted to determine the complexity of these tweaks.

6.1. Constructing a basis of the e-torsion of E.

We look at the complexity of building a basis of the e-torsion on E. By the group structure theorem of supersingular elliptic curves, since $\pi_{q^k} = (-p)^k$, $E(\mathbb{F}_{q^k}) \simeq \mathbb{Z}/((-p)^k - 1) \oplus \mathbb{Z}/((-p)^k - 1)$. Hence the smallest extension of $\mathbb{F}_q$ where the points of e torsion of E live is of degree k, the order of $-p$ modulo e, so $k = O(e)$. Sampling a e basis of E can be done by constructing the field $\mathbb{F}_{q^k}$, sampling random points in $E(\mathbb{F}_{q^k})$, multiplying by the cofactor $\frac{(-p)^k-1}{e}$ and then checking if we have a basis using the Weil pairing. The construction of $\mathbb{F}_{q^k}$ costs $\widetilde{O}(k^2 \log q + k \log^2 q)$ using

[Sho94] or $\widetilde{O}(k \log^5 q)$ using [CL13]. The dominant cost will be the sampling phase, which costs $O(k \log q)$ arithmetic operations in $\mathbb{F}_{q^k}$. In total we get $\widetilde{O}(k^2 \log^2 q) = O(e^2 \log^2 q)$ operations.

6.2. **Building a smooth isogeny on a supersingular elliptic curve $E/\mathbb{F}_{p^2}$.** We want to build a smooth isogeny of degree e. We can build it as a composition of $O(\log e)$ ℓ-isogenies, for primes $\ell \mid e$. If $\ell \mid N_A N_B$, since we have access to a rational N_A and N_B torsion basis, we can simply use it to sample an element of order f in time $O(\min(\log N_A, \log N_B))$ arithmetic operations, and the isogeny can then be computed in time $\widetilde{O}(\sqrt{\ell})$ arithmetic operations using `Velusqrt`[BDL+20].

We now detail the general case. Since $\pi_q = [-p]$, all cyclic kernels of order ℓ of E are rational, and their generators live in an extension of degree at most $k = O(\ell)$, the order of $-p$ modulo ℓ. We can construct $\mathbb{F}_{q^k}$ then sample a generator (any primitive point P of ℓ-torsion) in $O(k^2 \log^2 q)$ operations like in Section 6.1, then compute the isogeny using Vélu's formula [Vél71] or the `Velusqrt` algorithm [BDL+20] in time $O(\ell k \log q)$ (resp. $\widetilde{O}(\ell^{1/2} k \log q)$) for a total cost of $\widetilde{O}(k^2 \log^2 q + \ell^{1/2} k \log q) = \widetilde{O}(\ell^2 \log^2 q)$.

An alternative is to compute and factorize the ℓ-division polynomial ψ_ℓ. It is of degree $O(\ell^2)$ and can be computed in time $\widetilde{O}(\ell^2 \log q)$ via the recurrence formula. Furthermore, all points of ℓ-torsion live in the same extension of degree k. If ℓ is odd and $P \in E[\ell]$, x_P will live in the same extension as P unless k is even, in which case $\pi_q^{k/2} P = -P$ so x_P lives in an extension of degree $k/2$. This shows that the factors of ψ_ℓ are all of the same degree k if k is odd or $k/2$ if k is even. We can then skip the distinct degree factorisation phase, hence compute a factorisation of ψ_ℓ in time $\widetilde{O}(\ell^2 \log^2 q)$ by [VS92]. Any factor Q of ψ_f then gives us a construction of $\mathbb{F}_{q^k}$ and of a point of ℓ-torsion P in $E(\mathbb{F}_{q^k})$ via, if $E : y^2 = h(x)$, $P = \big(x \mod Q(x), y \mod (y^2 - h(x), Q(x))\big)$. Note that the polynomial $y^2 - h(x)$ splits in $\mathbb{F}_q[x]/Q(x)$ if $\deg Q = k$, otherwise it is irreducible, $\deg Q = k/2$ and it allows to construct $\mathbb{F}_{q^k}$ as a degree 2 tower over $\mathbb{F}_{q^{k/2}} = \mathbb{F}_q[x]/Q(x)$. We can then apply Vélu or `Velusqrt` to P as above, for a total cost of $\widetilde{O}(\ell^2 \log^2 q)$.

A third method is to construct an ℓ-isogeny using the ℓ-modular polynomial ϕ_ℓ (and its derivative), as in the SEA algorithm [Sch95]. We can evaluate this modular polynomial in time $\widetilde{O}(\ell^2 \log q)$ by an easy adaptation of [Kie20] (see [Rob21, Remark 5.3.9; Rob22c]), then recover a root in time $\widetilde{O}(\ell \log^2 q)$. Recovering the isogeny can then be done in quasi-linear time by solving a differential equation [BMS+08; Rob21, § 4.7.1]. This reduces the complexity to $\widetilde{O}(\ell^2 \log q + \ell \log^2 q)$ operations.

6.3. **Recovering a $N_A e$-isogeny from its action on the N_A-torsion.** We have a $N_A e$-isogeny F in dimension $2g$, that Eve built from the secret isogeny $\phi_B : E_0 \to E_B$ and some auxiliary isogeny she controls. She wants to recover F in order to retrieve ϕ_B from it.

One way to do that is to guess the action of ϕ_B on the eN_A-torsion of E_0. This requires to compute a basis of the eN_A-torsion on E_0, as described in Section 6.1, possibly taking an extension of degree k, and then guessing the images of Φ_B on the $N_A e$ torsion. Note that since the N_A-torsion is rational by assumption, we have $k = O(e)$. Guessing the image of ϕ_B on this basis involves $O(e^3)$-tries, using the compatibility of ϕ_B with the Weil pairing and the known image of the N_A-torsion.

An alternative strategy, when the codomain Y of $F : X \to Y$ is known, is as follow: since F is an $N'_A = eN_A$-isogeny, and we know the action of ϕ_B on the N_A-torsion, we can still recover $\operatorname{Ker} F \cap X[N_A]$. So taking a maximal isotropic subgroup of $\operatorname{Ker} F \cap X[N_A]$ for the Weil pairing e_{N_A} (for the F we build in Section 3, this intersection is already maximal isotropic), we can thus recover F_1 in a decomposition $F = F_2 \circ F_1$, with F_1 an N_A-isogeny and F_2 a e-isogeny. Then we can try to bruteforce F_2 by an e-isogeny search in dimension $2g$.

6.4. **Recovering a N_A^2-isogeny from its action on the N_A-torsion.** When $F : X \to Y$ is an $N_A e$-isogeny with $e \mid N_A$, and the action of F on $X[N_A]$ is known, then by using the dual $\tilde{F}$ there is a much better strategy to recover F than in Section 6.3. This is the same strategy used in [QKL+21] when F is an endomorphism of elliptic curves. We assume here for simplicity that $\operatorname{Ker} F$ is of rank $2g$, which is the case for our applications: the F constructed in Section 3 has this property. So $K = \operatorname{Ker} F$ admits a symplectic complement K': $X[eN_A] = K \oplus K'$, and $\operatorname{Ker} \tilde{F} = F(X[eN_A]) = F(K')$. Decompose $F = F_2 \circ F_1$, $F_1 : X \to X_1$, $F_2 : X_1 \to Y$, with $\operatorname{Ker} F_1 = \operatorname{Ker} F \cap X[N_A] = K[N_A]$.Then we have $\operatorname{Ker} \widetilde{F_2} = \operatorname{Im} F_2 \mid X_1[e] = \operatorname{Im} F \mid X[e] = \operatorname{Ker} \tilde{F} \cap Y[e] = F(K')[e] = F(K'[e])$ (indeed $\operatorname{Im} F \mid X[e] \subset \operatorname{Im} F_2 \mid X_1[e]$ but they have the same cardinality e^{2g} since the kernel is of rank $2g$, so we have equality). So we can build F_1 from X through its kernel $\operatorname{Ker} F \cap X[N_A]$ (which is maximal isotropic of rank $2g$ in $X[N_A]$), build $\widetilde{F_2}$ from Y through its kernel $\operatorname{Im} F \mid X[e]$, then compute $\operatorname{Ker} F_2 = \operatorname{Im} \widetilde{F_2} \mid Y[e]$ to recover F_2, hence $F = F_2 \circ F_1$.

In particular this strategy applies for the attacks in dimension 4 of Section 4 and in dimension 8 of Section 2.

Let us detail this case: in these examples, the endomorphism F of $E_0^g \times E_B^g$ is always of the form $F = \begin{pmatrix} \alpha_0 & \tilde{\beta}\widetilde{\phi_B}\,\mathrm{Id} \\ -\phi_B\beta & \widetilde{\alpha_B} \end{pmatrix}$with α_0 an a-endomorphism of E_0^g, β a b-endomorphism of E_0^g, and α_B the a-endomorphism of E_B^g making the diagram commute:

$$\begin{tikzcd}
E_0^g \arrow[r, "\phi_B\beta"] \arrow[d, "\alpha_0"] & E_B^g \arrow[d, "\alpha_B"] \\
E_0^g \arrow[r, "\phi_B\beta"] & E_B^g
\end{tikzcd}$$

We also have a, b, N_A coprime to each other. In particular, $\operatorname{Ker} F = \{(\widetilde{\alpha_0}(P), (\phi_B\beta)(P)) \mid P \in E_0^g[eN_A]\}$, and $\operatorname{Ker} \tilde{F} = \{(\alpha_0(P), (-\phi_B\beta)(P)) \mid P \in E_0^g[eN_A]\}$ are of rank g. We decompose $F = F_2 \circ F_1$, where $\operatorname{Ker} F_1 = \operatorname{Ker} F[N_A] = \{(\widetilde{\alpha_0}(P), (\phi_B\beta)(P)) \mid P \in E_0^g[N_A]\}$, and $\operatorname{Ker} \widetilde{F_2} = \operatorname{Ker} \tilde{F}[e] = \{(\alpha_0(P), (-\phi_B\beta)(P)) \mid P \in E_0^g[e]\}$. Since we know the image of ϕ_B on a basis of $E_0[N_A]$, we know the image of ϕ_B on a basis of $E_0[e]$ via $O(\log(N_A/e))$ arithmetic operations. So we can recover the image of $\phi_B\beta$ on this basis in $\widetilde{O}(\log N_A \ell_A^{1/2})$ and $O(1)$ evaluations of β by Lemma 3.3. We also need in $O(1)$ calls to α_0.

In these examples, the endomorphisms β and α_0 can be evaluated in time $O(\log N_A)$, so the kernel of F_1 and of $\widetilde{F_2}$ can be computed in time $\widetilde{O}(\log N_A \ell_A^{1/2})$. A linear complement of $\operatorname{Ker} \widetilde{F_2}$ is given by $0 \times E_B^g[e]$. Indeed it is of rank g and cardinal q^{2g}, and if $x = (0, Q) \in \operatorname{Ker} \widetilde{F_2}$, $Q = -\phi_B\beta(P)$ for a $P \in E_0^g[e]$ such that $\alpha_0 P = 0$. But this implies $aP = 0$, hence $P = 0$ since a is prime to $e \mid N_A$, so $Q = 0$. So $\operatorname{Ker} F_2 = \widetilde{F_2}(0 \times E_B^g[e])$, can be recovered in $2g$ calls to the evaluation of the e-isogeny $\widetilde{F_2}$.

The total cost to recover the domain of F_2 and a basis of its kernel is thus $\widetilde{O}(\log N_A \ell_A^{1/2} + \log e \ell_e^{2g}) = \widetilde{O}(\log N_A \ell_A^{2g})$.

Unfortunately, this strategy does not work for the dimension 2 attack of Section 5, because (with the notations of this Section), X_B is constructed as a pushout, and we only obtain it when we compute the codomain of F. But this means that if F is an N_A^2-isogeny, there is no easy way to obtain $\operatorname{Ker} \tilde{F}[N_A]$, hence split F as a product of two N_A-isogenies, without first computing F fully.

7. Conclusion

By Theorem 1.1 and Remark 1.2, we have a new toolbox for recovering an N_B-isogeny $f : A \to B$ given its action on the N_A-torsion as long as $N_A^2 \geq N_B$ and N_A is sufficiently smooth. This toolbox allows to break SIDH efficiently in all cases. Can it also be used to build new isogeny based cryptosystems?

References

[BDL+20] D. Bernstein, L. De Feo, A. Leroux, and B. Smith. "Faster computation of isogenies of large prime degree". In: *Algorithmic Number Theory Symposium*. 2020. arXiv: 2003.10118.

[BL04] C. Birkenhake and H. Lange. *Complex abelian varieties*. Second. Vol. 302. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Berlin: Springer-Verlag, 2004, pp. xii+635. ISBN: 3-540-20488-1.

[BCR10] G. Bisson, R. Cosset, and D. Robert. *AVIsogenies*. Magma package devoted to the computation of isogenies between abelian varieties. 2010. URL: https://www.math.u-bordeaux.fr/~damienrobert/avisogenies/. Free software (LGPLv2+), registered to APP (reference IDDN.FR.001.440011.000.R.P.2010.-000.10000). Latest version 0.7, released on 2021-03-13.

[BMS+08] A. Bostan, F. Morain, B. Salvy, and E. Schost. "Fast algorithms for computing isogenies between elliptic curves". In: *Mathematics of Computation* 77.263 (2008), pp. 1755–1778.

[BCG+17] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and É. Schost. *Algorithmes efficaces en calcul formel*. Published by the authors, 2017.

[CD22] W. Castryck and T. Decru. *An efficient key recovery attack on SIDH (preliminary version)*. Cryptology ePrint Archive, Paper 2022/975. 2022. URL: https://eprint.iacr.org/2022/975.

[CLM+18] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 395–427.

[Cos21] C. Costello. "The case for SIKE: a decade of the supersingular isogeny problem". In: *Cryptology ePrint Archive* (2021).

[CLN16] C. Costello, P. Longa, and M. Naehrig. "Efficient algorithms for supersingular isogeny Diffie-Hellman". In: *Advances in Cryptology*. Springer. 2016. URL: https://ecc2017.cs.ru.nl/slides/ecc2017-costello.pdf.

[CE14] J.-M. Couveignes and T. Ezome. "Computing functions on Jacobians and their quotients". In: *LMS Journal of Computation and Mathematics* 18.1 (2014), pp. 555–577. arXiv: 1409.0481.

[CL13] J.-M. Couveignes and R. Lercier. "Fast construction of irreducible polynomials over finite fields". In: *Israel Journal of Mathematics* 194.1 (2013), pp. 77–105.

[DDF+21] L. De Feo, C. Delpech de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, and B. Wesolowski. "Séta: Supersingular encryption from torsion attacks". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2021, pp. 249–278.

[DJP14] L. De Feo, D. Jao, and J. Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247.

[DKL+20] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. "SQISign: compact post-quantum signatures from quaternions and isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2020, pp. 64–93.

[Dir37] J. P. G. L. Dirichlet. "Beweis eines Satzes über die arithmetische Progression". In: *Bericht üuber die Verhandlungen der königlich Presussischen Akademie der Wissenschaften Berlin* (1837).

[DD94] J. P. G. L. Dirichlet and R. Dedekind. *Vorlesungen über Zahlentheorie*. 1894.

[Fer40] P. de Fermat. *Correspondence to Mersenne*. Dec. 25, 1640.

[FKM+22] T. B. Fouotsa, P. Kutas, S.-P. Merz, and Y. B. Ti. "On the isogeny problem with torsion point information". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2022, pp. 142–161.

[GPS17] S. D. Galbraith, C. Petit, and J. Silva. "Identification protocols and signature schemes based on supersingular isogeny problems". In: *International conference on the theory and application of cryptology and information security*. Springer. 2017, pp. 3–33.

[Gau01] C. F. Gauss. *Disquisitiones arithmeticae*. 1801.

[Gau32] C. F. Gauss. *Theoria residuorum biquadraticorum. Commentatio secunda*. Typis Dieterichchianis, 1832.

[Had96] J. Hadamard. "Sur la distribution des zéros de la fonction ζ(s) et ses conséquences arithmétiques". In: *Bulletin de la Société Mathématique de France* (1896).

[Ham44] W. R. Hamilton. "On Quaternions; or on a new System of Imaginaries in Algebra". In: *Philosophical Magazine* 25.3 (1844), pp. 489–495.

[Her48] C. Hermite. "Note au sujet de l'article precedent". In: *Journal de Mathématiques Pures et Appliquées* (1848), p. 15.

[JAC+17] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalili, B. Koziel, B. LaMacchia, P. Longa, et al. *SIKE: Supersingular isogeny key encapsulation*. 2017. URL: https://sike.org/.

[JD11] D. Jao and L. De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *International Workshop on Post-Quantum Cryptography*. Springer. 2011, pp. 19–34.

[Kan97] E. Kani. "The number of curves of genus two with elliptic differentials." In: *Journal für die reine und angewandte Mathematik* 485 (1997), pp. 93–122.

[Kan16] E. Kani. "The moduli spaces of Jacobians isomorphic to a product of two elliptic curves". In: *Collectanea mathematica* 67.1 (2016), pp. 21–54.

[Kie20] J. Kieffer. "Evaluating modular polynomials in genus 2". 2020. arXiv: 2010.10094 [math.NT]. HAL: hal-02971326.

[KLP+14] D. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. "On the quaternion-isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 418–432.

[Lag70] J. L. de Lagrange. "Démonstration d'un théoreme d'arithmétique". In: *Nouv. Mém. Acad. Roy. Sc. de Berlin* (1770), pp. 123–133.

[LeV12] W. J. LeVeque. *Topics in Number Theory, volumes I and II*. Courier Corporation, 2012.

[LB20] J. Love and D. Boneh. "Supersingular curves with small noninteger endomorphisms". In: *Open Book Series* 4.1 (2020), pp. 7–22.

[LR10] D. Lubicz and D. Robert. "Efficient pairing computation with theta functions". In: ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Comput. Sci. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. Springer–Verlag, July 2010. DOI: 10.1007/978-3-642-14518-6_21. URL: http://www.normalesup.org/~robert/pro/publications/articles/pairings.pdf. Slides: 2010-07-ANTS-Nancy.pdf (30min, International Algorithmic Number Theory Symposium (ANTS-IX), July 2010, Nancy), HAL: hal-00528944.

[LR15] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". In: *Journal of Symbolic Computation* 67 (Mar. 2015), pp. 68–92. DOI: 10.1016/j.jsc.2014.08.001. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192.

[LR22] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". Accepted for publication at ANTS XV Conference — Proceedings. Aug. 2022. URL: http://www.normalesup.org/~robert/pro/publications/articles/change_level.pdf.

[MM22] L. Maino and C. Martindale. *An attack on SIDH with arbitrary starting curve*. Cryptology ePrint Archive, Paper 2022/1026. 2022. URL: https://eprint.iacr.org/2022/1026.

[Mil76] G. L. Miller. "Riemann's hypothesis and tests for primality". In: *Journal of computer and system sciences* 13.3 (1976), pp. 300–317.

[MGE12] B. Moonen, G. van der Geer, and B. Edixhoven. *Abelian varieties*. Book project, 2012. URL: https://www.math.ru.nl/~bmoonen/research.html#bookabvar.

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354.

[Mum70] D. Mumford. *Abelian varieties*. Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970, pp. viii+242.

[Oud22] R. Oudompheng. "A note on implementing direct isogeny determination in the Castryck-Decru SIKE attack". Aug. 2022.

[Pet17] C. Petit. "Faster algorithms for isogeny problems using torsion point images". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2017, pp. 330–353.

[PH78] S. Pohlig and M. Hellman. "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (Corresp.)" In: *IEEE Transactions on information Theory* 24.1 (1978), pp. 106–110.

[POP+22] G. Pope, R. Oudompheng, L. Panny, et al. *Castryck-Decru Key Recovery Attack on SIDH*. Aug. 2022. URL: https://github.com/jack4818/Castryck-Decru-SageMath.

[QKL+21] V. d. Quehen, P. Kutas, C. Leonardi, C. Martindale, L. Panny, C. Petit, and K. E. Stange. "Improved torsion-point attacks on SIDH variants". In: *Annual International Cryptology Conference*. Springer. 2021, pp. 432–470.

[Rab80] M. O. Rabin. "Probabilistic algorithm for testing primality". In: *Journal of number theory* 12.1 (1980), pp. 128–138.

[RS86] M. O. Rabin and J. O. Shallit. “Randomized algorithms in number theory”. In: *Communications on Pure and Applied Mathematics* 39.S1 (1986), S239–S256.

[Rob21] D. Robert. “Efficient algorithms for abelian varieties and their moduli spaces”. HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux).

[Rob22a] D. Robert. “Breaking SIDH in polynomial time”. Aug. 2022. URL: http://www.normalesup.org/~robert/pro/publications/articles/breaking_sidh.pdf. eprint: 2022/1038.

[Rob22b] D. Robert. “Evaluating isogenies in polylogarithmic time”. Aug. 2022. URL: http://www.normalesup.org/~robert/pro/publications/articles/polylog_isogenies.pdf. eprint: 2022/1068.

[Rob22c] D. Robert. “Fast evaluation of modular polynomials and compact representation of isogenies between elliptic curves”. Aug. 2022. In preparation.

[Sch95] R. Schoof. “Counting points on elliptic curves over finite fields”. In: *J. Théor. Nombres Bordeaux* 7.1 (1995), pp. 219–254.

[Shi79] T. Shioda. “Supersingular K3 surfaces”. In: *Algebraic geometry*. Springer, 1979, pp. 564–591.

[Sho94] V. Shoup. “Fast construction of irreducible polynomials over finite fields”. In: *Journal of Symbolic Computation* 17.5 (1994), pp. 371–391.

[Sho09] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.

[Som21] A. Somoza. *thetAV*. Sage package devoted to the computation with abelian varieties with theta functions, rewrite of the AVIsogenies magma package. 2021. URL: https://gitlab.inria.fr/roberdam/avisogenies/-/tree/sage.

[Ste25] S. Stevin. *l'Arithmétique de Simon Stevin de Bruges*. annotated by Albert Girard. Leyde, 1625.

[Val96] C.-J. de la Vallée Poussin. “Recherches analytiques sur la théorie des nombres premiers”. In: *Annales de la Société scientifique de Bruxelle* (1896).

[Vél71] J. Vélu. “Isogénies entre courbes elliptiques”. In: *Compte Rendu Académie Sciences Paris Série A-B* 273 (1971), A238–A241.

[VS92] J. Von Zur Gathen and V. Shoup. “Computing Frobenius maps and factoring polynomials”. In: *Computational complexity* 2.3 (1992), pp. 187–224.

[Wes22a] B. Wesolowski. “The supersingular isogeny path and endomorphism ring problems are equivalent”. In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 1100–1111.

[Wes22b] B. Wesolowski. “Understanding and improving the Castryck-Decru attack on SIDH”. Aug. 2022.

[Zar74] J. G. Zarhin. “A remark on endomorphisms of abelian varieties over function fields of finite characteristic”. In: *Mathematics of the USSR-Izvestiya* 8.3 (1974), p. 477.

[Διό50] ὁ. Ἀ. Διόφαντος. *Ἀριθμητικά*. ≈250.

[Εὐκ00] Εὐκλείδης. *Στοιχεία*. ≈-300.

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE