Pushforward Problems and Applications to Isogeny-based Cryptography

Luciano Maino[1], Christophe Petit[1,2]

[1] University of Birmingham, Birmingham, UK
l.maino@bham.ac.uk
[2] Université Libre de Bruxelles, Brussels, Belgium
christophe.petit@ulb.be

Abstract. Let E and E' be two supersingular elliptic curves and let $\varphi : E \to E'$ be an isogeny of known degree d. Given a basis (P, Q) of $E[N]$ together with $(\varphi(P), \varphi(Q))$, it is possible to recover φ provided that N is sufficiently large and smooth, and that the torsion basis can be represented over a small extension of the base field.
In this work, we consider the more general setting where the N-torsion may not be efficiently representable. To address this setting, we introduce a new framework for encoding torsion information via an oracle that computes pushforwards of N-isogenies under φ. We then show that there exist instances for which access to the pushforward oracle allows for an efficient isogeny recovery.
Beyond their theoretical interest, these instances have direct cryptographic implications. We show a practical attack against the threshold signature scheme recently proposed by Kim, Kim, and Lee. We also identify weak instances for Basso's oblivious pseudorandom function, and we refine the security discussion for Leroux and Roméas's updatable encryption scheme.

Keywords: Isogenies, Pushforwards, Isogeny Problems

1 Introduction

The security of all isogeny-based cryptosystems relies on some variants of the following problem.

> **The General Isogeny Problem:**
> Given two supersingular elliptic curves E and E' defined over $\mathbb{F}_{p^2}$, find any isogeny $\varphi : E \to E'$.

In most of the constructions, significant additional information is provided. Sometimes, it is known that the curves E and E' are connected by an unknown isogeny $\varphi : E \to E'$ of known degree d. In other cases, some information about the images of some torsion points under φ is revealed.
A well-known weak variant of the general isogeny problem is the following.

This is an extended version of the paper accepted at Asiacrypt 2026. This version includes an appendix collecting some additional technical results.

Supersingular Isogeny with Torsion (SSI-T):
Let d and N be two integers coprime with a cryptographic prime p, two supersingular elliptic curves E and E' defined over $\mathbb{F}_{p^2}$ and connected by an unknown isogeny $\varphi : E \to E'$ of known degree d, and a basis (P, Q) of $E[N]$. Given $(E, \varphi(P), \varphi(Q))$, recover the isogeny φ.

Assuming that $4d + 1 \leq N^2$, early instances of the SSI-T were broken in a series of attacks [13,35,44] that targeted the SIDH cryptosystem [19,26]. It is worth noting that these attacks leverage the facts that N is a smooth integer and that the N-torsion is defined over $\mathbb{F}_{p^2}$.

Since the discovery of these attacks against SIDH, yet new variants of the SSI-T problem have appeared in the literature. The key insight to make SSI-T secure again is to mask the torsion information $(\varphi(P), \varphi(Q))$ by only revealing

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \cdot (\varphi(P), \varphi(Q)) = ([a]\varphi(P) + [b]\varphi(Q), [c]\varphi(P) + [d]\varphi(Q)),$$

for some special matrices $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$. Cryptosystems whose security is based on variants of the SSI-T include (Q)FESTA [8,38], M-SIDH [25], LIT-SiGamal [37], and POKÉ/PIKE [7,11].

In [18], De Feo, Fouotsa and Panny investigate the hardness of some of these variants. Despite acknowledging that masked torsion information can be represented in different forms, they assume that the N-torsion is always defined over $\mathbb{F}_{p^2}$, as this is the relevant case in most cryptographic applications. It is not hard to see that the arguments contained in [18] still hold in the case where the N-torsion is *representable*.

Definition 1. *Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ and let $N = \prod_{i=1}^{r} \ell_i^{e_i}$. We say that the N-torsion is representable if, for all $i = 1, \ldots, r$, the torsion group $E[\ell_i^{e_i}]$ can be represented over a field extension of $\mathbb{F}_{p^2}$ of degree polynomial in $\log(p)$. Also, we will say that a point $P \in E$ is representable if there exists an N' such that $P \in E[N']$ and the N'-torsion is representable.*

For instance, given a prime ℓ such that ℓ is polynomial in $\log(p)$, the ℓ-torsion is representable [24, Theorem 2].

1.1 Contributions

In this paper, we study the case where the provided torsion information is not representable. Our main contributions are as follows.

1. We introduce a formal framework for studying non-representable torsion information. We consider a secret isogeny $\varphi : E \to E'$ of degree d and assume that the N-torsion is not representable. Torsion information is encoded through *isogeny representations* of N-isogenies. Rather than revealing masked torsion points, we introduce the notion of a *pushforward oracle*: on input an N-isogeny $\sigma : E \to E^*$, the oracle outputs $[\varphi]_* \sigma : E' \to E'^*$, namely the pushforward of σ under φ. Using this formalism, we investigate several variants of the problem and highlight behaviours that differ significantly from the representable setting.

2. We show that there exist instances where, even if the torsion information is not representable, it is still possible to efficiently recover secret isogenies. The resulting techniques are reminiscent of the so-called *"lollipop attacks"* [42,43]. Our analysis substantially broadens the applicability of these attacks by removing the requirement that torsion information should be representable and smooth. Additionally, we explain how to create backdoor instances vulnerable to our techniques. The underlying strategy differs significantly from that of [43] and may be of independent interest.
3. We review existing constructions and identify cryptosystems that implicitly provide a pushforward oracle. In particular, we give a concrete attack against the threshold signature scheme introduced by Kim, Kim and Lee [28]. Additionally, we argue that Basso's oblivious pseudorandom function [2] admits backdoored instantiations, and we propose countermeasures to mitigate this issue. We also refine the security arguments for Leroux and Roméas's updatable encryption scheme [34], arguing that their design choices do not seem vulnerable to our techniques. Finally, we provide general recommendations for the design of cryptosystems exposing non-representable torsion information.

1.2 Technical overview

Let $\varphi : E \to E'$ be an isogeny of degree d. If the N-torsion is representable, the isogeny φ *reveals some torsion information* when we have the following data:

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \cdot (\varphi(P), \varphi(Q)) = ([a]\varphi(P) + [b]\varphi(Q), [c]\varphi(P) + [d]\varphi(Q)),$$

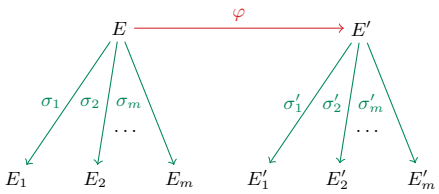
where (P, Q) is a basis of $E[N]$. Usually the masking matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ has a special form, *e.g.*, it is a scalar matrix $\left(\begin{smallmatrix} a & 0 \\ 0 & a \end{smallmatrix}\right)$, for some unknown $a \in \mathbb{Z}/N\mathbb{Z}$. A natural question to ask is what it means for φ to "reveal torsion information" when this torsion is no longer representable.

When there is no efficient explicit representation of an N-torsion basis, we can no longer use the above data to capture torsion information on the isogeny. However, we still have efficient representations of degree N isogenies, and those isogenies essentially encode torsion subgroups through their kernels.

Starting from this observation, we suggest to generally encode N-torsion information in terms of a *pushforward oracle* $\mathcal{O}^{\mathrm{PUSH}}_{\varphi,N}$: namely, on input an N-isogeny $\sigma : E \to E^*$, the oracle $\mathcal{O}^{\mathrm{PUSH}}_{\varphi,N}(\sigma)$ returns $\sigma' := [\varphi]_* \sigma : E' \to E'^*$, where the isogeny $[\varphi]_* \sigma$ is the isogeny with kernel $\langle \varphi(\ker \sigma) \rangle$. Pictorially, we have the following.

When the N-torsion is representable, this pushforward oracle can be instantiated with an algorithm using torsion point data as above, but our encoding of "torsion point information" also captures the more general setting where the torsion points are not necessarily representable.

As we will illustrate later, this encoding of non-representable torsion information is not merely a natural generalisation of the SSI-T with masked information, but it actually captures the information leakage in some existing isogeny cryptosystems. Therefore, it is fundamental to understand when such information can lead to recovering the isogeny φ.



We analyse the case where $\mathrm{End}(E)$ admits endomorphisms θ of norm $N^2 e$, for some small e, acting as a scalar multiplication on the d-torsion. In this case, we can write θ as $\widehat{\sigma_2} \circ \eta \circ \sigma_1$ where $\deg(\sigma_i) = N$ and $\deg(\eta) = e$. Since the endomorphism θ acts as the scalar multiplication on the d-torsion, we have that $[\varphi]_* \theta \in \mathrm{End}(E')$. In particular, access to $\mathcal{O}^{\mathrm{PUSH}}_{\varphi,N}$ allows us to reconstruct $[\varphi]_* \theta$; see Figure 1 for an intuitive description.



Fig. 1: The isogeny η' is represented via a zig-zag arrow because it is recovered via brute-forcing or meet-in-the-middle.

We show that if $d = \deg(\varphi)$ is smooth, then a single θ may suffice to recover φ. When $\deg(\varphi)$ is not smooth, we require two non-commuting θ_1 and θ_2. Interestingly, our attacks are reminiscent of the "lollipop attacks" on SIDH in the first case [42,43].

At first sight, having an elliptic curve E admitting such endomorphisms θ seems to be a restrictive condition, but we point out that such instances can be found in many protocols that have appeared in the literature. Indeed, we show a concrete attack against a recently proposed threshold signature construction [28] and we build weak instances for Basso's oblivious pseudorandom function [2]. We also refine the security discussion for Leroux and Roméas's updatable encryption scheme [34].

Roadmap. We begin by recalling some background knowledge in Section 2. In Section 3, we introduce a formal framework to capture the information coming from non-representable torsion. Then, in Section 4, we show that, in certain cases, it is possible to leverage the information coming from non-representable torsion. We explain how to construct such vulnerable instances in Section 5. These instances are not only of theoretical interest but also appear in some

isogeny-based cryptosystems. We discuss the impact of our findings in Section 6. Finally, we draw some conclusions in Section 7.

Acknowledgements. We thank the anonymous reviewers of Asiacrypt 2026 for identifying a mistake in the even analogue of Lemma 18 in an earlier version of this paper. Their feedback has helped us improve the quality and clarity of the manuscript.

Petit is partly supported by EPSRC through grant number EP/V011324/1 and FRS-FNRS through grant number T.0238.25. Part of this work was done while Maino was a postdoctoral fellow at FRS-FNRS and was supported by EPSRC through grant number EP/V011324/1.

2 Preliminaries

In this section, we recall some background knowledge. We begin by summarising some basic facts about elliptic curves and isogenies in Section 2.2 and the Deuring correspondence in Section 2.3. Then, we survey some previous work on similar attacks in Section 2.4.

2.1 Notations

When using $O(\cdot)$, we will refer to the standard "big-Oh" notation. Also, we will adopt $\mathsf{poly}(x)$ to denote a polynomial in x. We will use log to refer to the logarithm in base 2 and ln for the logarithm in base e.

Given two problems A and B, we use the notation $A \xRightarrow{\text{reduces to}} B$ to say that problem A reduces to B, *i.e.*, an algorithm to solve B can be used as a subroutine to (efficiently) solve A. In practice, we will always work with field extensions of $\mathbb{F}_p$. In our context, "efficiently" means that it runs in $O(\mathsf{poly}(\log(p)))$. Similarly, we will say that an algorithm runs in polynomial time if it runs in $O(\mathsf{poly}(\log(p)))$. An integer n is said to be smooth if all its prime factors are polynomial in $\log(p)$.

Given an isogeny $\varphi : E \to E'$, we use the notation $\ker(\varphi)[d] := \ker(\varphi) \cap E[d]$. Also, given two endomorphisms $\theta_1, \theta_2 \in \mathrm{End}(E)$, we will use the notation $\theta_{1|E[s]} = \theta_{2|E[s]}$ if there exists an endomorphism $\theta_3 \in \mathrm{End}(E)$ such that $\theta_1 = \theta_2 + [s]\theta_3$.

2.2 Elliptic curves and isogenies

We assume some familiarity with elliptic curves and their isogenies, and refer the reader to [46,17] for more details. In the rest of the paper, we will focus on supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ where $p = 3 \pmod 4$ for simplicity. This is the most common choice in cryptography.

In what follows, we will assume that isogenies are separable unless stated otherwise. Given two isogenies $\varphi : E_0 \to E_\varphi$ and $\psi : E_0 \to E_\psi$ of coprime degrees, the *pushforward of* ψ *under* φ is the isogeny $[\varphi]_*\,\psi : E_\varphi \to E'_\psi$ whose kernel is $\langle \varphi(\ker \psi) \rangle$. Conversely, given two isogenies $\varphi : E_0 \to E_\varphi$ and $\hat{\psi} : E_\varphi \to E'_\psi$ of

coprime degrees, the *pullback of* ψ *under* φ is the isogeny $[\varphi]^*\psi : E_0 \to E_\psi$ such that $\ker(\psi) = \langle \varphi(\ker([\varphi]^*\psi)) \rangle$.[3]

Isogenies can be represented in several different ways. Isogenies of smooth degrees can be efficiently represented as a chain of smaller degree isogenies, or equivalently by a kernel generator, if representable (see Definition 1). In our setting, we will have to deal with isogenies whose kernel generators are not representable. We therefore recall the notion of isogeny representation, simplifying the abstract notion given in [5, Definition 1].

Definition 2. *Let $\varphi : E \to E'$ be an isogeny. A representation of φ consists of the curves E and E' together with an algorithm that, on input a representable $P \in E$, outputs $\varphi(P)$ in $O(\mathsf{poly}(\log(p)))$.*

Following the attacks on SIDH [13,35,44], we have a representation of φ as soon as we have its image on some representable and smooth N-torsion, where $N^2 \geq 4\deg(\varphi) + 1$. We refer the reader to [45] for a survey on this topic. In particular, given a representation of φ, we also have a representation of $\widehat{\varphi}$. Additionally, the composition of isogeny representations is still a representation. Henceforth, when writing $\rho : E \to E'$ or simply ρ, we intend a representation of ρ.

2.3 The Deuring correspondence

The set $\mathrm{End}(E)$ of all endomorphisms of E is isomorphic to a maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$, the quaternion algebra ramified at p and ∞; more concretely $\mathcal{B}_{p,\infty} = \mathbb{Q}\langle i, j \rangle$ where $i^2 = -1$, $j^2 = -p$ and $ij = -ji$. When writing $\mathrm{End}(E)$, we will refer to a description of the endomorphism ring of E; this can be given either as representations of four endomorphisms of E generating $\mathrm{End}(E)$ as a lattice (see, for instance, [41]), or equivalently as an embedding of $\mathrm{End}(E)$ into $\mathcal{B}_{p,\infty}$ [23, Lemmata 4, 5]. We will say that a curve E has known endomorphism ring when a representation of $\mathrm{End}(E)$ is provided.

The Deuring correspondence [21] provides a correspondence between supersingular elliptic curves in characteristic p (up to isomorphism and Galois conjugacy) and maximal orders (up to conjugacy) of $\mathcal{B}_{p,\infty}$. In particular, isogenies $\varphi : E \to E'$ correspond to left $\mathrm{End}(E)$-ideals. We refer the reader to [31, §2] for a more modern treatment of the subject.

A useful fact that we will use in the rest of the paper is the following. Let E be a curve of *known* endomorphism ring, let $\theta \in \mathrm{End}(E)$ and consider N a positive integer. Independently of whether the N-torsion is representable, smooth or arbitrary, we can always compute the isogenies $\varphi_1 : E \to E_1$ and $\varphi_2 : E \to E_2$ whose kernels are given by $\ker(\theta)[N]$ and $\ker(\widehat{\theta})[N]$, respectively, in polynomial time [29,23,50].

[3] In what follows, we will also have to deal with pushforwards $[\varphi]_*\alpha$ of endomorphisms α of degree not coprime to $\deg(\varphi)$ that remain endomorphisms. In this case, we will extend the notion of pushforward as follows. We define $[\varphi]_*\alpha$ to be the endomorphism $\frac{\varphi \circ \alpha \circ \widehat{\varphi}}{[\deg(\varphi)]}$.

2.4 Lollipop attacks

The first attack against some instances of the SSI-T problem was designed by Petit in [42]. The key idea is as follows. Consider an isogeny $\varphi : E \to E'$ of degree d and let (P, Q) be a basis of the N-torsion, where the N-torsion is accessible and smooth. Assume that there exists $\theta \in \mathrm{End}(E)$ such that $\deg(\varphi \circ \theta \circ \widehat{\varphi} + [a]) = N$, for some integer a. Then, using $(\varphi(P), \varphi(Q))$, we can evaluate $\varphi \circ \theta \circ \widehat{\varphi} + [a]$ on the N-torsion and in turn retrieve a kernel generator of $\varphi \circ \theta \circ \widehat{\varphi} + [a]$, which gives a representation of $\varphi \circ \theta \circ \widehat{\varphi} + [a]$. From the representation of $\varphi \circ \theta \circ \widehat{\varphi} + [a]$, it is then possible to recover a representation of φ, assuming that the d-torsion is also smooth, representable and d does not have too many prime factors.

Petit's idea was subsequently generalised and improved in [43]. The condition $\deg(\varphi \circ \theta \circ \widehat{\varphi} + [a]) = N$ was then relaxed to $\deg(\varphi \circ \theta \circ \widehat{\varphi} + [a]) = N^2 e$, for some small and smooth e. These attacks were dubbed *lollipop attacks* because of the picture arising when drawing the endomorphism $\varphi \circ \theta \circ \widehat{\varphi}$. In the rest of the paper, we will use the same terminology when referring to this class of attacks.

The SSI-T problem was then completely broken by the aforementioned SIDH attacks. These attacks were further generalised by Castryck and Vercauteren [14]; however their generalisation does not cover the case of non-representable N-torsion.

The techniques we describe in Section 4 build upon the lollipop attacks. However, we apply a different strategy that allows us to exploit the information coming from the non-representable N-torsion.

As we will see in Section 6, non-representable N-torsion information was often (more or less explicitly) assumed to be useless by protocol designers, and protocols leaking such information were therefore considered to be safe. We contradict this assumption in some cases, including some instances that have appeared in the literature. Our generalisation also allows us to treat the case where $d = \deg(\varphi)$ is non-smooth, broadening the cases affected by our attacks.

3 Pushforward problems

In this section, we introduce a formalism to treat problems where the torsion information may or may not be representable. To do so, we first introduce the concept of a *pushforward oracle*. For a secret isogeny $\varphi : E_0 \to E$, we define $\mathcal{O}^{\mathrm{PUSH}}_{N,\varphi}$ to be the following oracle.

$\mathcal{O}^{\mathrm{PUSH}}_{N,\varphi}(\sigma)$:
1: **if** $\deg(\sigma) \neq N$ **then**
2: **return** $\perp$
3: **return** $[\varphi]_* \sigma$.

Equipped with this oracle, we can now define different problems where access to the pushforward oracle is guaranteed.

Problem 3 (Computational Isogeny Pushforward Problem). Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let d and N be two known positive integers

such that d, N and p are pairwise coprime. Assume we are given an elliptic curve E' such that there exists an unknown d-isogeny $\varphi : E \to E'$. Additionally, we are guaranteed access to the oracle $\mathcal{O}^{\text{PUSH}}_{N,\varphi}$. The *computational isogeny pushforward problem* requires recovering a representation of a d-isogeny $\widetilde{\varphi}$ such that $\widetilde{\varphi}_{|E[N]} = [a]\varphi_{|E[N]}$, for some $a \in (\mathbb{Z}/N\mathbb{Z})^*$.

Remark 4. We observe that the isogeny $\widetilde{\varphi}$ recovered in Problem 3 is unique if there is no endomorphism $\alpha \in \text{End}(E) \setminus \mathbb{Z}$ such that $\deg(\alpha) = d^2$ and $\alpha_{|E[N]} = [a]_{|E[N]}$ for some $a \in (\mathbb{Z}/N\mathbb{Z})^*$.

A necessary condition for this problem to be hard is that the isogeny φ must be long enough, *i.e.*, its degree d must be large enough. As customary in most cryptographic applications, given a security parameter λ, the degree d must be at least $2^{2\lambda}$, or $d > 2^\lambda$ if d is prime. In the rest of this paper, we will assume that this is the case, hence ruling out brute-forcing attacks [1].

In some applications, it might be interesting to look at weaker instances of this problem.

Problem 5 (One-more Isogeny Pushforward Problem). Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let d and N be two known positive integers such that d, N and p are pairwise coprime. Assume we are given an elliptic curve E' such that there exists an unknown d-isogeny $\varphi : E \to E'$, and we can query the oracle $\mathcal{O}^{\text{PUSH}}_{N,\varphi}$ on a set of isogenies $\mathcal{Q}$ of our choice. The *one-more isogeny pushforward problem* requires recovering the representation of $[\varphi]_* \psi$ for a cyclic N-isogeny $\psi : E \to E_\psi$ such that $\psi \neq \psi' \circ \iota$ for all $\psi' \in \mathcal{Q}$ and $\iota \in \text{Aut}(E)$.

Problem 6 (Decisional Isogeny Pushforward Problem). Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let d and N be two known positive integers such that d, N and p are pairwise coprime. Assume we are given an elliptic curve E and we can query the oracle $\mathcal{O}^{\text{PUSH}}_{N,\varphi}$ on a set of isogenies $\mathcal{Q}$ of our choice, where $\varphi : E \to E'$ is an unknown d-isogeny.

Additionally, we choose a cyclic N-isogeny $\psi : E \to E_\psi$, where $\psi \notin \mathcal{Q}$. Given a cyclic N-isogeny $\psi' : E' \to E_{\psi'}$, the *decisional isogeny pushforward problem* asks to recognise whether $\psi' = [\varphi]_* \psi$.[4]

It is clear that, assuming that we can compute N-isogenies originating from the starting curve E, it holds:

$$\text{Problem 6} \xRightarrow{\text{reduces to}} \text{Problem 5} \xRightarrow{\text{reduces to}} \text{Problem 3}$$

In general, to the best of our knowledge, we do not expect that Problem 5 $\xRightarrow{\text{reduces to}}$ Problem 6 and Problem 3 $\xRightarrow{\text{reduces to}}$ Problem 5. In Sections 3.2 and 3.3, we will show that, for some parameter choices, such reductions would have implied breaking some (believed-to-be) hard problems.

[4] We highlight the fact that, differently from Problem 5, we do not need to discard $\psi \neq \psi' \circ \iota$ for all $\psi' \in \mathcal{Q}$ and $\iota \in \text{Aut}(E)$. To the best of our knowledge, given the pushforward of any $\psi : E \to E_\psi$ under φ, it is not clear how to recognise whether an isogeny $\psi' : E' \to E_{\psi'}$ is equal to $[\varphi]_* (\psi \circ \iota)$, for some $\iota \in \text{Aut}(E)$.

3.1 Easy instances of Problem 3

When the N-torsion is representable and smooth, there are instances when we can solve Problem 3. For instance, if $N^2 \geq 4\deg(\varphi) + 1 =: 4d + 1$ and the number of prime factors of N is $O(\log\log(p))$, then we can use Robert's attack on SIDH [44] to solve Problem 3. We now explain how by following a similar argument as in the attack in [6] against one of the OPRFs in [9]. For the sake of simplicity, we assume that N is odd.

Let (P, Q) be a basis of $E[N]$; by our assumption, we have that the points P and Q are representable. Also, define the following three isogenies:

- $\sigma_1 : E \to E_{\sigma_1}$ such that $\ker \sigma_1 = \langle P \rangle$;
- $\sigma_2 : E \to E_{\sigma_2}$ such that $\ker \sigma_2 = \langle Q \rangle$;
- $\sigma_3 : E \to E_{\sigma_3}$ such that $\ker \sigma_3 = \langle P + Q \rangle$.

Querying the oracle $\mathcal{O}^{\text{PUSH}}_{N,\varphi}$ with these three isogenies, we obtain the following pushforwards:

- $\sigma'_1 := [\varphi]_* \sigma_1 : E' \to E'_{\sigma_1}$;
- $\sigma'_2 := [\varphi]_* \sigma_2 : E' \to E'_{\sigma_2}$;
- $\sigma'_3 := [\varphi]_* \sigma_3 : E' \to E'_{\sigma_3}$.

Given the representations of σ'_i's, we can recompute their kernels: $\ker(\sigma'_1) = \langle \varphi(P) \rangle$, $\ker(\sigma'_2) = \langle \varphi(Q) \rangle$ and $\ker(\sigma'_3) = \langle \varphi(P+Q) \rangle$. In other words, we have computed $[a_P]\varphi(P)$, $[a_Q]\varphi(Q)$ and $[a_{P+Q}]\varphi(P+Q)$, where $a_P, a_Q, a_{P+Q} \in (\mathbb{Z}/N\mathbb{Z})^*$.

Using the Weil pairing, we obtain the following system of equations.

$$\begin{cases} e_N([a_P]\varphi(P), [a_Q]\varphi(Q)) = e_N(P,Q)^{da_P a_Q} \\ e_N([a_{P+Q}]\varphi(P+Q), [a_Q]\varphi(Q)) = e_N(P,Q)^{da_{P+Q} a_Q} \\ e_N([a_{P+Q}]\varphi(P+Q), [a_P]\varphi(P)) = e_N(P,Q)^{-da_{P+Q} a_P} \end{cases} \tag{1}$$

From Equation (1), via discrete logarithm computations, which are efficient by the assumption that the N-torsion is representable and smooth, we recover a_P^2, a_Q^2 and a_{P+Q}^2 as follows. Define x_1, x_2, x_3 to be the discrete logarithms of $h_1 := e_N([a_P]\varphi(P), [a_Q]\varphi(Q))$, $h_2 := e_N([a_{P+Q}]\varphi(P+Q), [a_Q]\varphi(Q))$ and $h_3 := e_N([a_{P+Q}]\varphi(P+Q), [a_P]\varphi(P))$ in base $g := e_N(P, Q)$. Solving the system in the variables a_P, a_Q and a_{P+Q}, we get $a_P^2 = -x_1 x_3/(dx_2)$, $a_Q^2 = -x_2 x_1/(dx_3)$, $a_{P+Q}^2 = -x_2 x_3/(dx_1) \pmod N$.

Recall that modulo N, the number of square roots depends on the number of distinct prime factors of N. Since we assumed that such a number is $O(\log\log(p))$, we can indeed recover the correct values for a_P, a_Q and a_{P+Q}. Finally, since $N^2 \geq 4d + 1$ and N is smooth, we have a higher dimensional representation of φ following the argument in [44].

When the number of prime factors of N is polynomial in $\log(p)$, there are still weak instances of Problem 3. In [14, §4], Castryck and Vercauteren showed some weak instances of the problem by looking at the M-SIDH key exchange [25]. In particular, they prove that it is possible to break Problem 3 when either condition holds:

– E_0 is defined over $\mathbb{F}_p$ and $N > d$;
– we can compute an $\omega \in \mathrm{End}(E_0) \setminus \mathbb{Z}$ such that $N > d\sqrt{\deg(\omega)}$;
– we can compute an isogeny $\omega : E_0 \to E_0^{(p)}$ such that $N > d\sqrt{\deg(\omega)}$.

Remark 7. The example we have just shown proves that Problem 3 is equivalent to the M-SIDH problem when the N-torsion is smooth and representable. This is further evidence that Problem 3 correctly generalises the SSI-T problem when the N-torsion is not representable.

3.2 Easy instances of Problem 5, but hard Problem 3

Let $\varphi : E \to E'$ be an isogeny of degree d, and suppose we can query the oracle $\mathcal{O}^{\mathrm{PUSH}}_{N,\varphi}$. We will show an instance where Problem 5 can be easily solved but Problem 3 is believed to be hard.

This example is based on M-SIDH [25]. Let E be a curve of unknown endomorphism ring and such that there exists no isogeny $\omega : E \to E^{(p)}$ such that $N > d\sqrt{\deg(\omega)}$. We also assume that N is a square-free integer with $\approx \log(p)$ many prime factors, all of them small, and that $E[N]$ is defined over $\mathbb{F}_{p^2}$. Define (P, Q) to be a basis of $E[N]$.

Following the reasoning in Section 3.1, we can compute $([a]\varphi(P), [a]\varphi(Q))$ for some $a \in (\mathbb{Z}/N\mathbb{Z})^*$. This means that we can efficiently compute the pushforward of any N-isogeny after three calls to the pushforward oracle, *i.e.* we can solve Problem 5.

On the other hand, if we manage to break Problem 3, we can recover a d-isogeny $\widehat{\varphi} : E \to E'$ such that $\widetilde{\varphi}_{|E[N]} = [b]\varphi_{|E[N]}$, for some $b \in (\mathbb{Z}/N\mathbb{Z})^*$. This indeed breaks [25, Problem 4], which is still considered hard [18, §5.3].

3.3 Easy instances of Problem 6, but hard Problem 5

Let $\varphi : E \to E'$ be an isogeny of degree d, and suppose we can query the oracle $\mathcal{O}^{\mathrm{PUSH}}_{N,\varphi}$. We will show an instance where Problem 6 can be easily solved, assuming that we have a quantum computer available or at least that we can solve elliptic curve discrete logarithm problems, but Problem 5 is believed to be difficult.

The example is based on the idea contained in [33]. Let E be a curve of known endomorphism ring and let N be a large prime. Also, assume that $E[N]$ is defined over $\mathbb{F}_{p^4}$. We now explain how to break Problem 6.

Let (P, Q) be a basis of $E[N]$. Since we know $\mathrm{End}(E)$, we can compute any N-isogeny originating from E. We query the isogenies with kernels $\langle P \rangle$, $\langle Q \rangle$ and $\langle P + Q \rangle$ to $\mathcal{O}^{\mathrm{PUSH}}_{\varphi,N}$ to obtain their pushforwards. We can obtain kernel generators of such pushforwards by solving some discrete logarithm problems modulo N; such computations would require employing quantum algorithms.

Once we obtain some kernel generators, we end up with the same system of equations as in Equation (1), which we can solve using quantum algorithms. This means that we have $(\varphi(P), \varphi(Q))$. It might be tempting to conclude that we can

actually recover φ, but, in general, this is not the case since N is not smooth [18, §5.2]. Recovering φ is still possible for some instances, as shown by Theorems 10 and 11 below, but these instances require some additional conditions that are usually not satisfied, and we now assume they do not hold.

Nevertheless, we can still solve Problem 6: we choose the isogeny with kernel $\langle R \rangle$, where $R = [a]P + [b]Q$; once we receive an isogeny $\sigma' : E \to E_{\sigma'}$, we simply check whether $\sigma'(\varphi(R)) = \sigma'([a]\varphi(P) + [b]\varphi(Q)) = 0$.

Assuming that we cannot recover the isogeny φ, the best known approach to breaking Problem 5 appears to be computing N-isogenies originating from E' without knowing $\mathrm{End}(E')$. At the moment, this seems to be out of reach as it would imply breaking [33, Problem 1].

4 Non-representable torsion

We now assume that N-torsion is not representable. At first sight, it may seem that having a pushforward oracle does not reveal anything since the information about the N-torsion cannot even be represented, let alone manipulated. However, when the domain of the secret isogeny $\varphi : E \to E'$ has known endomorphism ring, we show that there are still instances of Problem 3 (hence Problems 5 and 6) that can be solved in polynomial time.

We start by analysing the case where $d := \deg(\varphi) = \prod_{i=1}^{n} \ell_i^{e_i}$ is smooth. Given a $\theta \in \mathrm{End}(E)$ such that $\theta_{|E[d]} = [a]_{|E[d]}$ for some $a \in \mathbb{Z}/d\mathbb{Z}$, we define the *diagonalisable exponent* as follows.[5]

Definition 8. *Let θ and d as above. Let d' be the largest integer such that $\alpha_0 := \frac{2\theta - \mathrm{tr}(\theta)}{d \cdot d'}$ is in $\mathrm{End}(E)$, all prime divisors of d' divide $2d$, and for all $t \mid d$ we have $\alpha_{0|E[t]} \neq [0]_{|E[t]}$. Let $\alpha := [\varphi]_* [d]\alpha_0$. We define the diagonalisable exponent to be the largest integer m such that there exist distinct $i_1, \ldots, i_m \in [1, n]$ and, for all ℓ_{i_j}, we have that $\alpha_{|E'[\ell_{i_j}]} = [0]_{|E'[\ell_{i_j}]}$.*

Additionally, we introduce another technical definition that is needed to extend the proof to d divisible by 4. We highlight that such a condition could be further refined and is not relevant in the applications we describe in Section 6.

Definition 9. *Let θ and d as above and let α_0 as in Definition 8. We say that θ satisfies the mod-4 condition if either d is odd, or $d = 0 \pmod 4$ and $\deg(\alpha_0) \neq 3 \pmod 4$.*

Theorem 10. *Let $N \in O(\mathsf{poly}(p))$ and $d = \prod_{i=1}^{n} \ell_i^{e_i}$ be two coprime positive integers and suppose $\ell_i \in O(\mathsf{poly}(\log p))$ and $n \in O(\log(p))$. Also, let E be a supersingular elliptic curve of known endomorphism ring and let $\theta \in \mathrm{End}(E)$ such that, θ satisfies the mod-4 condition, $\theta_{|E[d]} = [a]_{|E[d]}$ for some $a \in \mathbb{Z}/d\mathbb{Z}$, $\theta \in \mathrm{End}(E) \setminus \mathbb{Z}$ and $\deg(\theta) = N^2 e$, for some smooth $e \in O(\mathsf{poly}(p))$. Consider a*

[5] This notion, although complicated, is only used to give a more precise complexity estimate.

cyclic d-isogeny $\varphi : E \to E'$. Then Algorithm 1 solves Problem 3 using two calls to the pushforward oracle $\mathcal{O}^{PUSH}_{\varphi,N}$, with a time complexity

$$O\left(\max\{2^m, \sqrt{e}\} \cdot \mathsf{poly}(\log(p))\right),$$

where m is the diagonalisable exponent in Definition 8.

We also generalise Theorem 10 to relax the smoothness requirements on d, provided we have *two* non-commuting endomorphisms with norms equal to small multiples of N^2, both acting as scalar multiplications on the d-torsion. As we will see in Section 5, this assumption is unlikely to hold in general, but specially crafted vulnerable instances can be efficiently constructed.

Theorem 11. *Let $N, d \in O(\mathsf{poly}(p))$ be two coprime positive integers. Also, let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ of known endomorphism ring $\mathrm{End}(E)$ and let $\theta_1, \theta_2 \in \mathrm{End}(E)$ such that $\theta_{i|E[d]} = [a_i]_{|E[d]}$ for some $a_i \in \mathbb{Z}/d\mathbb{Z}$ and $\deg(\theta_i) = N^2 e_i$, for some smooth e_i. Suppose there exists a set $\mathcal{S}$ of distinct prime powers such that*

- *$\left(\prod_{t\in\mathcal{S}} t\right)^2 \geq 4d + 1$;*
- *for all $t \in \mathcal{S}$, the t-torsion is representable and smooth;*
- *for all $t \in \mathcal{S}$, $\gcd(\deg(\theta_1\theta_2 - \theta_2\theta_1), t) = 1$.*

Consider a d-isogeny $\varphi : E \to E'$. Then, given oracle access to $\mathcal{O}^{PUSH}_{\varphi,N}$, there exists an algorithm to solve Problem 3 whose complexity is

$$O\left(\max\{2^{\#\mathcal{S}}, \sqrt{e_1}, \sqrt{e_2}\} \cdot \mathsf{poly}(\log(p))\right).$$

More precisely, the algorithm only requires four calls to $\mathcal{O}^{PUSH}_{\varphi,N}$.

The complexity of Theorem 11 heavily depends on the size of $\mathcal{S}$. We estimate such a quantity in Appendix A. The end result is summarised in the following corollary.

Corollary 12. *Let $N, d \in O(\mathsf{poly}(p))$ be two coprime positive integers. Also, let E be a supersingular elliptic curve of known endomorphism ring $\mathrm{End}(E)$ and let $\theta_1, \theta_2 \in \mathrm{End}(E)$ such that $\theta_{i|E[d]} = [a_i]_{|E[d]}$ for some $a_i \in \mathbb{Z}/d\mathbb{Z}$ and $\deg(\theta_i) = N^2 e_i$, for some smooth $e_i \in O(\mathsf{poly}(p))$.*

Consider a d-isogeny $\varphi : E \to E'$. Then, given oracle access to $\mathcal{O}^{PUSH}_{\varphi,N}$, there exists an algorithm to solve Problem 3 whose complexity is

$$O\left(\max\left\{2^{\frac{\ln(d)}{\ln\left(\ln(d^{1/2}\cdot N^4\cdot e_1\cdot e_2)\right)}}, \sqrt{e_1}, \sqrt{e_2}\right\} \cdot \mathsf{poly}(\log(p))\right).$$

More precisely, the algorithm only requires four calls to $\mathcal{O}^{PUSH}_{\varphi,N}$.

Remark 13. We highlight that the complexity estimated in Corollary 12 deals with the most general case. In practice, isogeny-based cryptosystems utilise primes of a special form where a lot of smooth torsion is defined over the base field. Such a choice may allow for an $\mathcal{S}$ of smaller size, hence lowering the overall cost.

4.1 Proof of Theorem 10

To prove Theorem 10, we need to prove the complexity of Algorithm 1. Such a proof requires several technical lemmata we prove below. In the rest of this section, we will assume that

- N and d are two coprime positive integers, with d smooth,
- E is a supersingular elliptic curve of known endomorphism ring $\mathrm{End}(E)$,
- $\theta \in \mathrm{End}(E)$ such that, $\theta_{|E[d]} = [a]_{|E[d]}$ for some $a \in \mathbb{Z}/d\mathbb{Z}$, $\theta \in \mathrm{End}(E) \setminus \mathbb{Z}$ and $\deg(\theta) = N^2 e$, for some smooth e,
- $\varphi : E \to E'$ is a cyclic d-isogeny,
- m is the diagonalisable exponent in Definition 8,
- α_0 is the endomorphism defined in Definition 8,
- d' is the quantity defined in Definition 8,
- θ satisfies the mod-4 condition in Definition 9, *i.e.*, either d is odd, or $d = 0 \pmod 4$ and $\deg(\alpha_0) \neq 3 \pmod 4$.

Lemma 14. *There exists an algorithm to compute a representation of $[\varphi]_* \theta \in \mathrm{End}(E')$ that requires querying $\mathcal{O}_{\varphi,N}^{PUSH}$ twice and runs in $O\left(\sqrt{e} \cdot \mathsf{poly}(\log(p))\right)$.*

Proof. Since $\theta_{|E[d]} = [a]_{|E[d]}$, it is easy to see that $[\varphi]_* \theta \in \mathrm{End}(E')$. Since E is a curve of known endomorphism ring, we can compute the isogenies $\psi_1 : E \to E_1$ and $\psi_2 : E \to E_2$ whose kernels are given by $\ker(\theta)[N]$ and $\ker(\widehat{\theta})[N]$, respectively.

We now query the oracle $\mathcal{O}_{\varphi,N}^{\mathrm{PUSH}}$ on ψ_1 and ψ_2 and obtain $[\varphi]_* \psi_1 : E' \to E_1'$ and $[\varphi]_* \psi_2 : E' \to E_2'$. Since $[\varphi]_* \theta \in \mathrm{End}(E')$, there exists an isogeny $\rho : E_1' \to E_2'$ of degree e. Since e is smooth, we compute this isogeny at the cost of $O\left(\sqrt{e} \cdot \mathsf{poly}(\log(p))\right)$ via a meet-in-the-middle attack [1]. We conclude the proof observing that a representation of $[\varphi]_* \theta$ is given by $(\widehat{[\varphi]_* \psi_2}) \circ \rho \circ ([\varphi]_* \psi_1)$. □

We now need to show that we can efficiently compute α_0 and d'.

Lemma 15. *There exists a polynomial-time algorithm[6] to compute d' and a representation of α_0 as defined in Definition 8.*

Proof. Using the division algorithm in [36, Theorem 3], we compute a representation of $\frac{[2]\theta - \mathrm{tr}(\theta)}{[d]}$ in polynomial time. Remember that d is smooth. For each prime divisor ℓ of $2d$, we compute the largest k such that $\frac{[2]\theta - \mathrm{tr}(\theta)}{[d \cdot \ell^k]}$ using again [36, Theorem 3]; d' is the product of such prime powers. Since there are at most $O(\log(p))$ prime divisors and since $\deg\left(\frac{[2]\theta - \mathrm{tr}(\theta)}{[d]}\right) \leq 4\frac{(N^2 e - (\mathrm{tr}(\theta)/2)^2)}{d^2} \in O(\mathsf{poly}(p))$, this procedure is polynomial time. □

Lemma 16. *Given a representation of $[\varphi]_* \theta$, we also have a representation of the pushforward of $[d]\alpha_0$ under φ.*

[6] We remark that, by definition, we assume knowledge of $\mathrm{End}(E)$, *i.e.*, an efficient representation of a basis of $\mathrm{End}(E)$ and its correspondence as a maximal order of the quaternion algebra. This means that technically we do not need to use [36, Theorem 3]. Nevertheless, we prefer to use this result as we will need it in Lemma 16.

Algorithm 1 Smooth Isogeny Recovery

Input: The input data of Theorem 10. The degree d has factorisation $\prod_{i=1}^{n} \ell_i^{e_i}$.
Output: The isogeny $\varphi : E \to E'$.

Compute $[\varphi]_* \theta$ using $\mathcal{O}_{\varphi,N}^{\text{PUSH}}$.
Compute $\alpha = [\varphi]_* [d]\alpha_0$.
$s \leftarrow 1$
for $i = 1, \ldots, n$ **do**
    $k \leftarrow 0$
    while $\alpha_{|E'[\ell_i^k]} = [0]_{|E'[\ell_i^k]}$ **do**
        $k \leftarrow k + 1$
    if $\ell_i = 2$ **then**
        Compute up to four 2^k-isogenies $\varphi_{2^k}^{(1)} : E \to E_1, \ldots, \varphi_{2^k}^{(4)} : E \to E_4$, according to Lemma 20.
        $s \leftarrow s \cdot 2^k$
    else
        Compute up to two ℓ_i^k-isogenies $\varphi_{\ell_i^k}^{(1)} : E \to E_1$ and $\varphi_{\ell_i^k}^{(2)} : E \to E_2$ according to Lemma 19.
        $s \leftarrow s \cdot \ell_i^k$
Compute the set $\mathcal{S}_{\mathsf{Isog}}$ of all the possible combinations of pushforwards of the isogenies $\varphi_{\ell_i^k}^{(i)}$'s.
$d \leftarrow d/s$
Factor $d = \prod_{i=1}^{k} \ell_i^{f_i}$.
$\rho \leftarrow [1]$
for $i = 1, \ldots, k$ **do**
    for $j = 1, \ldots, f_i$ **do**
        Find the ℓ_i-isogeny ψ such that $[\psi]_* \alpha$ is divisible by $[\ell_i]$.
        $\rho \leftarrow \psi \circ \rho$
        $\alpha \leftarrow [\psi]_* \alpha$
Find in $\mathcal{S}_{\mathsf{Isog}}$ the isogeny χ whose codomain corresponds to the codomain of ρ.
return $\widehat{\rho} \circ \chi$

Proof. It is easy to see that $[d]$ divides $[2] \cdot [\varphi]_* [\theta] - \mathrm{tr}(\theta)$. By definition, we have that $[\varphi]_* ([d]\alpha_0) = \frac{[2] \cdot [\varphi]_* \theta - \mathrm{tr}(\theta)}{[d']}$. Additionally $\deg ([\varphi]_* ([d]\alpha_0)) \leq 4 \frac{(N^2 e - (\mathrm{tr}(\theta)/2)^2)}{d'^2}$, which is a quantity polynomial in p. The result follows from [36, Theorem 3]. □

The following lemmata characterise the information we recover from the pushforward of an endomorphism.

Lemma 17. *Let $\tilde{d}$ be a factor of d. Then, $([\varphi]_* [d]\alpha_0)_{|E'[\tilde{d}]} = [0]_{|E'[\tilde{d}]}$ if and only if $\alpha_0(\ker(\varphi)[\tilde{d}]) \subset \ker(\varphi)[\tilde{d}]$.*

Proof. Define $\alpha := [\varphi]_* [d]\alpha_0$. We observe that $\alpha_{|E'[\tilde{d}]} = [0]_{|E'[\tilde{d}]}$ is equivalent to saying that there exists $\alpha' \in \mathrm{End}(E')$ such that $[\tilde{d}]\alpha' = \alpha$. In other words $[\tilde{d}]\alpha' = \varphi \circ \alpha_0 \circ \widehat{\varphi}$, which happens if and only $\alpha_0(\ker(\varphi)[\tilde{d}]) \subset \ker(\varphi)[\tilde{d}]$. □

Lemma 18. *Let $\ell > 2$ be a prime dividing d such that $([\varphi]_* [d]\alpha_0)_{|E'[\ell]} = [0]_{|E'[\ell]}$. Then, there exist at most two subgroups K_1 and K_2 of order ℓ such that either $\ker(\varphi)[\ell] = K_1$ or $\ker(\varphi)[\ell] = K_2$. Moreover, these subgroups can be computed in polynomial time in $\log(p)$.*

Proof. From Lemma 17, we have that $\alpha_0(\ker(\varphi)[\ell]) \subset \ker(\varphi)[\ell]$. To prove this lemma, we will show how to compute the subgroups of the ℓ-torsion fixed by α_0.

By fixing a basis (P, Q) of the ℓ-torsion, we can identify $\alpha_{0|E[\ell]}$ with a matrix $\mathbf{M} = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{M}_{2,2}(\mathbb{Z}/\ell\mathbb{Z})$, i.e., $\alpha_0(P) = [a]P + [b]Q$ and $\alpha_0(Q) = [c]P + [d]Q$; the quantities a, b, c, d can be computed in polynomial time by the assumption that ℓ is polynomial in $\log(p)$. In particular $\det(\mathbf{M}) = (\mathrm{nrd}(\alpha_0) \bmod \ell)$ and $\mathrm{tr}(\mathbf{M}) = 0 = (\mathrm{tr}(\alpha_0) \bmod \ell)$. Moreover, finding the subgroup fixed by α_0 amounts to finding the subgroups fixed by $\mathbf{M}$. To do so, we analyse the diagonalisability of $\mathbf{M}$.

We start by computing the eigenvalues of $\mathbf{M}$, *i.e.*, the solutions of the equation $x^2 - \det(\mathbf{M})$. We first remark that it cannot happen that $\det(\mathbf{M})$ is not a square modulo ℓ: if this were the case, we would have that there are no eigenspaces, which would contradict the fact that $\alpha_0(\ker(\varphi)[\ell]) \subset \ker(\varphi)[\ell]$. If $\det(\mathbf{M}) = 0$, we must have that $\mathbf{M} \neq 0$ since $[\ell]$ does not divide α_0. If this occurs, the matrix $\mathbf{M}$ is not diagonalisable, but will still admit an eigenspace of rank one. In particular, this eigenspace coincides with $\ker(\varphi)[\ell]$.

We now look at the case where $\det(\mathbf{M}) \neq 0$ is a square. This means that the eigenvalues are among the solutions of $x^2 - \det(\mathbf{M})$ and all have multiplicity one. In particular, $\mathbf{M}$ is diagonalisable and one of the eigenspaces coincides with $\ker(\varphi)[\ell]$. □

Lemma 19. *Let ℓ^k be an odd prime power dividing d such that $([\varphi]_* [d]\alpha_0)_{|E[\ell^k]} = [0]_{|E[\ell^k]}$, for some $k > 0$. Then, there exists a polynomial-time algorithm to compute at most two ℓ^k-isogenies; one of those is the isogeny with kernel $\ker(\varphi)[\ell^k]$.*

Proof. By Lemma 18, we can compute at most two ℓ-isogenies, such that one of those is the one with kernel $\ker(\varphi)[\ell]$, in polynomial time. Let $\psi : E \to E_1$ be one such isogeny and define $\alpha_1 := [\psi]_* \alpha_0 \in \mathrm{End}(E_1)$. It is easy to observe that $\mathrm{tr}(\alpha_0) = \mathrm{tr}(\alpha_1)$ and $\mathrm{nrd}(\alpha_0) = \mathrm{nrd}(\alpha_1)$. Hence, when we consider the reduction of α_0 and α_1 to matrices $\mathbf{M}_0, \mathbf{M}_1 \in \mathrm{M}_{2,2}(\mathbb{Z}/\ell\mathbb{Z})$, respectively; such matrices have the same characteristic polynomial.

We observe that, if $k > 1$, the matrix $\mathbf{M}_1$ must be diagonalisable. If this is not the case, by Lemma 17 applied to E_1 and α_1, there exists a unique eigenspace for $\mathbf{M}_1$, which coincides with the dual of ψ. This leads to a contradiction since it is not possible that $([\varphi]_* [d]\alpha_0)_{|E[\ell^k]} = [0]_{|E[\ell^k]}$.

Proceeding in the same fashion as in the proof of Lemma 18, one of the two eigenspaces will describe the backtracking isogeny (*i.e.* the dual of ψ); we then compute the ℓ-isogeny corresponding to the other eigenspace. Mutatis mutandis, we proceed for the remaining $k - 1$ steps. By construction, one of the two paths is the correct one.

As for the complexity, since $\log(d)$ is polynomial in $\log(p)$, we have that the number of steps is at most polynomial in $\log(p)$. Therefore, our complexity claim follows. □

Lemma 20. *Let $k > 0$ be an exponent such that $2^k \mid d$ and $([\varphi]_* [d]\alpha_0)_{|E[2^k]} = [0]_{|E[2^k]}$. Then, there exists a polynomial-time algorithm to compute at most four 2^k-isogenies; one of those is the isogeny with kernel $\ker(\varphi)[2^k]$.*

Proof. The proof is a straightforward adaptation of the techniques in the proofs of Lemma 18 and Lemma 19. We refer the reader to Appendix B for more details. □

Lemma 21. *Let $\tilde{d}$ be a prime factor of d. Suppose that $\alpha_0(\ker(\varphi)[\tilde{d}]) \not\subset \ker(\varphi)[\tilde{d}]$. Define $\alpha := [\varphi]_* [d]\alpha_0$ and let $\widetilde{\varphi} : E' \to \tilde{E}$ be a $\tilde{d}$-isogeny. Then $\widetilde{\alpha} := [\widetilde{\varphi}]_* \alpha \in [\tilde{d}] \operatorname{End}(\widetilde{E})$ if and only if $\ker(\widetilde{\varphi}) = \ker(\widehat{\varphi})[\tilde{d}]$.*

Proof. The implication that, if $\ker(\widetilde{\varphi})[\tilde{d}] = \ker(\widehat{\varphi})[\tilde{d}]$ then $\widetilde{\alpha} := [\widetilde{\varphi}]_* \alpha \in [\tilde{d}] \operatorname{End}(\widetilde{E})$ is clear. We now prove the converse.

From Lemma 17, it follows that $\alpha_{|E[\tilde{d}]} \neq [0]_{|E[\tilde{d}]}$. Since $\widetilde{\alpha} \in [\tilde{d}] \operatorname{End}(\widetilde{E})$, there exists $\widetilde{\alpha'} \in \operatorname{End}(\widetilde{E})$ such that $[\tilde{d}^2]\widetilde{\alpha'} = \widetilde{\varphi} \circ \alpha \circ \widehat{\widetilde{\varphi}}$. Also, from $\alpha_{|E'[\tilde{d}]} \neq [0]_{|E'[\tilde{d}]}$, it follows that $\alpha(E'[\tilde{d}]) = \ker(\widetilde{\varphi})$. By definition, we have that $[d^2]\alpha_0 = \widehat{\varphi} \circ \alpha \circ \varphi$; this implies that $\alpha(E'[d]) = \ker(\widehat{\varphi})[\tilde{d}]$. □

Proof (of Theorem 10). Let $d = \prod_{i=1}^{n} \ell_i^{e_i}$. We briefly give an overview of Algorithm 1 and then discuss its complexity. We first compute $[\varphi]_* [d]\alpha_0$. Then, we identify the prime powers $\ell_i^{k_i}$ such that $\alpha_{|E[\ell_i^k]} = [0]_{|E[\ell_i^k]}$. This means that, for each of these prime powers, there are at most four candidate $\ell_i^{k_i}$-isogenies $\varphi_{\ell_i^{k_i}}^{(1)} : E \to E_1, \ldots, \varphi_{\ell_i^{k_i}}^{(4)} : E \to E_4$ according to Lemma 20 if $\ell_i = 2$, or there are at most two candidate $\ell_i^{k_i}$-isogenies $\varphi_{\ell_i^{k_i}}^{(1)} : E \to E_1$ and $\varphi_{\ell_i^{k_i}}^{(2)} : E \to E_2$ according to Lemma 19. One of these isogenies factors φ.

We define $\mathcal{S}_{\mathsf{lsog}}$ to be the set of all the possible combinations of pushforwards of the isogenies $\varphi_{\ell_i^k}^{(i)}$'s. A priori only one of the isogenies in $\mathcal{S}_{\mathsf{lsog}}$ factors φ. The rest of the algorithm is dedicated to computing the remaining part of φ by iteratively using Lemma 21 to meet one of the isogenies in $\mathcal{S}_{\mathsf{lsog}}$ starting from E'.

We now discuss the complexity of Algorithm 1. The computation in line 1 can be carried in $O(\sqrt{e} \cdot \mathsf{poly}(\log(p)))$ with two queries from $\mathcal{O}_{\varphi,N}^{\mathrm{PUSH}}$ as explained in Lemma 14. A representation of the endomorphism α_0 can be obtained in polynomial time via Lemma 15. From this representation we can then get a representation of $\alpha = [\varphi]_* [d]\alpha_0$ in line 2 via Lemma 16.

We now discuss the complexity of the loop in line 4. In each single step, we have to perform at most $O(\log(p))$ divisions in line 6 using [36, Theorem 3] The cost of the while loop in line 6 is $O(\mathsf{poly}(\log(p)))$. The isogenies in lines 9 and 12 can be computed in polynomial time via Lemma 20 and Lemma 19. Since $n \in O(\log(p))$, the whole loop in line 4 is polynomial time. Moreover, we

end up with at most $m-1$ pairs and 1 quadruple of isogenies, where m is the diagonalisable exponent.

We now explain why the step in line 14 can be carried in $O\left(2^m \cdot \mathsf{poly}(\log(p))\right)$. Since we have at most $m-1$ pairs and 1 quadruple of isogenies, there are up to 2^{m+1} possible combinations of pushforwards. To be clearer, assume that the s in line 13 is equal to $\prod_{k=1}^{m} \ell_{i_k}^{f_{i_k}}$, where $\ell_{i_1} \leq \ldots \leq \ell_{i_m}$. The set $\mathcal{S}_{\mathsf{Isog}}$ consists of all isogenies of degree s obtained by iteratively pushing forward the isogenies $\varphi_{\ell_{i_1}^{f_{i_1}}}^{(j_1)}, \varphi_{\ell_{i_2}^{f_{i_2}}}^{(j_2)}, \ldots, \varphi_{\ell_{i_m}^{f_{i_m}}}^{(j_m)}$ along one another, for all binary strings $(j_1, \ldots, j_m) \in \{1,2,3,4\} \times \{1,2\}^{m-1}$.

The factorisation in line 16 is extremely efficient since d is known to be smooth. This cost can be ignored.

We finally discuss the cost of the loop in line 18, which is polynomial time. We start by observing that in lines 18 and 19 require at most $O\left(\log(p)^2\right)$ iterations. The step in line 20 takes $O\left(\ell_i^2\right)$ field operations, which is again polynomial time since the $\ell_i \in O\left(\mathsf{poly}(\log(p))\right)$. The step in line 22 is again polynomial time because ψ is an isogeny of degree ℓ_i. This shows that the loop in line 18 takes $O\left(\mathsf{poly}(\log(p))\right)$ operations and thus concludes the proof. □

4.2 Proof of Theorem 11

To prove Theorem 11, we need the following results. In the rest of this section, we will assume that

- N and d are two coprime positive integers,
- E is a supersingular elliptic curve of known endomorphism ring $\mathrm{End}(E)$,
- $\theta_1, \theta_2 \in \mathrm{End}(E)$ such that $\theta_{i|E[d]} = [a_i]_{|E[d]}$ for some $a_i \in \mathbb{Z}/d\mathbb{Z}$ and $\deg(\theta_i) = N^2 e_i$, for some smooth e_i,
- $\mathcal{S}$ is a set of distinct prime powers such that
  - $\left(\prod_{t \in \mathcal{S}} t\right)^2 \geq 4d+1$;
  - for all $t \in \mathcal{S}$, the t-torsion is representable and smooth;
  - for all $t \in \mathcal{S}$, $\gcd(\deg(\theta_1\theta_2 - \theta_2\theta_1), t) = 1$.

By Lemma 14, we can compute $\theta_i^{E'} := [\varphi]_* \theta_i$ in $O\left(\sqrt{e_i} \cdot \mathsf{poly}(\log(p))\right)$ using two calls to $\mathcal{O}_{\varphi,N}^{\mathrm{PUSH}}$. For each $a, b, c, d \in \mathbb{Z}$, we define $\theta_{a,b,c,d} := [a] + [b]\theta_1 + [c]\theta_2 + [d]\theta_1\theta_2$ and $\theta_{a,b,c,d}^{E'} := [a] + [b]\theta_1^{E'} + [c]\theta_2^{E'} + [d]\theta_1^{E'}\theta_2^{E'}$.

Lemma 22. *If* $\gcd(\deg(\theta_1\theta_2 - \theta_2\theta_1), t) = 1$, *then there exist* $(a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2) \in \mathbb{Z}^4$ *such that*

- $\theta_{a_1,b_1,c_1,d_1}\theta_{a_2,b_2,c_2,d_2} - \theta_{a_2,b_2,c_2,d_2}\theta_{a_1,b_1,c_1,d_1|E_0[\ell]} \neq [0]_{|E_0[\ell]}$, *where* $t = \ell^k$;
- $\mathrm{tr}(\theta_{a_i,b_i,c_i,d_i})^2 - 4\,\mathrm{nrd}(\theta_{a_i,b_i,c_i,d_i})$ *is a non-zero square modulo* t.

Moreover, such $(a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2) \in \mathbb{Z}^4$ *can be computed in polynomial time.*

Proof. We fix $t = \ell^k$, where ℓ is a prime. We first compute the matrices $\mathbf{M}_1, \mathbf{M}_2 \in \mathrm{M}_{2,2}(\mathbb{Z}/\ell^k\mathbb{Z})$ corresponding to θ_1 and θ_2, respectively; this can be done in polynomial time. Similarly to what it was done in Lemma 18, we can compute a basis (P, Q) of $E[t]$ in $O(\mathsf{poly}(\log(p)))$ because t is smooth and representable. Following the same strategy as in the proof of Lemma 18, we compute $\mathbf{M}_i = \begin{pmatrix} c_1^{(i)} & c_2^{(i)} \\ c_3^{(i)} & c_4^{(i)} \end{pmatrix} \in \mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$ corresponding to $\theta_i(P) = \left[c_1^{(i)}\right] P + \left[c_2^{(i)}\right] Q$ and $\theta_i(Q) = \left[c_3^{(i)}\right] P + \left[c_4^{(i)}\right] Q$; this step can be carried out in polynomial time by the assumption that ℓ is polynomial in $\log(p)$ and $k \in O(\log(p))$.

We recall that $\mathrm{End}(E)/t\,\mathrm{End}(E)$ is isomorphic to $\mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$ [49, Theorem 42.1.9]. In particular, we can assume that there is an isomorphism such that $\theta_{i|E_0[t]}$ corresponds to $\mathbf{M}_i$. We now prove that $\langle 1, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_1\mathbf{M}_2 \rangle$ coincides with $\mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$, and in turn $\langle [1]_{|E[t]}, \theta_{1|E[t]}, \theta_{2|E[t]}, \theta_1\theta_{2|E[t]} \rangle$ is equal to $\mathrm{End}(E)/t\,\mathrm{End}(E)$.

We preliminary observe that $\mathbf{T} := \mathbf{M}_1 \cdot \mathbf{M}_2 - \mathbf{M}_2 \cdot \mathbf{M}_1$ is an invertible matrix by the assumption $\gcd(\deg(\theta_1\theta_2 - \theta_2\theta_1), t) = 1$. Since $\mathbf{M}_1$ and $\mathbf{M}_2$ do not commute, we have that $\langle 1, \mathbf{M}_1, \mathbf{M}_2 \rangle$ is a lattice of rank 3. Now, suppose by contradiction that $\mathbf{M}_1\mathbf{M}_2 \in \langle 1, \mathbf{M}_1, \mathbf{M}_2 \rangle$. This implies that there exists a non-trivial tuple (a, b, c, d) such that

$$a + b\mathbf{M}_1 + c\mathbf{M}_2 + d\mathbf{M}_1\mathbf{M}_2 = 0. \tag{2}$$

From which, by multiplying by $\mathbf{M}_1$, we have that

$$a\mathbf{M}_1 + b\mathbf{M}_1^2 + c\mathbf{M}_1\mathbf{M}_2 + d\mathbf{M}_1^2\mathbf{M}_2 = 0, \tag{3}$$

and

$$a\mathbf{M}_1 + b\mathbf{M}_1^2 + c\mathbf{M}_2\mathbf{M}_1 + d\mathbf{M}_1\mathbf{M}_2\mathbf{M}_1 = 0. \tag{4}$$

Subtracting Equation (4) from Equation (3), we have that

$$c\mathbf{T} + d\mathbf{M}_1\mathbf{T} = 0,$$

i.e.

$$(c + d\mathbf{M}_1)\mathbf{T} = 0. \tag{5}$$

From Equation (5), since $\mathbf{T}$ is invertible, we get that $c + d\mathbf{M}_1 = 0 \pmod{\ell^k}$. Since $\mathbf{M}_1$ and $\mathbf{M}_2$ do not commute (modulo ℓ), this means that $c = d = 0 \pmod{\ell}$. Mutatis mutandis, multiplying Equation (2) by $\mathbf{M}_2$, we get that $b = 0 \pmod{\ell}$, which in turn implies that $a = 0 \pmod{\ell}$. As a consequence, we have that $a = b = c = d = 0 \pmod{\ell}$. This means that we can divide a, b, c, d by ℓ and consider Equation (2) modulo ℓ^{k-1}. Repeating this process k times, we obtain that $a = b = c = d = 0 \pmod{t}$, a contradiction.

We have proven that $\langle 1, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_1\mathbf{M}_2 \rangle$ is a rank-4 sublattice of $\mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$, which is also a rank-4 lattice modulo ℓ. This means that $\langle 1, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_1\mathbf{M}_2 \rangle = \mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$. As a result we have an explicit isomorphism $\varepsilon : \mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z}) \to \mathrm{End}(E_0)/t\,\mathrm{End}(E_0)$ given as $\varepsilon(1) = [1]_{|E[t]}$, $\varepsilon(\mathbf{M}_1) = \theta_{1|E[t]}$, $\varepsilon(\mathbf{M}_2) = \theta_{2|E[t]}$ and $\varepsilon(\mathbf{M}_1\mathbf{M}_2) = \theta_1\theta_{2|E[t]}$.

We conclude the proof of the theorem by finding (a_1, b_1, c_1, d_1) such that $a_1 + b_1\mathbf{M}_1 + c_1\mathbf{M}_2 + d_1\mathbf{M}_1\mathbf{M}_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ and (a_2, b_2, c_2, d_2) such that $a_2 + b_2\mathbf{M}_1 + c_2\mathbf{M}_2 + d_2\mathbf{M}_1\mathbf{M}_2 = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. The endomorphisms θ_{a_1,b_1,c_1,d_1} and θ_{a_2,b_2,c_2,d_2} satisfy the properties required. □

For each $t = \ell^k \in \mathcal{S}$, we will assume that we have computed $\theta_{a_i^t,b_i^t,c_i^t,d_i^t}$ such that

- $\theta_{a_1^t,b_1^t,c_1^t,d_1^t}\theta_{a_2^t,b_2^t,c_2^t,d_2^t} - \theta_{a_2^t,b_2^t,c_2^t,d_2^t}\theta_{a_1^t,b_1^t,c_1^t,d_1^t}{}_{|E[\ell]} \neq [0]_{|E[\ell]}$;
- $\mathrm{tr}(\theta_{a_i^t,b_i^t,c_i^t,d_i^t})^2 - 4\,\mathrm{nrd}(\theta_{a_i^t,b_i^t,c_i^t,d_i^t})$ is a non-zero square modulo t.

For the sake of simplifying notation, we define $\theta_{i^t} := \theta_{a_i^t,b_i^t,c_i^t,d_i^t}$.

Lemma 23. *Given $t \in \mathcal{S}$, we can compute a basis (P, Q) of $E[t]$ and two integers (s_P, s_Q) such that $\theta_{i^t}(P) = [s_P]P$ and $\theta_{i^t}(Q) = [s_Q]Q$ in polynomial time.*

Proof. We can compute a basis (P', Q') of $E[t]$ in $\mathsf{poly}(\log(p))$ since t is smooth and representable. Then we compute a matrix $\mathbf{M} = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{M}_{2,2}(\mathbb{Z}/t\mathbb{Z})$ corresponding to θ_{i^t}, *i.e.* $\theta_{i^t}(P') = [a]P' + [b]Q'$ and $\theta_{i^t}(Q') = [c]P' + [d]Q'$. Since $\mathrm{tr}(\theta_{i^t})^2 - 4\,\mathrm{nrd}(\theta_{i^t})$ is a non-zero square modulo t, then the matrix $\mathbf{M}$ is diagonalisable. The eigenvectors correspond to (P, Q). □

Clearly, if $\theta_{i^t}(P) = [s_P]P$ and $\theta_{i^t}(Q) = [s_Q]Q$, then $\theta_{i^t}^{E'}(\varphi(P)) = [s_P]\varphi(P)$ and $\theta_{i^t}^{E'}(\varphi(Q)) = [s_Q]\varphi(Q)$. It is also worth mentioning that Lemma 23 applies to $\theta_{i^t}^{E'}$ as well.

Proposition 24. *Let $t \in \mathcal{S}$. Then, we can compute a basis (P, Q) of $E[t]$ and $\pm(\varphi(P), \varphi(Q))$ if $t \neq 0 \pmod 8$, or $\pm([1 + \epsilon 2^{k-1}]\varphi(P), [1 + \epsilon 2^{k-1}]\varphi(Q))$ where $t = 2^k$ and $\epsilon \in \{0, 1\}$. This computation can be carried in polynomial time.*

Proof. We postpone the proof to Appendix C. □

Proof (of Theorem 11). For all $t \in \mathcal{S}$, we can compute a basis (P_t, Q_t) of $E_0[t]$ and $\pm(\varphi(P), \varphi(Q))$ if $t \neq 0 \pmod 8$, or $\pm([1 + \epsilon 2^{k-1}]\varphi(P), [1 + \epsilon 2^{k-1}]\varphi(Q))$ where $t = 2^k$ and $\epsilon \in \{0, 1\}$, in polynomial time via Proposition 24. For each t, we have some ambiguities. This means that there are up to $2^{\#\mathcal{S}+1}$ possibilities, and among these, only two of them represent $\pm\left(\sum_{t\in\mathcal{S}} \varphi(P_t), \sum_{t\in\mathcal{S}} \varphi(Q_t)\right)$.

To check that a choice is correct, we need to try to represent φ using that torsion information [45, Remark 5.20],[44]. Every time we try, we incur a cost of $O\left(\#\mathcal{S} \cdot \mathsf{polylog}(p)\right) = O\left(\mathsf{polylog}(p)\right)$. As a consequence, the overall cost of the attack using this strategy is

$$O\left(\max\{2^{\#\mathcal{S}}, \sqrt{e_1}, \sqrt{e_2}\} \cdot \mathsf{poly}(\log(p))\right),$$

as claimed. We finally remark that we need four calls to $\mathcal{O}_{\varphi,N}^{\mathrm{PUSH}}$ because computing $\theta_i^{E'} := [\varphi]_*\,\theta_i$, for $i = 1, 2$, requires four calls to $\mathcal{O}_{\varphi,N}^{\mathrm{PUSH}}$. □

5 Vulnerable instances

In this section, we explain how to find parameter sets and curves verifying the conditions in Theorems 10 and 11. While it is highly unlikely that vulnerable curves can just be encountered by randomly walking the isogeny graph, such instances might be created as backdoors.

The rest of this section is organised as follows. We first rephrase the conditions for Theorems 10 and 11. To find vulnerable parameters, we adapt the "intersection method" to our use case, see Section 5.3. The intersection method builds on top of some standard results for rational conics, which we recall in Section 5.2. In Section 5.4, we explain how to find parameters susceptible to Theorem 10. In Section 5.5, we explain how to find parameters susceptible to Theorem 11; we also provide some accompanying code at

https://github.com/LucianoMaino/Pushforward-Problems-and-Applications-to-Isogeny-based-Cryptography

5.1 Rephrasing conditions for Theorems 10 and 11

For Theorem 10, we need to find a prime p, integers N, e, d with e small and d suitably smooth, a curve E over $\mathbb{F}_{p^2}$, and an endomorphism $\theta \in \mathrm{End}(E) \setminus \mathbb{Z}$ such that $\theta_{|E[d]} = [w]_{|E[d]}$ for some $w \in \mathbb{Z}/d\mathbb{Z}$ and $\deg(\theta) = N^2 e$. Since we require $\theta_{|E[d]} = [w]_{|E[d]}$ for some $w \in \mathbb{Z}/d\mathbb{Z}$, we can write $\theta = w + d\theta'$, and search for w in $\mathbb{Z}$ and θ' in the trace zero subring of $\mathrm{End}(E)$, such that

$$\deg(\theta) = N^2 e = w^2 + d^2 \deg(\theta').$$

For Theorem 11, we need two endomorphisms $\theta_1, \theta_2 \in \mathrm{End}(E)$ satisfying the same condition, which moreover generate a full-rank suborder. In both cases, a natural approach is to identify endomorphisms with elements of $\mathcal{B}_{p,\infty}$. For the sake of simplifying our search, we ignore the role of the set $\mathcal{S}$ in the statement of Theorem 11. As explained in Remark 13, in some applications in isogeny-based cryptography, it is customary to use primes where a lot of smooth torsion is accessible.

In the case of Theorem 10, we first search for parameters p, N, e, d and quaternion elements θ, θ' satisfying a corresponding norm equation, i.e.

$$\mathrm{nrd}(\theta) = N^2 e = w^2 + d^2 \,\mathrm{nrd}(\theta')$$

with $\mathrm{tr}(\theta') = 0$. Interestingly, this equation is exactly the one appearing in the "lollipop attacks" (see Section 2.4), though we will give an alternative resolution method in Section 5.4. Next, we search for a maximal order $\mathcal{O}$ containing θ. Finally, we compute a supersingular elliptic curve E with an explicit isomorphism $\mathrm{End}(E) \simeq \mathcal{O}$ [29,23,50]. Applying this isomorphism to θ gives the required endomorphism; a similar approach is used in [43].

In the case of Theorem 11, additional algebraic constraints follow from the fact that θ_1 and θ_2 must generate a rank 4 suborder.

Lemma 25. *Two integral elements $\theta_1, \theta_2 \in B_{p,\infty}$ generate an order if and only if $t := \mathrm{tr}(\theta_1\theta_2) \in \mathbb{Z}$. When the two elements have the same trace in absolute value and the same norm, the order is full-rank unless $\theta_2 \in \{\pm\theta_1, \pm\bar{\theta}_1\}$.*

Proof. We postpone the proof to Appendix D. □

The search for parameters for Theorem 11 will therefore proceed by finding parameters p, N, e, d and two distinct pairs of quaternions $\theta_1, \theta_1', \theta_2, \theta_2'$ satisfying

$$\mathrm{nrd}(\theta_i) = N^2 e = w^2 + d^2\,\mathrm{nrd}(\theta_i'), \qquad \mathrm{tr}(\theta_i') = 0, \qquad \mathrm{tr}(\theta_1\theta_2) \in \mathbb{Z}\backslash\{\pm 2\,\mathrm{nrd}(\theta_1)\}.$$

We will then extend the order $\Lambda := \langle 1, \theta_1, \theta_2, \theta_1\theta_2\rangle$ into a maximal order, and proceed as before.

5.2 Solvability conditions for rational conics

We recall standard results regarding the solvability of conics [16]. Let $C(X, Y, Z) = 0$ be a projective quadratic equation over a field $\mathbb{Q}$. By the Hasse principle, this equation has solutions over $\mathbb{Q}$ if and only if it has solutions over $\mathbb{R}$ and $\mathbb{Q}_p$ for all primes p.

Any such equation can be transformed into an equation in *diagonal* or *Legendre* form

$$AX_1^2 + BX_2^2 + CX_3^2 = 0$$

with A, B, C pairwise coprime integers, through an efficiently computable invertible linear change of variables. In that case, the equation has solutions over $\mathbb{Q}$ if it has solutions over $\mathbb{R}$ and modulo every prime dividing ABC.

At the cost of factoring ABC, one can further assume that ABC is squarefree, and in this case the equation has solutions over $\mathbb{Q}$ if and only if it has solutions over $\mathbb{R}$ and modulo every prime dividing ABC. Moreover, when a solution exists, a small solution can be computed in polynomial time using the algorithm from [16].

5.3 The intersection method

We recall the *intersection method* which we will use below for both Theorem 10 and Theorem 11. Let $C(X, Y, Z) = 0$ be a projective irreducible quadratic equation over a field K. The method provides a parameterisation of all solutions over K from a single solution $P_0 := [x_0 : y_0 : z_0]$ over the same field.

We choose a projective line passing through P_0, namely a set of points P of the form

$$P = P_0 + tS$$

parameterised by some slope $S = [\alpha : \beta : \gamma] \in \mathbb{P}^2(K)$. Intersecting this line with the curve defined by C, we obtain a quadratic equation

$$C^{(S)}(t) := C(P_0 + tS) = C_2^{(S)}t^2 + C_1^{(S)}t + C_0^{(S)} = 0.$$

Because P_0 is a solution, we have $C_0^{(S)} = 0$ and a new solution to $C = 0$ is obtained as $P_1 := P_0 + t_1 S$, with $t_1 := -C_1^{(S)}/C_2^{(S)}$ (where the irreducibility condition on C ensures this is well-defined). These formulae also show that P_1 is defined over K. More generally, almost all solutions to $C = 0$ can be obtained this way and parameterised with S: indeed, any other rational solution P_1 defines a line going through P_0 and P_1, hence a slope S.

The method is well-known to the number theory community, and it should also feel familiar to cryptographers: indeed replacing C by an elliptic curve equation and using lines that intersect at two points (or are tangent to one point) gives the addition law formulae.

5.4 Instances susceptible to Theorem 10

We now focus on Theorem 10. As argued above, it is sufficient to find a prime p, integers N, e, d with e small and d suitably smooth, and quaternion elements $\theta, \theta' \in B_{p,\infty}$ satisfying a corresponding norm equation, i.e.

$$\mathrm{nrd}(\theta) = N^2 e = w^2 + d^2 \,\mathrm{nrd}(\theta') \tag{6}$$

with $\mathrm{tr}(\theta') = 0$.

As mentioned above, this equation is exactly the "lollipop equation" considered in [42,43]. In [43], three methods are given when at least some of these parameters can be chosen. The methods in Algorithm 3 and Section 5.3 there require $N > d^2$, while the method in Section 5.4 can only guarantee that d is "somewhat" smooth. We now provide an alternative approach ensuring $N \approx d$ with d arbitrarily smooth. We recall that N does not need to be smooth.

We arbitrarily fix $e = 1$ and $\theta' = i$, such that Equation 6 becomes $w^2 + d^2 - N^2 = 0$. A trivial solution to this equation is $(w, d, N) = (1, 0, 1)$. Applying the intersection method, solutions can be written in the form

$$\begin{cases} w = -\alpha^2 + \beta^2 - \gamma^2 + 2\alpha\gamma, \\ d = 2\beta(\gamma - \alpha), \\ N = \alpha^2 + \beta^2 + \gamma^2 - 2\alpha\gamma. \end{cases} \tag{7}$$

To obtain a suitably smooth solution of the correct size, one can then, for example, fix $\beta = 1$, then choose an arbitrary smooth even number d,[7] choose γ arbitrarily and let $\alpha = \gamma - d/2$.

Remark 26. We have worked under the simplification that $e = 1$. However, we could have slightly more general parameters by defining e as the square of some smooth factors in the last equality of Equation (7).

[7] Such a d can be sampled to be $\not\equiv 0 \pmod 4$, so it satisfies the mod-4 condition (Definition 9).

5.5 Instances susceptible to Theorem 11

We now show how to compute instances for which Theorem 11 applies. At a high level, we will first fix all parameters p, N, e, d, λ, as well as $M := \deg \theta'_i$ and $n := \mathrm{tr}(\theta'_2\theta'_1)$ to guarantee existence of a solution; this involves a first application of the intersection method. At the end of this first step we also deduce a first quaternion θ_1 with trace w and norm N^2e as required. We then search for a second such quaternion θ_2 that also satisfies $\mathrm{tr}(\theta'_2\theta'_1) = n$. Another application of the intersection method gives us all quaternions with the required norm and trace, parameterised by a slope (α, β, γ). Using this parameterisation with the constraint $n := \mathrm{tr}(\theta'_2\theta'_1)$ leads to a conic in α, β, γ which we explicitly solve, a posteriori justifying our initial parameter choices. We now detail all these steps.

We first fix all integer parameters. We arbitrarily let $e := 1$ as in the previous section. We choose an integer a of size half the desired size for p, and we let $M := 4a^2$. We then choose b odd such that $p := 2M - b^2$ is prime. Note that $p = 3 \bmod 4$ by construction. We also let $n := (p - b^2)/2$. The parameters w, N and d must satisfy the following relation: $w^2 + Md^2 - N^2 = 0$.

A trivial (but undesirable) solution to this equation is $(w, d, N) = (1, 0, 1)$. The intersection method provides parameterised solutions of the form

$$\begin{cases} w = -\alpha^2 + M\beta^2 - \gamma^2 + 2\alpha\gamma, \\ d = 2\beta(\gamma - \alpha), \\ N = \alpha^2 + M\beta^2 + \gamma^2 - 2\alpha\gamma. \end{cases}$$

With our choice of parameters we have $M = 4a^2$; letting $\beta = 1/2$ and α, γ integers of the right size with $\alpha \neq \gamma \bmod 2$ then ensures d is odd and of the right size.

We fix our first quaternion as $\theta_1 := w + d(x_1 i + y_1 j + z_1 k)$, where (x_1, y_1, z_1) is a solution of

$$x^2 + py^2 + pz^2 = M.$$

This can be computed by applying the intersection method starting from the solution $(2a, 0, 0)$. This ensures that $\mathrm{tr}(\theta_1) = w$ and

$$\mathrm{nrd}(\theta_1) = w^2 + d^2(x_1^2 + py_1^2 + pz_1^2) = w^2 + Md^2 = N^2$$

as required.

In order to find a suitable $\theta_2 = w + d(x_2 i + y_2 j + z_2 k)$, we again apply the intersection method, this time for the equation

$$x^2 + py^2 + pz^2 = M$$

and starting from the solution (x_1, y_1, z_1). For the slope (α, β, γ) we get the solution

$$(x_2, y_2, z_2) = (x_1, y_1, z_1) + \frac{t_1}{t_2}(\alpha, \beta, \gamma)$$

with $t_1 := -2(x_1\alpha + py_1\beta + pz_1\gamma)$, $t_2 := \alpha^2 + p\beta^2 + p\gamma^2$.

The condition $n := \mathrm{tr}(\theta_1'\theta_2')$ then leads to the equation

$$n = x_1x_2 + py_1y_2 + pz_1z_2$$

or

$$(M-n)(\alpha^2 + p\beta^2 + p\gamma^2) = 2(x_1\alpha + py_1\beta + pz_1\gamma)^2. \tag{8}$$

To solve this equation, we first put it into diagonal form.

Lemma 27. *Equation (8) is equivalent to*

$$(n^2 - M^2)(x_1^2 + py_1^2)p\alpha'^2 + M\beta'^2 + \gamma'^2 = 0. \tag{9}$$

through an efficiently computable invertible linear change of variables.

Proof. We postpone the proof of Lemma 27 to Appendix E. □

With our choices of parameters, we have $M = 4a^2$, $n^2 = \left(\frac{p-b^2}{2}\right)^2 = (M-b^2)^2$ and $n^2 - M^2 = -(2M - b^2)b^2 = -b^2p$. Equation (9) therefore becomes

$$-(x_1^2 + py_1^2)(bp\alpha')^2 + (2a\beta')^2 + \gamma'^2 = 0 \tag{10}$$

To solve Equation (10), after applying the change of variables

$$\alpha'' = bp\alpha', \quad \beta'' = 2a\beta', \quad \gamma'' = \gamma',$$

we then need to solve the equation $-(x_1^2 + py_1^2)(\alpha'')^2 + (\beta'')^2 + (\gamma'')^2 = 0$. This can be done by fixing α'' as the least common multiple of the denominators of x_1 and y_1, then solving $(\beta'')^2 + (\gamma'')^2 = (x_1^2 + py_1^2)(\alpha'')^2$ via Cornacchia (and repeating with different values of x_1, y_1 if needed). Inverting the linear transformation of Lemma 27 and the linear transformation above then gives a solution to Equation (8), and a second quaternion θ_2 which by construction has trace w and norm N^2.

We have $\mathrm{tr}(\theta_1\theta_2) = w^2 + d^2n$. By our choices of parameters this is an integer, and moreover $\mathrm{tr}(\theta_1\theta_2) \neq \pm 2N^2$. By Lemma 25, we deduce that $\Lambda = \mathbb{Z}[\theta_1, \theta_2]$ is a full-rank order of $B_{p,\infty}$. We then compute the discriminant of Λ, factor it, and compute a maximal order $\mathcal{O}_0 \supset \Lambda$ using [48]. We finally compute E and an explicit isomorphism $\mathrm{End}(E) \simeq \mathcal{O}$.

An implementation of the algorithm to generate weak instances for Theorem 11 can be found at

https://github.com/LucianoMaino/Pushforward-Problems-and-Applications-to-Isogeny-based-Cryptography

Remark 28. Remember that $N^2 = w^2 + Md^2$ by definition. With our choice of parameters, p and M are about the same size, so $N \approx p^{1/2}d$, preventing us from having d and N of about the same size. This limitation seems hard to avoid. Indeed, writing $\theta_1 = w + d(ix_1 + jy_1 + ijz_1) =: w + d\theta_1'$ and $\theta_2 = w + d(ix_2 + jy_2 + ijz_2) =: w + d\theta_2'$, we have that θ_1 and θ_2 do not commute if

and only if θ_1' and θ_2' do not commute. Also $\mathrm{nrd}(\theta_i') = M$ by definition. By [32, Proposition 5], based on [27, Theorem 2'], we have that $\mathrm{nrd}(\theta_1')\,\mathrm{nrd}(\theta_2') \gtrapprox p^2$, which implies that either $\mathrm{nrd}(\theta_1') \gtrapprox p$ or $\mathrm{nrd}(\theta_2') \gtrapprox p$. This shows that our method, which assumes $\mathrm{tr}(\theta_1) = \mathrm{tr}(\theta_2)$, produces results that are, in a certain sense, optimal. The only hope to get more balanced d and N is to find a new method that does not require $\mathrm{tr}(\theta_1) = \mathrm{tr}(\theta_2)$.

6 Applications

In this section, we consider the impact of our findings on existing isogeny-based constructions. We first briefly remark that we can break the non-interactive key exchange in [30] without using any quantum polynomial-time algorithms. Then, we give a concrete attack against the recent threshold signature scheme proposed by Kim, Kim and Lee [28]. We also show that the Oblivious Pseudorandom Function in [2] needs to be instantiated using a curve of unknown endomorphism ring, contrary to what was claimed in the paper. Finally, we refine the security analysis of the updatable encryption in [34]; this does not give an attack on [34] for the current parameter choice.

6.1 pSIDH

In [30], Leroux describes a non-interactive key exchange, called pSIDH, where secret keys consist of large prime-degree isogenies $\varphi : E_0 \to E$ and public keys are given by E together with a full-rank suborder of $\mathrm{End}(E)$ obtained by φ; here, the curve E_0 is a curve of known endomorphism ring. The pSIDH protocol was shown to be vulnerable to a quantum-polynomial attack in [15]. The attack recovers the full endomorphism ring of E starting from the information given by the subring.

It is worth observing that in pSIDH, to recover the shared secret (and in turn effectively break the scheme), it is enough to recover an isogeny representation of the secret isogeny φ. Indeed, given isogenies representations of the two secret isogenies, we can easily recover the shared secret by computing the pushforward of one isogeny under the other one [45, Prop. 6.15]. This breaks [30, Problem 4].

We remark that the technique we employed in Section 5.5 utilises the push-forward oracle to construct a full-rank suborder of the endomorphism ring on the codomain curve. Then, from this full-rank suborder we recover the image of some torsion points under the secret isogeny. Since in pSIDH we already have the suborder information, we can already recover the image of some torsion points. This gives us an isogeny representation.

In [30, §4.5], Leroux already pointed out that it was possible to recover torsion information from the full-rank suborder. However, the conclusion was that the lollipop attacks could not have been applied simply because the kernel generators of the secret isogenies were not representable. This remark further strengthens the main novelty of our attacks.

Remark 29. Something worth remarking is that our techniques only allow us to recover an isogeny representation of the secret isogeny. On the other hand, the attack in [15] is significantly more powerful since it recovers the endomorphism ring of the codomain curve. Being able to recover the endomorphism ring of the codomain curve in polynomial time from an isogeny representation without using quantum algorithms does not seem to be easy, see *e.g.* [12, Problem 6].

6.2 Threshold PRISM

Very recently, Kim, Kim and Lee introduce a threshold signature scheme [28] based on PRISM [3]. In PRISM, a secret key consists in an isogeny $\varphi : E_0 \to E$, where E_0 is the curve $E_0 : y^2 = x^3 + x$. The key idea in Threshold PRISM is to split the secret isogeny φ into a chain of isogenies $\varphi_i : E_{i-1} \to E_i$ of degree 3^ν. To be more precise, the secret isogeny $\varphi : E_0 \to E$ is given as $\varphi = \varphi_t \circ \ldots \circ \varphi_1 : E_0 \to E_1 \to \ldots \to E_{t-1} \to E_t = E$, where each isogeny $\varphi_i : E_{i-1} \to E_i$ is a secret share.

To sign messages, each of the parties involved in the signing procedure computes the pushforward of an isogeny of special degree. A message m corresponds to a large prime q such that $2^a - q$ is also a prime; here a is a fixed parameter. The first party, holding the secret share $\varphi_1 : E_0 \to E_1$, computes any isogeny $\sigma_1 : E_1 \to E_1^*$ of degree $q(2^a - q)$; this can be easily achieved because the endomorphism ring of E_0 is known. The second party, holding the secret share $\varphi_2 : E_1 \to E_2$, receives $\sigma_1 : E_1 \to E_1^*$ and computes $\sigma_2 := [\varphi_2]_* \sigma_1 : E_2 \to E_2^*$. This procedure is repeated for all the signing parties until the last party, the t-th party, computes the final signature $\sigma := [\varphi_t]_* \sigma_{t-1} : E_t \to E_t^*$.

When the shares σ_i are public, it is clear that each of the parties has access to something resembling a pushforward oracle. More specifically, the first party holding the secret share $\varphi_1 : E_0 \to E_1$ can build a pushforward oracle for φ_2. Indeed, since $\deg(\varphi_1) = 3^\nu$, the first party can also compute a representation of the endomorphism ring $\mathrm{End}(E_1)$. Now, assume that all the parties have to sign some messages, say $\mathsf{m}_1, \ldots, \mathsf{m}_n$, each of them corresponding to degrees $q_1, \ldots, q_n$. This means that the first party can query the pushforward of any isogeny $\sigma_j : E_1 \to E_1^*$ of degree $q_j(2^a - q_j)$, for $j = 1, \ldots, n$. [8]

Define $N := \prod_{j=1}^n q_j(2^a - q_j)$. Since the $q_j(2^a - q_j)$ are pairwise coprime, the first party can build a pushforward oracle $\mathcal{O}^{\mathrm{PUSH}}_{\varphi_2, N}$ as follows. On input an isogeny $\psi : E_1 \to E_1^*$ of degree N, the first party computes a representation of the isogeny ψ_j such that $\ker(\psi_j) := \ker(\psi) \cap E_1[q_j(2^a - q_j)]$; this can be efficiently done as explained in [45, Proposition 6.11]. The second party will compute $\psi'_j := [\varphi_2]_* \psi_j$. The first party can reconstruct $[\varphi_2]_* \psi$ by computing $[\psi'_n]_* [\psi'_{n-1}]_* \ldots \psi'_1$ iteratively; this can be efficiently done as explained in [45, Proposition 6.15].

Equipped with pushforward oracle $\mathcal{O}^{\mathrm{PUSH}}_{\varphi_2, N}$, the first party can recover the secret share φ_2. Since $\deg(\varphi_2) = 3^\nu$, following Theorem 10, it suffices to find an

[8] It is worth remarking that it is possible to sign the same message m several times with distinct isogenies of the same degree.

endomorphism $\theta \in \mathrm{End}(E_1)$ such that $\theta_{|E_1[3^\nu]} = [a]_{|E_1[3^\nu]}$ for some $a \in \mathbb{Z}/3^\nu\mathbb{Z}$, $\theta \in \mathrm{End}(E_1) \setminus \mathbb{Z}$ and $\deg(\theta) = N^2$.[9] It is worth noting that, depending on the number n of messages $\mathsf{m}_1, \ldots, \mathsf{m}_n$, we could obtain $N \gg 3^\nu$. For the parameter sets chosen in [28], a modest number n could even suffice.

We now briefly explain how to generate such a θ. Assume[10] that $N^2 > 3^{4\nu}p^2$. Applying [20, Algorithm 12], we can compute a $\theta_0 \in \mathbb{Z}+[3^{2\nu}]\,\mathrm{End}(E_0)$ of norm N^2, which corresponds to $\theta \in \mathrm{End}(E_1) \setminus \mathbb{Z}$ of norm N^2 such that $\theta_{|E_1[3^\nu]} = [a]_{|E_1[3^\nu]}$. To be more precise $\theta = [\varphi_1]_* \theta_0$.[11]

This effectively shows that [28, Problem 2] does not hold. However, this attack goes beyond breaking [28, Problem 2]; it breaks Problem 30.[12]

Problem 30. Let a, ν and p as above, and let $q_1, \ldots, q_n$ be some primes such that $2^a - q_j$ is a prime and $\left(\prod_{j=1}^n q_j(2^a - q_j)\right)^2 > p^3 3^{2\nu}$. Let $\varphi = \varphi_t \circ \ldots \circ \varphi_1 : E_0 \to E_1 \to \ldots \to E_{t-1} \to E_t = E$ be an isogeny such that $\deg(\varphi_i) = 3^\nu$. Assume that, on input an isogeny $\sigma_j : E_0 \to E_0^*$ of degree $q_j(2^a - q_j)$, an adversary is given $(\sigma_j^{(1)} := [\varphi_1]_* \sigma_j, \sigma_j^{(2)} := [\varphi_2]_* \sigma_j^{(1)}, \ldots, \sigma_j^{(t)} := [\varphi_t]_* \sigma_j^{(t-1)})$. The problem asks to recover φ.

Once the first party recovers φ_2, they can reiterate the same reasoning for all the remaining secret shares $\varphi_3, \ldots, \varphi_t$. Given φ_2, the first party can retrieve $\mathrm{End}(E_2)$. From the knowledge of $\mathrm{End}(E_2)$, they can compute an isogeny $\tau : E_0 \to E_2$ of degree $\approx \sqrt{p}$. If $N^2 > p^2 \cdot \deg(\tau)^2 3^{2\nu} \approx p^3 3^{2\nu}$, the first party can repeat the same reasoning using τ in lieu of φ_1 to get a $\theta_2 \in \mathrm{End}(E_2) \setminus \mathbb{Z}$ such that $\theta_{|E_2[3^\nu]} = [a]_{|E_2[3^\nu]}$ for some $a \in \mathbb{Z}/3^\nu\mathbb{Z}$ and $\deg(\theta) = N^2$, and then recover φ_3. Proceeding similarly, they can retrieve all the remaining secret shares $\varphi_4, \ldots, \varphi_t$.

The attacks described so far assumed that the shares σ_i were publicly available to the attacker, whereas shares could also be privately passed by one party to the next one. We remark that in that case, any set of $t-1$ colluding parties can collaborate to recover all shares, thereby also breaking the security properties of the threshold scheme.

Countermeasures. To the best of our knowledge, there is no easy countermeasure to this attack. One possible solution could be to redesign the construction to hide the degree of the secret isogenies. This approach was already followed in [10]. In [10], the authors presented a different PRISM-based threshold signature

[9] We remark that it is not necessary to check the mod-4 condition (Definition 9) in this case since the degree of the targeted isogeny is odd.

[10] Below, we will show that it suffices to consider $N^2 > 3^{2\nu}p^3$; the quantity $3^{2\nu}p^3$ is $< 3^{4\nu}p^2$.

[11] Recall that the first party knows the isogeny φ_1.

[12] We remark that [28, Problem 2] should hold for all the structures $(b_1, \ldots, b_N)$; see [28, Theorem 4.3]. The formulation of [28, Problem 2] is rather technical. For the sake of simplicity, we consider Problem 30, which can be considered a coarse generalisation of the computational version of [28, Problem 2].

scheme where each secret share is an isogeny of secret degree. The attacks we presented in Section 4 do not seem to affect this construction.

6.3 Basso's Oblivious Pseudorandom Function

Here, we describe how to create backdoor instances for Basso's Oblivious Pseudorandom Function (OPRF) [2]. In [2], Basso adapts the construction in [9] to design an OPRF that is resistant to the attack described in [6]. For the sake of conciseness, we do not recall Basso's construction; rather we highlight that its security depends on [2, Problem 5], which can be turned into an instance of Problem 5.

Remark 31. We highlight that [2, Problem 5] is used to prove [2, Theorem 11]. The proof is only sketched, and it is claimed to follow from [9, Theorem 20]. However, we highlight that differently from [9], [2, Problem 5] does not mention that an adversary has access to some torsion information. In the case of Basso's OPRF, this torsion information is masked by the same scalar. Hence, to take this missing information into account, we will assume that the adversary can query the pushforward oracle with isogenies of degree equal to the product of the degree of the message isogeny times the degree of the blinding isogeny; we refer the reader to [2, Fig. 1] for alignment with Basso's construction.

From our perspective, this does not change anything about the attack we will consider. Following the notation in Problem 5, the quantity N is larger than what would be needed if we only had to attack [2, Theorem 11]. This gives a further advantage when looking for weak instances.

In [2, Problem 5], the pushforward isogeny is not revealed, rather only the codomain of the pushforward isogeny is output. However, in [2, §4], Basso states that we can assume that this OPRF is subject to the attack in [6] as well. This implies that we can work under the assumption that we can recover the pushforward of queried isogenies, potentially incurring a subexponential cost. In other words, we can instantiate a pushforward oracle.

We now translate [2, Problem 5] to our language. We have a cyclic isogeny $\varphi : E_0 \to E$ of degree d and a pushforward oracle $\mathcal{O}^{\mathrm{PUSH}}_{\varphi,N}$, where the N-torsion is not representable. Since Basso's idea to prevent the SIDH attacks is to use the countermeasures in M-SIDH [25], he chooses the degree $d = \prod_{i=1}^{m} \ell_i$ for some small distinct primes ℓ_i's. In particular, as explained in [2, §7], we have that $N \gg d$.[13] Additionally, in [2, §7], paragraph "Parameter selection", Basso states that the starting curve E_0 can have known endomorphism ring.

Under these conditions, we can compute a backdoor instance as explained in Section 5.4, *i.e.*, we can compute an elliptic curve E_0 satisfying the hypotheses in Theorem 10.[14] In Section 5.4, it is explained that N can be arbitrary. However,

[13] Following the notation in [2, §7], we have $d = N_K \approx N_B$ and $N = \deg(\phi_m) \cdot N_B \approx 2^{4\lambda} \cdot d$, where λ denotes the security parameter. Additionally, it is not necessary to check the mod-4 condition (Definition 9) since $d \neq 0 \pmod 4$.

[14] This backdoor can be used to allow users to retrieve the server's secret key.

in our case, we also want N to be smooth. It is possible to craft such parameters d and N at the cost of searching among the solutions parametrised in Equation (7) and attacking an isogeny $\varphi \circ \widetilde{\varphi}$, where $\deg(\varphi \circ \widetilde{\varphi})$ can be used as d in Equation (7) and $\widetilde{\varphi}$ is an additional isogeny an attacker computes.

One possible way to counter this attack could be to use an elliptic curve E_0 of unknown endomorphism ring. This could be done using a multiparty trusted setup ceremony, *e.g.* [4]. While this procedure can be computationally expensive, we highlight that this ceremony must be run only once, during the setup phase. Another (more risky and expensive) countermeasure could be to force the server to use secret isogenies significantly longer than the blinding isogenies such that the information provided by the pushforward oracle is not enough to run any of our attacks. We recommend that a robust implementation of Basso's OPRF should use a starting curve E_0 of unknown endomorphism ring.

6.4 Leroux and Roméas's Updatable Encryption

In [34], Leroux and Roméas introduce an updatable encryption scheme whose algorithm to update ciphertexts relies on the pushforward of large-prime-degree isogenies. For the sake of conciseness, we do not recall Leroux and Roméas's construction. In [34, §4.1], the authors argue that the security of their construction depends on the fact that there seems to be no obvious way to meaningfully exploit the additional information provided by pushforwards of large-prime-degree isogenies. In this section, we will not give a concrete attack against Leroux and Roméas's Updatable Encryption; rather, we will highlight that one of the assumptions used by the authors does not hold in full generality (though it might hold for their choice of parameters).

In [34, Theorem 3], the deterministic indistinguishability under chosen plaintext attacks of their construction is proved under the assumption that the advantage in [34, Definition 8] is negligible; we briefly recall this assumption, adapting it to our notation.

Assumption 32. Let E be a supersingular curve over $\mathbb{F}_{p^2}$ and let N and d be two coprime integers such that N is prime and the N-torsion is not representable. Consider a secret bit $b \in \{0, 1\}$. The adversary $\mathcal{A}$ receives a supersingular curve E' defined as follows:

- If $b = 0$, there exists an isogeny $\varphi : E \to E'$ of degree d;
- If $b = 1$, the curve E' is uniformly random.

Define $\sigma_0 := \mathcal{O}^{\text{PUSH}}_{\varphi,N}$ and σ_1 a uniformly random bijection between sets of N-isogenies with domain E and E', respectively.

The adversary $\mathcal{A}$ is given oracle access to σ_b and outputs a bit $\tilde{b}$. We assume that the adversary outputs $\tilde{b} = b$ with negligible probability over $1/2$.

In [34], there is no mention whether the curve E in Assumption 32 has known endomorphism ring. However, in the proof of [34, Theorem 3], it is assumed that

an adversary $\mathcal{A}$ does know the endomorphism ring of E. Hence, we will assume that $\mathrm{End}(E)$ is known.

It is worth observing that Assumption 32 $\xRightarrow{\text{reduces to}}$ Problem 6 $\xRightarrow{\text{reduces to}}$ Problem 3. As we showed in Section 5.5, there are instances susceptible to Theorem 11. However, as mentioned in Remark 28, our techniques require $N \gtrapprox p^{1/2}d$.

In [34], the authors suggest using a prime $d := 2^f - N$, for some $f > 0$ such that $d > 0$. For this reason, to the best of our knowledge, it does not seem easy to find a curve E susceptible to Theorem 11.[15] The situation would have been different if d had not been prime since for a smooth d we could hope to have $N \approx d$ (see Section 5.4). To conclude, our analysis supports the intuition mentioned by the authors in [34, §6.1]: picking $d = 2^f - N$ to be prime seems to make Assumption 32 harder.

7 Future directions and recommendations

Isogeny pushforwards are an extremely common technique in isogeny-based cryptography. Indeed, as already hinted in Section 3, even the security of SIDH [19,26] relied on a (weak) instance of a pushforward oracle problem. Another more recent example of the pushforward usage can be found in the SQIsign2D variants [5,40,22,51,39]. In particular, to generate a response, the signer has to compute the pushforward of an (auxiliary) isogeny under the commitment isogeny. Each SQIsign2D variant does this in a different way.

To the best of our knowledge, it is not possible to instantiate any pushforward oracle from any of the SQIsign2D variants. However, depending on how these auxiliary isogenies are generated, it might be possible to instantiate pushforward oracles via side-channel or fault-injection attacks. Here, we remark that we do not know how to instantiate such attacks, but we point out that this could become a potential attack avenue in the future.

Based on our analysis, we give some recommendations for designing protocols that reveal pushforward oracles. First and foremost, the best countermeasure against our attacks is to hide the endomorphism ring of the domain curve. However, we appreciate that this design choice might not always be available.

We also suggest hiding the degree of the isogeny to which the pushforwards are applied to. Indeed, to the best of our knowledge it is not possible to generalise Theorems 10 and 11 to the case of unknown degree. Finally, we recommend that designers limit the amount of torsion information revealed by the pushforward oracle in favour of longer secret isogenies. As we have shown in this paper, even if the torsion information is not representable, it can still provide additional information that could be exploited to reveal secret information.

[15] This is because $d > 2^f > N$. Therefore, it is not possible that $N \gtrapprox p^{1/2}d$.

References

1. Adj, G., Cervantes-Vázquez, D., Chi-Domínguez, J.J., Menezes, A., Rodríguez-Henríquez, F.: On the cost of computing isogenies between supersingular elliptic curves. In: Cid, C., Jacobson, Jr., M.J. (eds.) SAC 2018: 25th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 11349, pp. 322–343. Springer, Cham, Switzerland, Calgary, AB, Canada (Aug 15–17, 2019). https://doi.org/10.1007/978-3-030-10970-7_15 8, 13
2. Basso, A.: A post-quantum round-optimal oblivious PRF from isogenies. In: Carlet, C., Mandal, K., Rijmen, V. (eds.) SAC 2023: 30th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 14201, pp. 147–168. Springer, Cham, Switzerland, Fredericton, Canada (Aug 14-18, 2024). https://doi.org/10.1007/978-3-031-53368-6_8 3, 4, 25, 28
3. Basso, A., Borin, G., Castryck, W., Corte-Real Santos, M., Invernizzi, R., Leroux, A., Maino, L., Vercauteren, F., Wesolowski, B.: PRISM: Simple and compact identification and signatures from large prime degree isogenies. In: Jager, T., Pan, J. (eds.) PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 15676, pp. 300–332. Springer, Cham, Switzerland, Røros, Norway (May 12–15, 2025). https://doi.org/10.1007/978-3-031-91826-1_10 26
4. Basso, A., Codogni, G., Connolly, D., De Feo, L., Fouotsa, T.B., Lido, G.M., Morrison, T., Panny, L., Patranabis, S., Wesolowski, B.: Supersingular curves you can trust. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part II. Lecture Notes in Computer Science, vol. 14005, pp. 405–437. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30617-4_14 29
5. Basso, A., Dartois, P., De Feo, L., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-West - the fast, the small, and the safer. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 339–370. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_11 6, 30
6. Basso, A., Kutas, P., Merz, S.P., Petit, C., Sanso, A.: Cryptanalysis of an oblivious PRF from supersingular isogenies. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 160–184. Springer, Cham, Switzerland, Singapore (Dec 6–10, 2021). https://doi.org/10.1007/978-3-030-92062-3_6 9, 28
7. Basso, A., Maino, L.: POKÉ: A compact and efficient PKE from higher-dimensional isogenies. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 94–123. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91124-8_4 2
8. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 98–126. Springer, Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8739-9_4 2
9. Boneh, D., Kogan, D., Woo, K.: Oblivious pseudorandom functions from isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 520–550. Springer,

Cham, Switzerland, Daejeon, South Korea (Dec 7–11, 2020). https://doi.org/10.1007/978-3-030-64834-3_18 9, 28
10. Borin, G.: Compact threshold signatures from pushforwards of large-degree isogenies. https://csrc.nist.gov/csrc/media/presentations/2026/mpts2026-4a2/images-media/mpts2026-4a2-slides-prism-isogenies-borin.pdf, presentation given at https://csrc.nist.gov/presentations/2026/mpts2026-4a2, Accessed: 18th May, 2026 27
11. Cai, S., Chen, M., Lai, Y.F., Lin, K.: Pike: Faster isogeny-based public key encryption with pairing-assisted decryption. In: Bai, S., Persichetti, E. (eds.) Public-Key Cryptography – PKC 2026. pp. 58–91. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-26737-5_3 2
12. Castryck, W., De Feo, L., Galbraith, S.D., Kutas, P., Reijnders, K., Wesolowski, B.: The isogeny problems. Cryptology ePrint Archive, Paper 2026/1431 (2026), https://eprint.iacr.org/2026/1431 26
13. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 423–447. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_15 2, 6
14. Castryck, W., Vercauteren, F.: A polynomial time attack on instances of M-SIDH and FESTA. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 127–156. Springer, Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8739-9_5 7, 9
15. Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part III. Lecture Notes in Computer Science, vol. 14440, pp. 99–130. Springer, Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8727-6_4 25, 26
16. Cremona, J.E., Rusin, D.: Efficient solution of rational conics. Mathematics of Computation **72**(243), 1417–1441 (2003) 21
17. De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017) 5
18. De Feo, L., Fouotsa, T.B., Panny, L.: Isogeny problems with level structure. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part VII. Lecture Notes in Computer Science, vol. 14657, pp. 181–204. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58754-2_7 2, 10, 11
19. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Cryptology ePrint Archive, Report 2011/506 (2011), https://eprint.iacr.org/2011/506 2, 30
20. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer, Cham, Switzerland, Daejeon, South Korea (Dec 7–11, 2020). https://doi.org/10.1007/978-3-030-64837-4_3 27
21. Deuring, M.: Die typen der multiplikatorenringe elliptischer funktionenkörper: G. herglotz zum 60. geburtstag gewidmet. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg. vol. 14, pp. 197–272. Springer (1941) 6

22. Duparc, M., Fouotsa, T.B.: SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 396–429. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_13 30
23. Eisenträger, K., Hallgren, S., Lauter, K.E., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 329–368. Springer, Cham, Switzerland, Tel Aviv, Israel (Apr 29 – May 3, 2018). https://doi.org/10.1007/978-3-319-78372-7_11 6, 20
24. Eriksen, J., Panny, L., Sotáková, J., Veroni, M.: Deuring for the people: Supersingular elliptic curves with prescribed endomorphism ring in general characteristic, pp. 339–373 (01 2024). https://doi.org/10.1090/conm/796/16008 2
25. Fouotsa, T.B., Moriya, T., Petit, C.: M-SIDH and MD-SIDH: Countering SIDH attacks by masking information. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 282–309. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_10 2, 9, 10, 28
26. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011. pp. 19–34. Springer Berlin Heidelberg, Germany, Tapei, Taiwan (Nov 29 – Dec 2 2011). https://doi.org/10.1007/978-3-642-25405-5_2 2, 30
27. Kaneko, M.: Supersingular j-invariants as singular moduli mod p. Osaka Journal of Mathematics **26**, 849–855 (1989) 25
28. Kim, H., Kim, W., Lee, C.: Threshold PRISM signature schemes via graph-based threshold access structures. Cryptology ePrint Archive, Paper 2026/945 (2026), https://eprint.iacr.org/2026/945 3, 4, 25, 26, 27
29. Kohel, D., Lauter, K.E., Petit, C., Tignol, J.: On the quaternion ℓ-isogeny path problem. LMS J. Comput. Math. **17**(Theory), 418–432 (2014). https://doi.org/10.1112/S1461157014000151 6, 20
30. Leroux, A.: A new isogeny representation and applications to cryptography. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022, Part II. Lecture Notes in Computer Science, vol. 13792, pp. 3–35. Springer, Cham, Switzerland, Taipei, Taiwan (Dec 5–9, 2022). https://doi.org/10.1007/978-3-031-22966-4_1 25
31. Leroux, A.: Quaternion algebras and isogeny-based cryptography. PhD thesis (September 2022), https://www.lix.polytechnique.fr/page/index.php?username=aleroux&path=manuscrit_these.pdf 6
32. Leroux, A.: An effective lower bound on the number of orientable supersingular elliptic curves. In: Smith, B., Wu, H. (eds.) SAC 2022: 29th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 13742, pp. 263–281. Springer, Cham, Switzerland, Windsor, Canada (Aug 24-26, 2024). https://doi.org/10.1007/978-3-031-58411-4_12 25
33. Leroux, A.: Verifiable random function from the deuring correspondence and higher dimensional isogenies. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part VII. Lecture Notes in Computer Science, vol. 15607, pp. 167–194. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91098-2_7 10, 11

34. Leroux, A., Roméas, M.: Compact, efficient and CCA-secure updatable encryption from isogenies. Cryptology ePrint Archive, Report 2025/1853 (2025), https://eprint.iacr.org/2025/1853 3, 4, 25, 29, 30
35. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 448–471. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_16 2, 6
36. Merdy, A.H.L., Wesolowski, B.: The supersingular endomorphism ring problem given one endomorphism. IACR Communications in Cryptology (CiC) **2**(1), 6 (2025). https://doi.org/10.62056/akgyivrzn 13, 14, 16
37. Moriya, T., Stopar, M.: LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 275–306. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_9 2
38. Nakagawa, K., Onuki, H.: QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part V. Lecture Notes in Computer Science, vol. 14924, pp. 75–106. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68388-6_4 2
39. Nakagawa, K., Onuki, H.: SQIsign2DPush: Faster Signature Scheme Using 2-Dimensional Isogenies. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 511–540. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-25327-9_18 30
40. Nakagawa, K., Onuki, H., Castryck, W., Chen, M., Invernizzi, R., Lorenzon, G., Vercauteren, F.: SQIsign2D-East: A new signature scheme using 2-dimensional isogenies. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 272–303. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_9 30
41. Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part VI. Lecture Notes in Computer Science, vol. 14656, pp. 388–417. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58751-1_14 6
42. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 330–353. Springer, Cham, Switzerland, Hong Kong, China (Dec 3–7, 2017). https://doi.org/10.1007/978-3-319-70697-9_12 3, 4, 7, 22
43. de Quehen, V., Kutas, P., Leonardi, C., Martindale, C., Panny, L., Petit, C., Stange, K.E.: Improved torsion-point attacks on SIDH variants. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 432–470. Springer, Cham, Switzerland, Virtual Event (Aug 16–20, 2021). https://doi.org/10.1007/978-3-030-84252-9_15 3, 4, 7, 20, 22
44. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer

Science, vol. 14008, pp. 472–503. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_17 2, 6, 9, 19
45. Robert, D.: On the Efficient Representation of Isogenies: A Survey for NuTMiC 2024. In: International Conference on Number-Theoretic Methods in Cryptology. pp. 3–84. Springer (2024). https://doi.org/10.1007/978-3-031-82380-0_1 6, 19, 25, 26
46. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009) 5
47. The online encyclopedia of integer sequences: A002110. https://oeis.org/A002110, accessed: 16th May, 2026 35
48. Voight, J.: Identifying the Matrix Ring: Algorithms for Quaternion Algebras and Quadratic Forms, pp. 255–298. Springer New York, New York, NY (2013). https://doi.org/10.1007/978-1-4614-7488-3_10 24
49. Voight, J.: Quaternion Algebras, Graduate Texts in Mathematics, vol. 288. Springer Cham (2021). https://doi.org/10.1007/978-3-030-56694-4 18
50. Wesolowski, B.: The supersingular isogeny path and endomorphism ring problems are equivalent. In: 62nd Annual Symposium on Foundations of Computer Science. pp. 1100–1111. IEEE Computer Society Press, Denver, CO, USA (Feb 7–10, 2022). https://doi.org/10.1109/FOCS52979.2021.00109 6, 20
51. Xu, Z., Lin, K., Zhao, C.A., Ouyang, Y.: SQIsign2D^2: New SQIsign2D variant by leveraging power smooth isogenies in dimension one. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 341–371. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_11 30

A Estimating the size of $\mathcal{S}$

In this section, we estimate the size of $\mathcal{S}$ of Theorem 11 in the most general case. Our goal is to prove Corollary 12, which we recall below.

Corollary 33. *Let $N, d \in O(\mathsf{poly}(p))$ be two coprime positive integers. Also, let E be a supersingular elliptic curve of known endomorphism ring $\mathrm{End}(E)$ and let $\theta_1, \theta_2 \in \mathrm{End}(E)$ such that $\theta_{i|E[d]} = [a_i]_{|E[d]}$ for some $a_i \in \mathbb{Z}/d\mathbb{Z}$ and $\deg(\theta_i) = N^2 e_i$, for some smooth $e_i \in O(\mathsf{poly}(p))$.*

Consider a d-isogeny $\varphi : E \to E'$. Then, given oracle access to $\mathcal{O}^{PUSH}_{\varphi,N}$, there exists an algorithm to solve Problem 3 whose complexity is

$$O\left(\max\left\{2^{\frac{\ln(d)}{\ln\left(\ln(d^{1/2}\cdot N^4\cdot e_1\cdot e_2)\right)}}, \sqrt{e_1}, \sqrt{e_2}\right\}\cdot \mathsf{poly}(\log(p))\right).$$

More precisely, the algorithm only requires four calls to $\mathcal{O}^{PUSH}_{\varphi,N}$.

Proof. Let q_k represent the k-th prime number. We first recall the following two facts.

1. For an integer n, we define $s_n := \min_k\left\{\prod_{i=1}^{k} q_k > n\right\}$. Following the estimates in [47], we have that $n \lessapprox e^{(1+o(1))s_n \ln(s_n)}$. A rough estimate we are going to use is $s_n \gtrapprox \frac{\ln(n)}{\ln(\ln(n))}$.

2. We approximate $q_k \approx k\ln(k)$. This is a consequence of the prime number theorem.

We estimate the size of a set $\mathcal{S}$ to use in Theorem 11 by looking at the number of minimal primes coprime to $r := \deg(\theta_1\theta_2 - \theta_2\theta_1)$ such that their product is greater than $\sqrt{d}$. In particular, such a quantity is upper bounded by $s_{r\sqrt{d}} - s_r \approx \frac{\ln(r\sqrt{d})}{\ln(\ln(r\sqrt{d}))} - \frac{\ln(r)}{\ln(\ln(r))}$, which can be roughly estimated to $\frac{\ln(d)}{2\ln(\ln(r\cdot d^{1/2}))}$. It is worth noting that the largest prime defining $s_{r\sqrt{d}}$ is $\approx \frac{\ln(r\sqrt{d})}{\ln(\ln(r\sqrt{d}))}$.

Since $e_1, e_2 \in O(\mathsf{poly}(p))$, we obtain that $r \in O(\mathsf{poly}(p))$. In fact $r = \deg(\theta_1\theta_2 - \theta_2\theta_1) \leq 4\deg(\theta_1)\deg(\theta_2) = 4N^4e_1e_2$. As a consequence, the largest prime defining $s_{r\sqrt{d}}$ is in $O(\mathsf{poly}(\log(p)))$. In conclusion, we have constructed an $\mathcal{S}$ of size in $O\left(\frac{\ln(d)}{\ln(\ln(d^{1/2}\cdot N^4\cdot e_1\cdot e_2))}\right)$. □

B Proof of Lemma 20

In this section, we give a proof of Lemma 20, which we recall in what follows.

Lemma 34. *Let $k > 0$ be an exponent such that $2^k \mid d$ and $([\varphi]_*[d]\alpha_0)_{|E[2^k]} = [0]_{|E[2^k]}$. Then, there exists a polynomial-time algorithm to compute at most four 2^k-isogenies; one of those is the isogeny with kernel $\ker(\varphi)[2^k]$.*

Proof. We first generalise Lemma 18 for $\ell = 2, 4$: we will prove that there exist at most three subgroups K_1, K_2 and K_3 of order ℓ such that $\ker(\varphi)[\ell] = K_i$, for some $i = 1, 2$ or 3, and that these groups can be computed in polynomial time in $\log(p)$.

If $\ell = 2$, there are at most three subgroups of order two fixed by α_0; such subgroups can be efficiently computed as they correspond to the points of order 2. If $\ell = 4$, we show that, assuming that θ satisfies the mod-4 condition (Definition 9), α_0 fixes exactly two subgroups, hence giving a direct generalisation of Lemma 18.

Similarly to Lemma 18, we fix a basis (P, Q) of the 4-torsion and then identify $\alpha_{0|E[4]}$ with a matrix $\mathbf{M} = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{M}_{2,2}(\mathbb{Z}/4\mathbb{Z})$, i.e., $\alpha_0(P) = [a]P + [b]Q$ and $\alpha_0(Q) = [c]P + [d]Q$; the quantities a, b, c, d can be computed in polynomial time. In particular $\mathrm{tr}(\mathbf{M}) = 0 \pmod 4$, and $\det(\mathbf{M}) \neq 3 \bmod 4$ by the mod-4 condition.

Lemma 17 implies that there exists at least one subgroup of order 4 fixed by $\mathbf{M}$. We remark that there are six distinct subgroups of order 4 in $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$. A trace-zero matrix in $\mathrm{M}_{2,2}(\mathbb{Z}/4\mathbb{Z})$ could potentially fix all of them. Via a direct computation of all the possible trace-zero matrices in $\mathrm{M}_{2,2}(\mathbb{Z}/4\mathbb{Z})$, we get that the only matrices fixing all 6 subgroups are $\left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right)$; the matrix $\mathbf{M}$ cannot coincide with any of them since $\alpha_{0|E[2]} \neq [0]_{|E[2]}$. Additionally, there is no matrix fixing exactly 5 subgroups.

Similarly, via a direct computation, we have that all the matrices fixing exactly four subgroups are

$$\left(\begin{smallmatrix} 1 & 0 \\ 0 & 3 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 2 & 3 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 2 \\ 0 & 3 \end{smallmatrix}\right), \left(\begin{smallmatrix} 3 & 0 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 3 & 0 \\ 2 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 3 & 2 \\ 0 & 1 \end{smallmatrix}\right);$$

the matrix $\mathbf{M}$ cannot coincide with any of them since $\det(\mathbf{M}) \neq 3 \pmod 4$. Also, there are no matrices fixing exactly either 3 subgroups or 1 subgroup. This means that the matrix $\mathbf{M}$ fixes exactly two subgroups as claimed. Moreover, computing such subgroups can be performed in polynomial time in $\log(p)$ since we are only working with points of order 4.

We now generalise Lemma 19. Suppose that k, the exponent such that $2^k \mid d$ and $([\varphi]_* [d]\alpha_0)_{|E'[2^k]} = [0]_{|E'[2^k]}$, is even. In this case, the proof of Lemma 19 can be perfectly reproduced considering steps of 4-isogenies and the generalisation of Lemma 18 shown above.

Suppose that k is odd. We repeat the reasoning above for $k-1$, which is even. This leads us to up to two 2^{k-1}-isogenies. At the end, however, we need to compute an additional 2-isogeny for each of the 2^{k-1}-isogenies. Applying the generalisation of Lemma 18 to the case of $\ell = 2$, we have that there are at most three suitable subgroups describing the next isogeny. Since one of them is the backtracking isogeny (*i.e.* the dual of the last 2-isogeny in the 2^{k-1}-isogeny), this will give us two options for each of the 2^{k-1}-isogeny. The claim about at most four 2^k-isogenies follows. □

C Proof of Proposition 24

In this section, we give a proof of Proposition 24, which we recall below.

Proposition 35. *Let $t \in \mathcal{S}$. Then, we can compute a basis (P, Q) of $E[t]$ and $\pm(\varphi(P), \varphi(Q))$ if $t \neq 0 \pmod 8$, or $\pm([1 + \epsilon 2^{k-1}]\varphi(P), [1 + \epsilon 2^{k-1}]\varphi(Q))$ where $t = 2^k$ and $\epsilon \in \{0, 1\}$. This computation can be carried in polynomial time.*

Proof. Using Lemma 23, we compute (P_i, Q_i), a basis of eigenvectors for $\theta_{i^t|E[t]}$, *i.e.*, $\theta_{i^t}(P_i) = [s_P^{(i)}]P_i$ and $\theta_{i^t}(Q_i) = [s_Q^{(i)}]Q_i$, for some $s_{P_i}^{(i)}, s_{Q_i}^{(i)} \in \mathbb{Z}$. Also, let $(P_i^{E'}, Q_i^{E'})$ be a basis of eigenvectors for $\theta_{i^t}^{E'}{}_{|E'[t]}$.

Compute $a_{P_2}, b_{P_2}, a_{Q_2}, b_{Q_2} \in \mathbb{Z}$ such that $P_2 = [a_{P_2}]P_1 + [b_{P_2}]Q_1$ and $Q_2 = [a_{Q_2}]P_1 + [b_{Q_2}]Q_1$; this can be computed in polynomial time in $\log(p)$ by the assumption on t. Since $\theta_{1^t}\theta_{2^t} - \theta_{2^t}\theta_{1^t|E[\ell]} \neq [0]_{|E[\ell]}$, either $a_{P_2}b_{P_2} \neq 0 \pmod \ell$ or $a_{Q_2}b_{Q_2} \neq 0 \pmod \ell$. Without loss of generality, assume that $a_{P_2}b_{P_2} \neq 0 \pmod \ell$.

We may also assume that $\gcd(t, d) = 1$. If $\gcd(t, d) > 1$, we can apply Lemma 21 to recover the isogeny $\rho : E' \to E''$ with kernel $\ker(\widehat{\varphi})[\gcd(t, d)]$: at least one of the endomorphisms θ_1 and θ_2 has to satisfy the conditions of Lemma 21 otherwise $\gcd(\deg(\theta_1\theta_2 - \theta_2\theta_1), t) > 1$, a contradiction.[16] Let $\varphi' : E \to E''$ be the isogeny such that $\varphi = \widehat{\rho} \circ \varphi'$; in particular $\gcd(t, \deg(\varphi')) = 1$. This means that we could study the isogeny φ' and then transfer the information to φ via the isogeny $\widehat{\rho}$.

Proceeding in a similar way as in Section 3.1, we retrieve information about $(\varphi(P_1), \varphi(Q_1))$. First, observe that there exists some invertible scalars $c_{P_1}, c_{Q_1}, c_{P_2}$

[16] We remark that it is not necessary to check the mod-4 condition (Definition 9) in this case, since there is no need to apply Lemma 20.

modulo t such that $\varphi(P_1) = [c_{P_1}]P_1^E$, $\varphi(Q_1) = [c_{Q_1}]Q_1^E$ and $\varphi(P_2) = [c_{P_2}]P_2^E$. We now recover such scalars by looking at the following system of equations.

$$\begin{cases} e_t(P_1^E, Q_1^E) = e_t(P_1, Q_1)^{dc_{P_1}c_{Q_1}} \\ e_t(P_2^E, Q_1^E) = e_t(P_2, Q_1)^{dc_{P_2}c_{Q_1}} = e_t(P_1, Q_1)^{da_{P_2}c_{P_2}c_{Q_1}} \\ e_t(P_2^E, P_1^E) = e_t(P_2, P_1)^{dc_{P_2}c_{P_1}} = e_t(P_1, Q_1)^{-db_{P_2}c_{P_2}c_{P_1}} \end{cases} \tag{11}$$

From Equation (11), we get $c_{P_1}^2$ and $c_{Q_1}^2$. If t is odd or $t = 2, 4$, we can compute c_{P_1} and c_{Q_1} up to the same sign by extracting two square roots and comparing such choices against the first equation of Equation (11). If $t = 2^k$, for $k > 2$, there are four square roots. As a consequence, we recover $\pm c_{P_1} + \epsilon_1 \cdot 2^{k-1}$ and $\pm c_{Q_1} + \epsilon_2 \cdot 2^{k-1}$, for $\epsilon_1, \epsilon_2 \in \{0, 1\}$. Again, comparing such values against the first equation of Equation (11), we obtain $\pm(c_{P_1} + \epsilon \cdot 2^{k-1}, c_{Q_1} + \epsilon \cdot 2^{k-1})$, for the same $\epsilon \in \{0, 1\}$. □

D Proof of Lemma 25

In this section, we give a proof of Lemma 25, which we now recall.

Lemma 36. *Two integral elements $\theta_1, \theta_2 \in B_{p,\infty}$ generate an order if and only if $t := \mathrm{tr}(\theta_1\theta_2) \in \mathbb{Z}$. When the two elements have the same trace in absolute value and the same norm, the order is full rank unless $\theta_2 \in \{\pm\theta_1, \pm\bar{\theta}_1\}$.*

Proof. As $\theta_1\theta_2 \in \mathbb{Z}[\theta_1, \theta_2]$, we clearly need $t \in \mathbb{Z}$ for $\mathbb{Z}[\theta_1, \theta_2]$ to be an order.

Now assume that $t \in \mathbb{Z}$. We show that $\mathbb{Z}[\theta_1, \theta_2] = \Lambda$, where Λ is the lattice generated by $\{1, \theta_1, \theta_2, \theta_1\theta_2\}$. It suffices to show that any element of the form

$$\theta_1^{e_{1,1}}\theta_2^{e_{1,2}}\theta_1^{e_{2,1}}\theta_2^{e_{2,2}} \dots \theta_1^{e_{m,1}}\theta_2^{e_{m,2}}$$

with positive exponents $e_{i,j}$ is in Λ. First, notice that for any integral element θ and any positive integer e, we have $\theta^e \in \mathbb{Z}[1, \theta]$. Indeed the property is clear for $e = 0, 1$, and we prove it by recursion for larger e values. Assume that for any $i < e$, there exist integers a_i and b_i such that $\theta^i = a_i\theta + b_i$. Then $\theta^{i+1} = \theta(a_i\theta + b_i) = a_i\theta^2 + b_i\theta = (b_i + a_ia_2)\theta + a_ib_2$, as required. By substituting these expressions above, it now suffices to prove that any element of the form

$$\theta_1\theta_2\theta_1\theta_2 \dots \theta_i$$

is in Λ with $i \in \{1, 2\}$. The property is clear when the product has at most two factors. Suppose it is true up to i factors, i.e. the ith product can be written as $a_i\theta_1\theta_2 + b_i\theta_2 + c_i\theta_1 + d_i$. Then the $(i+1)$th product is either $(a_i\theta_1\theta_2 + b_i\theta_2 + c_i\theta_1 + d_i)\theta_1$ or $(a_i\theta_1\theta_2 + b_i\theta_2 + c_i\theta_1 + d_i)\theta_2$. The two elements are in Λ since

$$\begin{aligned} \theta_1^2 &= -\mathrm{tr}(\theta_1)\theta_1 + n(\theta_1) \\ \theta_2^2 &= -\mathrm{tr}(\theta_2)\theta_2 + n(\theta_2) \\ \theta_2\theta_1 &= \mathrm{tr}(\theta_2\theta_1) - \bar{\theta}_1\bar{\theta}_2 = -\theta_1\theta_2 + \mathrm{tr}(\theta_1)\theta_2 + \mathrm{tr}(\theta_2)\theta_1 + \mathrm{tr}(\theta_1\theta_2) - \mathrm{tr}(\theta_1)\,\mathrm{tr}(\theta_2) \\ \theta_1\theta_2\theta_1 &= \theta_1(\mathrm{tr}(\theta_2\theta_1) - \bar{\theta}_1\bar{\theta}_2) = n(\theta_1)\theta_2 + \mathrm{tr}(\theta_1\theta_2)\theta_1 - n(\theta_1)\,\mathrm{tr}(\theta_2) \\ \theta_1\theta_2^2 &= \theta_1(-\mathrm{tr}(\theta_2)\theta_2 - n(\theta_2)) = -\mathrm{tr}(\theta_2)\theta_1\theta_2 - n(\theta_2)\theta_1 \end{aligned}$$

are all in Λ.

Suppose that θ_1, θ_2 generate a quadratic order, and let ω with trace zero such that $\mathbb{Z}[\theta_i] \subset \mathbb{Z}[\omega]$ for $i = 1, 2$. Write $\theta_i = w_i + d_i\omega$ with $w_i, d_i \in \mathbb{Z}$. If θ_1 and θ_2 have the same trace, then $w_1 = \pm w_2$, and if they also have the same norm then $d_1 = \pm d_2$. This shows that $\theta_2 \in \{\pm\theta_1, \pm\bar{\theta}_1\}$. □

E Proof of Lemma 27

In this section, we give a proof of Lemma 27, which we recall in what follows.

Lemma 37. *Equation (8) is equivalent to*

$$(n^2 - M^2)(x_1^2 + py_1^2)p\alpha'^2 + M\beta'^2 + \gamma'^2 = 0. \tag{12}$$

through an efficiently computable invertible linear change of variables.

Proof. We define the inner product $\langle (a, b, c), (a', b', c') \rangle = aa' + pbb' + pcc'$, as well as the vector $v_1 := (x_1, y_1, z_1)$ and the (unknown) vector $v := (\alpha, \beta, \gamma)$. Equation (8) can be rewritten as

$$(M - n)\langle v, v \rangle = 2(\langle v, v_1 \rangle)^2.$$

Let v_2, v_3 be such that $\{v_1, v_2, v_3\}$ are an orthogonal basis. We can choose $v_2 := (-py_1, x_1, 0)$, then let $v_3' := (-pz_1, 0, x_1)$ and

$$v_3 := v_3' - \frac{\langle w_3', w_2 \rangle}{\langle w_2, w_2 \rangle} w_2.$$

Next, we define $s_i := \langle v, v_i \rangle$ for $i = 1, 2, 3$. This also defines an invertible linear change of variables between $\{\alpha, \beta, \gamma\}$ and $\{s_1, s_2, s_3\}$. Because $\{v_1, v_2, v_3\}$ is an orthogonal basis, we can write

$$v = \frac{s_1}{\langle v_1, v_1 \rangle} v_1 + \frac{s_2}{\langle v_2, v_2 \rangle} v_2 + \frac{s_3}{\langle v_3, v_3 \rangle} v_3$$

and express its norm as

$$\langle v, v \rangle = \frac{s_1^2}{\langle v_1, v_1 \rangle} + \frac{s_2^2}{\langle v_2, v_2 \rangle} + \frac{s_3^2}{\langle v_3, v_3 \rangle}.$$

Equation (8) can then be put in diagonal form

$$A_1 s_1^2 + A_2 s_2^2 + A_3 s_3^2 = 0$$

where

$$A_1 := \frac{(M - n)}{\langle v_1, v_1 \rangle} - 2, \qquad A_2 := \frac{(M - n)}{\langle v_2, v_2 \rangle}, \qquad A_3 := \frac{(M - n)}{\langle v_3, v_3 \rangle}.$$

More precisely, we compute

$$\begin{aligned}\langle v_1, v_1\rangle &= x_1^2 + py_1^2 + pz_1^2 = M\\ \langle v_2, v_2\rangle &= p^2y_1^2 + px_1^2 = p(x_1^2 + py_1^2)\\ \langle v_3', v_2\rangle &= p^2y_1z_1\end{aligned}$$

then

$$\begin{aligned}v_3 &= (-pz_1, 0, x_1) - \frac{py_1z_1}{x_1^2 + py_1^2}(-py_1, x_1, 0)\\ &= \frac{1}{x_1^2 + py_1^2}(-px_1^2z_1, -px_1y_1z_1, x_1(x_1^2 + py_1^2))\end{aligned}$$

and

$$\begin{aligned}\langle v_3, v_3\rangle &= \frac{1}{(x_1^2 + py_1^2)^2}(p^2x_1^4z_1^2 + p^3x_1^2y_1^2z_1^2 + px_1^2(x_1^2 + py_1^2)^2)\\ &= \frac{px_1^2}{(x_1^2 + py_1^2)^2}(px_1^2z_1^2 + p^2y_1^2z_1^2 + (x_1^2 + py_1^2)^2)\\ &= \frac{px_1^2M}{x_1^2 + py_1^2}.\end{aligned}$$

Finally, we have

$$A_1 = -\frac{(M+n)}{M}, \qquad A_2 = \frac{(M-n)}{p(x_1^2 + py_1^2)}, \qquad A_3 = \frac{(M-n)(x_1^2 + py_1^2)}{px_1^2M}.$$

Applying a second change of variables

$$\begin{aligned}s_1 &= (M-n)M(x_1^2 + py_1^2)pt_1,\\ s_2 &= p(x_1^2 + py_1^2)Mt_2,\\ s_3 &= px_1Mt_3\end{aligned}$$

and simplifying common factors, we get

$$(n^2 - M^2)(x_1^2 + py_1^2)pt_1^2 + Mt_2^2 + t_3^2 = 0.$$

□