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

We extend the recent attacks by [CD22; MM22] and prove that there exists a proven deterministic polynomial time attack on SIDH [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 class of parameter 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.

It is also possible to break a large class of parameters N_A, N_B by going to dimension 4 rather than 8, see Section 4. Namely, this is possible whenever we can write $N_A = bN_B + a$ with $a, b > 0$ sum of two squares (along with some slight technical conditions). This is a much more likely condition than smoothness of $N_A - N_B$, hence (if possible tweaking the parameters N_A and N_B) we expect this attack to be highly likely and more efficient than the dimension 8 attack in practice.

The idea of the dimension 8 attack is that we can always write a, b as a sum of four squares, hence we always get an attack in dimension 8.

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).

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.*

Date: August 25, 2022.

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 we can recover ϕ_B in classical deterministic time $O(\ell_A^8 \log \ell_A \log N_A + \log^2 N_A + \log^2 N_B)$ arithmetic operations in $\mathbb{F}_q$ where ℓ_A is the largest prime divisor of N_A.*

Note that in in the context of SIDH, if $N_B > N_A$ we will simply try to recover Alice's secret isogeny Φ_A instead.

2. Dimension 8 attack

Since $N_A > N_B$, write $N_A = N_B + a$ for a positive integer $a > 0$. 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 $M^T M = a \operatorname{Id}$. 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 $\mathbb{Z}[i, j, k]$. 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 $\tilde{\alpha}_0 \alpha_0 = a \operatorname{Id}$, hence α_0 is an a-isogeny. 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 $\phi_B : E_0^4 \to E_B^4$ 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: $\phi_B \alpha_0 = \alpha_B \phi_B$:

$$\begin{array}{ccc} E_0^g & \xrightarrow{\phi_B} & E_B^g \\ \downarrow{\scriptstyle \alpha_0} & & \downarrow{\scriptstyle \alpha_B} \\ E_0^g & \xrightarrow{\phi_B} & E_B^g \end{array}$$

Remark 2.1. 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 $O(\log^2 a)$ by [RS86].

Let $F = \begin{pmatrix} \alpha_0 & \widetilde{\phi_B} \\ -\phi_B & \widetilde{\alpha_B} \end{pmatrix}$, 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 $\tilde{F} = \begin{pmatrix} \widetilde{\alpha_0} & -\widetilde{\phi_B} \\ \phi_B & \alpha_B \end{pmatrix}$, we compute

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

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

[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 $\mathbb{F}_{p^2}$.

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

We can compute the action of F on the N_A-torsion. Indeed, since N_A is prime to N_B, we know how $\widetilde{\phi_B}$ acts on $E_B[N_A]$: 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 $\widetilde{\phi_B}(Q_1) = N_B P_1$, $\widetilde{\phi_B}(Q_2) = N_B P_2$. From the action of F on $X[N_A]$ it is easy to compute its kernel using some linear algebra and discrete logarithms, either directly in X or, via the Weil pairing, in $\mu_{N_A}(\overline{\mathbb{F}}_q)$. These discrete logarithms are inexpensive because N_A is assumed to be smooth.

But in fact we can directly recover 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 $\operatorname{Ker} F = \{\widetilde{\alpha_0}(P), \phi_B(P) \mid P \in E_0^4[N_A]\}$. 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 $O(\ell^8 \log \ell)$ 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 $O(\log^2 N_A + \log N_A \ell_A^8 \log \ell_A)$.

Remark 2.2. The isogenies computations in [LR22; BCR10; Som21] use a (level $n = 4$ or $n = 2$) 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 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 $n = 4$, the theta model uses 2^{16} coordinates, so we would need in practice to switch to the *Kummer* model by working in level $n = 2$ 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 $n = 4$, or 2^4 coordinates in level $n = 2$.

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 $(0, 0, 0, 0, Q_i, 0, 0, 0)$, 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 $O(\omega(N_B) \log N_B) = O(\log^2 N_B)$ operations in $E_0(\mathbb{F}_q)$ along with two calls to the evaluation of F. This concludes the complexity analysis of Theorem 1.1.

Remark 2.3.

- 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 only difference is that this time we get Ker ϕ_B as the image of $\widetilde{\phi}_B$ on a $2g$-dimensional basis of $E_B[N_B]$. To extract a g dimensional basis of the kernel from these images, 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). Hence, since the dimension g is fixed, this still costs $O(\omega(N_B)\log N_B) = O(\log^2 N_B)$.

- When $\ell_A = O(1)$, we can use a SIDH style fast evaluation of the N_A-isogeny F as in [DJP14, § 4.2.2]. If $\omega(N_B) = O(1)$ also (for instance if $\ell_B = O(1)$), the attack becomes quasi-linear: $\widetilde{O}(\log N_A)$, 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.

Recall that an N-isogeny $f : (A,\lambda_A) \to (B,\lambda_B)$ is an isogeny such that $f^*\lambda_B := \tilde{f}\circ\lambda_B\circ f = N\lambda_A$, where $\tilde{f} : \widetilde{B} \to \widetilde{A}$ is the dual isogeny. Letting $\tilde{f} = \lambda_A^{-1}\tilde{f}\lambda_B$ be the dual isogeny $\tilde{f} : B \to A$ 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 $K = \operatorname{Ker} f$ 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 $O(N^g \log N)$ arithmetic operations over the base field.

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_1d_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:

$$\begin{array}{ccc} A & \xrightarrow{f_1} & A_1 \\ \downarrow{\scriptstyle f_2} & & \downarrow{\scriptstyle f_1'} \\ A_2 & \xrightarrow{f_2'} & B \end{array}$$

Lemma 3.1 (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 $\operatorname{Ker} F = \{(\widetilde{f_1}(P), f_2'(P)) \mid P in A_1[d]\}$.

Proof. We check 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 3.1 where $A = E_0^g, B = E_B^g$ and F is actually an endomorphism.

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. Suppose that we can find an explicit b-isogeny $\beta : E_B^g \to X_B$, and a a-isogeny $\alpha : E_0^g \to X_0$. Let $\gamma = \beta \circ \phi_B : E_0^g \to X_B$, it is a bN_B-isogeny. Then we ca build the following pushouts,

$$\begin{array}{ccccc} E_0^g & \xrightarrow{\phi_B} & E_B^g & \xrightarrow{\beta} & X_B \\ \downarrow{\scriptstyle\alpha} & & \downarrow{\scriptstyle\alpha''} & & \downarrow{\scriptstyle\alpha'} \\ X_0 & \xrightarrow{\phi'_B} & Y & \xrightarrow{\beta'} & X \end{array}$$

and since a is prime to bN_B, $\gamma' = \beta'\phi'_B : X_0 \to X$ is a $N_B b$-isogeny and α' a a-isogeny.

We thus have the following isogeny diamond

$$\begin{array}{ccc} X_0 & \xrightarrow{\tilde{\alpha}} & E_0^g \\ \downarrow{\scriptstyle\gamma'} & & \downarrow{\scriptstyle\gamma} \\ X & \xrightarrow{\widetilde{\alpha'}} & X_B \end{array}$$

so by Lemma 3.1 $F = \begin{pmatrix} \tilde{\alpha} & \tilde{\gamma} \\ -\gamma' & \alpha' \end{pmatrix}$ is a N_A-isogeny $F : X_0 \times X_B \to E_0^g \times X$. In particular, $\operatorname{Ker} F$ is the image of $\tilde{F}$ on $(E_0^g \times X)[N_A]$. Since a is prime to bN_b, it is also the image of $\tilde{F}$ on $E_0^g[N_A] \times 0$: $\operatorname{Ker} F = \{(\alpha(P), \gamma(P)) \mid P \in E_0^g[N_A]\}$. Note that this means that we don't need to construct X explicitly, we only need to know it exists, and will recover it when we evaluate F.

This allows to compute F as a smooth N_A-isogeny of dimension $2g$ in time $O(\log^2 N_A + \log N_A \ell_A^{2g} \log \ell_A)$ by [LR22], hence evaluate $\gamma = \beta \circ \phi_B$ on any point of E_0^g. It remains to recover ϕ_B from γ. Applying $\tilde{\beta}$ 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 $\gcd(b, N_B) < N_B$.

A variant is to construct $\beta : E_0^g \to X_B$, and to form the pushout squares:

$$\begin{array}{ccccc} X_B & \xleftarrow[\beta]{} & E_0^g & \xrightarrow{\phi_B} & E_B^g \\ \downarrow{\scriptstyle\alpha'} & & \downarrow{\scriptstyle\alpha} & & \downarrow{\scriptstyle\alpha''} \\ Y & \xleftarrow[\beta']{} & X_0 & \xrightarrow{\phi'_B} & X \end{array}$$

We then have the following isogeny diamond

$$\begin{array}{ccc} Y & \xrightarrow{\widetilde{\alpha'}} & X_B \\ \downarrow{\scriptstyle\phi'_B \circ \widetilde{\beta'}} & & \downarrow{\scriptstyle\phi_B \circ \tilde{\beta}} \\ X & \xrightarrow{\widetilde{\alpha''}} & E_B^g \end{array}$$

from which we construct the N_A-isogeny $F : Y \times E_B^g \to X_B \times X$

$$F = \begin{pmatrix} \widetilde{\alpha'} & \beta \circ \widetilde{\phi_B} \\ -\phi'_B \circ \widetilde{\beta'} & \alpha'' \end{pmatrix},$$

where $\operatorname{Ker} F = \{(\alpha'(P), \phi_B \circ \tilde{\beta}(P)) \mid P \in X_B[N_A]$. Here we need to construct the pushout Y first, but not X, it will be recovered when we compute F. And like in the previous case, from $\phi_B \circ \tilde{\beta}(P)$ we can always recover $b\phi_B$, which is enough to recover a partial information on ϕ_B as long as $\gcd(b, N_B) < N_B$.

We leave to the reader the case where α, β are constructed from E_B.

4. Dimension 4 attack

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 tweaks the parameters N_A and N_B as explained in Section 6.

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 or E_B and commute with ϕ_B. Furthermore, $\tilde{\alpha}\alpha = (a_1^2 + a_2^2)\,\mathrm{Id}$, so α is an a-endomorphism, and similarly β is a b-endomorphism. Lemma 3.1 or a direct computation shows that $F = \begin{pmatrix} \alpha_0 & \widetilde{\phi_B}\widetilde{\beta_B} \\ -\beta_B\phi_B & \widetilde{\alpha_B} \end{pmatrix}$ is a $N_A = a + bN_B$-endomorphism of $E_0^2 \times E_B^2$.

We can thus evaluate F, hence evaluate $\beta_B\phi_B = \phi_B\beta_0$ on any point in $E_0^2(\mathbb{F}_q)$ in $O(\log^2 N_A + \log N_A \ell_A^4 \log \ell_A)$ arithmetic operations over $\mathbb{F}_q$ by [LR22]. In this situation we can recover more than just $b\phi_B$. Indeed from the matrix $\beta_B\phi_B$ 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 4.1. It is well known that 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 $4 \nmid b$. In particular, if the odd prime divisors of $\gcd(b, N_B)$ are congruent to 1 modulo 4, and if $2 \mid N_B$ then $4 \nmid b$, we can find a decomposition $b = b_1^2 + b_2^2$ such that $\gcd(b_1, b_2, N_B) = 1$.

Furthermore, by Perron's formula, the number of integers less than x that can be written as a sum of two squares (resp. a sum of two primitive squares) is roughly $0.7642x/\sqrt{\log x}$ where 0.7642 is an approximation of the Landau-Ramanujan constant (resp. $\approx 0.49x/\sqrt{\log x}$).

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

Theorem 4.2. *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) 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 factorisation of a and b, we can recover ϕ_B in classical deterministic time $O(\ell_A^4 \log \ell_A \log N_A + \log^2 N_A + \log^2 N_B)$ arithmetic operations in $\mathbb{F}_q$.

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 $\alpha : E_0 \to X_0$ and a b-isogeny $\beta : E_B \to X_B$. 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), the framework of Section 3 shows that the attack applies whenever $N_A = b_1^2 e N_B + a_1^2 f$ where e, f are sufficiently smooth. This is essentially the attack of [MM22], except they only look at $N_A - N_B$ smooth (and tweaking of parameters).

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]. The fact that we could directly recover ϕ_B from F was also noticed independently at least by Oudompheng [Oud22], Petit, and Wesolowski [Wes22].

However, in [CD22], the authors also use the fact that for the parameters of SIKE submitted to NIST (or the Microsoft challenge [Cos21]), E_0 has a know endomorphism $\gamma = 2i$, hence $\mathrm{End}(E_0) \supset \mathbb{Z}[2i]$. 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. Hence, 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 [PO+22] Oudompheng, inspired by an earlier version of this paper describing the dimension 4 attack, implemented the more general formula above.

Finally we mention that [Wes22] gives a method to construct an a-isogeny on any supersingular elliptic curve with known endomorphism ring. Applying this to $a = N_A - N_B$, computing this a-endomorphism 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.

6. Parameter tweaks

We can tweak the parameters N_A and N_B as follow, as in the strategies of [CD22; MM22] (upon which we improve a bit). 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_A by $N'_A = eN_A$ where e is a small integer prime to N_B. This requires to compute a basis of the eN_A-torsion on E, possibly taking an extension, and then guessing the images of Φ_B on the $N_A e$ torsion. 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 eN_A torsion of E live is of degree k, the order of $-p$ modulo eN_A. Since the N_A-torsion is rational by assumption, we have $k = O(e)$. Sampling a eN_A basis of E_0, E_B can be done by sampling random points, multiplying by the cofactor $\frac{(-p)^k - 1}{eN_A}$ and then checking if we have a basis using the Weil pairing. This costs $O(k^2 \log^2 q) = O(e^2 \log^2 q)$ operations. 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.
(3) 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.
(4) We can replace N_B by $N'_B = fN_B$ where f is any smooth integer prime to N_A. This requires prolonging Φ_B by an f-isogeny. If $f \mid N_B$, we can simply use the existing N_B-torsion basis, hoping that we don't accidentally backtrack through Bob's isogeny. For the general case, since $\pi_q = [-p]$, all cyclic kernels of order f of E_B

are rational, and their generators live in an extension of degree at most $k = O(f)$. We can then sample a generator (any primitive point P of f-torsion) in $O(f^2 \log^2 q)$ operations like in Item 2, then compute the isogeny using Vélu's formula [Vél71] or the `Velusqrt` algorithm [BDL+20] in time $O(f^2 \log q)$ (resp. $\widetilde{O}(f^{3/2} \log q)$) for a total cost of $O(f^2 \log^2 q)$.

It is more expansive to compute and factorize the f-division polynomial ψ_f, since it is of degree $O(f^3)$. An alternative is to construct an f-isogeny using the f-modular polynomial ϕ_f (and its derivative), as in the SEA algorithm [Sch95]. We can evaluate this modular polynomial in time $\widetilde{O}(f^2 \log q)$ by an easy adaptation of [Kie20] (see [Rob21, Remark 5.3.9; Rob22]), then recover a root in time $\widetilde{O}(f \log^2 q)$. 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 $\widetilde{O}(f^2 \log q + f \log^2 q)$ operations.

References

[BDL+20] D. Bernstein, L. De Feo, A. Leroux, and B. Smith. “Faster computation of isogenies of large prime degree”. 2020. arXiv: 2003.10118.

[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.

[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.

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

[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.

[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/.

[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.

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

[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.

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

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

[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).

[Rob22] 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.

[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.

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

[Wes22] B. Wesolowski. "Understanding and improving the Castryck-Decru attack on SIDH". Aug. 2022.

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