SQIsignHD: New Dimensions in Cryptography

Pierrick Dartois, Antonin Leroux, Damien Robert, Benjamin Wesolowski

SQIsignHD: New Dimensions in Cryptography

Pierrick Dartois[1,2][0009−0008−2808−9867], Antonin Leroux[3,4][0009−0002−3737−0075], Damien Robert[1,2][0000−0003−4378−4274] and Benjamin Wesolowski[5][0000−0003−1249−6077]

[1] Univ. Bordeaux, CNRS, INRIA, IMB, UMR 5251, F-33400 Talence, France
[2] INRIA, IMB, UMR 5251, F-33400, Talence, France
{pierrick.dartois,damien.robert}@inria.fr
[3] DGA-MI, Bruz, France,
[4] IRMAR - UMR 6625, Université de Rennes, France
antonin.leroux@polytechnique.org
[5] ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France
benjamin.wesolowski@ens-lyon.fr

Abstract. We introduce SQIsignHD, a new post-quantum digital signature scheme inspired by SQIsign. SQIsignHD exploits the recent algorithmic breakthrough underlying the attack on SIDH, which allows to efficiently represent isogenies of arbitrary degrees as components of a higher dimensional isogeny. SQIsignHD overcomes the main drawbacks of SQIsign. First, it scales well to high security levels, since the public parameters for SQIsignHD are easy to generate: the characteristic of the underlying field needs only be of the form $2^f 3^{f'} - 1$. Second, the signing procedure is simpler and more efficient. Our signing procedure implemented in C runs in 28 ms, which is a significant improvement compared to SQISign. Third, the scheme is easier to analyse, allowing for a much more compelling security reduction. Finally, the signature sizes are even more compact than (the already record-breaking) SQIsign, with compressed signatures as small as 109 bytes for the post-quantum NIST-1 level of security. These advantages may come at the expense of the verification, which now requires the computation of an isogeny in dimension 4, a task whose optimised cost is still uncertain, as it has been the focus of very little attention. Our experimental `sagemath` implementation of the verification runs in around 600 ms, indicating the potential cryptographic interest of dimension 4 isogenies after optimisations and low level implementation.

Acknowledgements. We thank Luca De Feo for his advice all along this project and for suggesting the title of this paper. This project was supported by ANR grant CIAO (ANR-19-CE48-0008), PEPR PQ-TLS (the France 2030 program under grant agreement ANR-22-PETQ-0008 PQ-TLS) and the European Research Council under grant No. 101116169 (AGATHA CRYPTY).

1 Introduction

Isogeny-based cryptography has been a promising area of research in post-quantum cryptography since Couveignes, Rostovtsev and Stolbunov introduced

the first key exchange using ordinary isogenies [8, 34]. Schemes from this family often distinguish themselves by their compactness, in particular with respect to key sizes. It is notably the case of the digital signature scheme SQIsign [10, 13], the most compact post-quantum signature scheme by a decent margin. However, efficiency has been a recurring challenge for isogeny-based schemes, and indeed, SQIsign is much slower than other post-quantum signatures.

In this paper, we introduce SQIsignHD, a new digital signature scheme derived from SQIsign. As in [15], SQIsign uses the Deuring correspondence between supersingular elliptic curves and quaternion orders. This Deuring correspondence is a powerful tool to construct cryptosystems because it is one way: it is easy to turn an order into the corresponding elliptic curve, but the converse direction is the presumably hard *supersingular endomorphism ring problem* [12, 41]. In SQIsign, the signer's public key is a supersingular elliptic curve, and a signature effectively proves that the signer knows the associated quaternion order. This requires algorithms to translate between orders (and ideals in these orders) and elliptic curves (and isogenies from these curves). This translation is costly, and crucially requires the ideals (or isogenies) to have smooth norms (or degrees). The original methods have been improved upon [13], but that remains the bottleneck of SQIsign. Another issue with SQIsign is its scalability to higher security levels. Indeed, to set public parameters, one needs to find a prime p such that $p^2 - 1$ has a very large smooth factor. Searching for such primes p becomes harder as the security level grows, and is still an active area of research [7, 4, 1]. Besides, the security of SQIsign relies on the fact that signatures are computationally indistinguishable from random isogenies of fixed powersmooth degrees. There is no known formal proof of this *ad hoc* heuristic assumption.

The new scheme SQIsignHD follows a similar outline as SQIsign, but resolves its main drawbacks by fundamentally reforging the computational approach. The main ingredient is the ground-breaking technique that has recently led to the downfall of SIDH [5, 27, 33]. Namely, these attacks use a lemma due to Kani [19] combined with Zahrin's trick, which allows one to "embed" any isogeny into an isogeny of higher dimension. As remarked in [32], this technique allows one to describe an isogeny by listing only the image of a few well-chosen points; from this description, one can efficiently evaluate the isogeny on any other point, regardless of the factorisation pattern of the underlying isogeny. This newly gained freedom on usable isogenies unlocks challenges in efficiency, security, and scalability.

Our Contribution. We introduce the digital signature scheme SQIsignHD. It leverages recent algorithmic breakthroughs [5, 27, 33] to overcome the main drawbacks of SQIsign. It has the following advantages:

– SQIsignHD scales well to high security levels. Indeed, while SQIsign requires a search for primes p with strong constraints, the primes used in SQIsignHD may be of the form $c2^f3^{f'} - 1$, where c is some (preferably small) cofactor. Such primes, already used in SIDH [18], are easy to find, and allow for efficient field arithmetic.

- The signing procedure of SQIsignHD is simpler and more efficient than SQIsign. Let us stress that no high dimensional isogeny needs to be computed when signing. Our proof-of-concept implementation, which still lacks many standard optimisations, is already about ten times faster than the fastest SQIsign implementation. This is discussed in further detail in Section 6.2.
- SQIsignHD is easier to analyse, allowing for a much more compelling security reduction to the supersingular endomorphism problem. Unlike in SQIsign, our proof of the zero-knowledge property in SQIsignHD relies on simple and plausible heuristic assumptions. In fact, we propose two variants of SQIsign, one of which is less efficient but benefits from a heuristic-free analysis. In both cases, the zero-knowledge property is based on a simulator which is given access to a non-standard oracle. We carefully discuss the impact of this oracle on the supersingular endomorphism problem.
- SQIsignHD signatures are even more compact than SQIsign, as they are only 6.5λ bits long, for λ bits of security. In particular, they are as small as 109 bytes for the NIST-1 security level. SQIsign already had the most compact signature and public keys combined of all post-quantum signature schemes, and SQIsignHD breaks this record.

These advantages may come at the expense of the verification, which now requires the computation of a chain of 2-isogenies in dimension 4 (or 8 in the less efficient variant). We provide an algorithm for the verification, and an experimental implementation in `sagemath` [36, 28]. An optimised low-level implementation is left for future work, hence the true cost of verification is still uncertain. The verification in SQIsign also requires the computation of a (longer!) chain of 2-isogenies, but only in dimension 1.

1.1 A Modular Overview of SQIsignHD

We introduce two distinct versions of SQIsignHD, optimised in different directions. FastSQIsignHD is optimised for speed, while RigorousSQIsignHD is optimised for the security proof. Note that the security proof applies to both: the difference lies in the proof being unconditional for RigorousSQIsignHD when given access to an oracle, but requiring additional heuristics for FastSQIsignHD (see [9, § D.2] and Section 5.2). Under the hood, FastSQIsignHD relies on isogenies of dimension 4, while RigorousSQIsignHD relies on isogenies of dimension 8. The reader may sense the parallel with the heuristic (dimension 4) and rigorous (dimension 8) variants of the algorithms of [33].

We present here the main algorithmic building blocks of the identification scheme undelying SQIsignHD to give a modular overview of the protocol. Those algorithms are presented in detail in this paper for FastSQIsignHD and in [9, § B] for RigorousSQIsignHD. Unsurprisingly, the protocol shares a lot of similarities with SQIsign. The full signature scheme can be derived from there with the Fiat-Shamir transform [14] as in [10, § 3.4] (see [9, § A.1] for details).

Public set-up. We choose a prime p and a supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ of known endomorphism ring $\mathcal{O}_0 \cong \mathrm{End}(E_0)$ such that E_0 has smooth torsion

defined over a small extension of $\mathbb{F}_{p^2}$ (of degree 1 or 2). In practice, one may use the curve $E_0 : y^2 = x^3 + x$ (and $p \equiv 3 \mod 4$).

Key generation. The prover generates a random secret isogeny $\tau : E_0 \longrightarrow E_A$ of fixed smooth degree D_τ. Then, the prover publishes E_A. Knowing τ, only the prover can compute the endomorphism ring $\mathrm{End}(E_A)$. In the fast method $\mathsf{FastKeyGen}$, the isogeny τ has degree $D_\tau = \Theta(p)$, which is heuristically sufficient to ensure that the distribution of E_A is computationally indistinguishable from uniform. In the alternate method $\mathsf{RigorousKeyGen}$, the degree is chosen a bit larger to make the distribution of E_A statistically close to uniform.

Commitment. The prover generates a random isogeny $\psi : E_0 \longrightarrow E_1$ of smooth degree D_ψ and returns E_1 to the verifier (ψ being secret). The resulting distribution for E_1 is as close as possible to the uniform distribution in the supersingular isogeny graph. As in the key generation, we propose a fast procedure $\mathsf{FastCommit}(E_0)$ in Section 3.3 resulting in a distribution heuristically indistinguishable from uniform, and a slower variant $\mathsf{RigorousCommit}(E_0)$ in [9, § B.2] which guarantees statistical closeness to uniform.

Challenge. The verifier generates a random isogeny $\varphi : E_A \longrightarrow E_2$ of smooth degree D_φ sufficiently large for φ to have high entropy. Then, φ is sent to the prover. The $\mathsf{Challenge}$ procedure is described in Section 3.2. Unlike SQIsign, we chose to start the challenge from E_A instead of E_1 in order to optimize the response process.

Response. The prover generates an *efficient representation* of an isogeny $\sigma : E_1 \longrightarrow E_2$ of small degree $q \simeq \sqrt{p}$ in the sense of the following definition and returns it to the verifier.

Definition 1. Let $\mathscr{A}$ be an algorithm and $\varphi : E \longrightarrow E'$ be an isogeny defined over a finite field $\mathbb{F}_q$. An *efficient representation* of φ (with respect to $\mathscr{A}$) is some data $D \in \{0,1\}^*$ of polynomial size in $\log(\deg(\varphi))$ and $\log(q)$ such that, on input D and $P \in E(\mathbb{F}_{q^k})$, $\mathscr{A}$ returns $\varphi(P)$ in polynomial time in $k \log(q)$ and $\log(\deg(\varphi))$.

There always exists an efficient representation of a smooth degree isogeny. For instance, it can be written as a chain of small degree isogenies. Until the recent attacks on SIDH [5, 27, 33], we did not know how to efficiently represent isogenies with non-smooth degrees without revealing the endomorphism ring of the domain. For that reason, the original version of SQIsign uses smooth degree isogenies for the signature. These smooth degree isogenies are found with a variant of the KLPT algorithm [20] and have very big degree $\simeq p^{15/4}$. This not only hurts efficiency, but also security: the isogeny σ is so carefully crafted that it is hard to simulate, and as a result, the zero-knowledge property of SQIsign is very *ad hoc*.

Now, the methods from [5, 27, 33] give much more freedom on the isogenies that can be efficiently represented. This allows SQIsignHD to improve both efficiency (using isogenies σ of degree as low as $\simeq \sqrt{p}$), and security (the isogenies σ are now nicely distributed, hence simulatable).

The idea is to "embed" σ into an isogeny of higher dimension — and that only requires knowing the image of a few points through σ. As in the attacks against SIDH, such an isogeny can have dimension 2, 4 or 8. We shall see that dimension 2 has little interest compared to the original SQIsign protocol from an efficiency and security point of view. In SQIsignHD, we propose a response procedure FastRespond to represent σ in dimension 4, and an alternative procedure RigorousRespond based on an isogeny computation in dimension 8. The procedure FastRespond is fast, and its security analysis relies on reasonable heuristics. On the other hand, RigorousRespond is much slower (though still polynomial time), but allows for a rigorous analysis.

In either case, for efficiency reasons, the prover does not actually compute higher dimensional isogenies but only images of some points through σ (we explain how these points are evaluated in the course of the paper). Those points provide an efficient representation of σ (along with $\deg(\sigma)$) and this data is sent to the verifier who can then compute higher dimensional isogenies representing σ.

Verification. The verifier checks that the response returned by the prover (points of E_2) correctly represents an isogeny $\sigma : E_1 \longrightarrow E_2$. We propose two procedures FastVerify and RigorousVerify computing isogenies embedding σ in dimension 4 or 8. So far, isogeny computations in dimension 4 has been the subject of very little literature.

Nonetheless, our proof of concept implementation of dimension 4 isogenies in `sagemath` [36, 28] demonstrates the cryptographic feasibility of this phase. We expect an optimized implementation to be at worse twice as slow as the original SQIsign verification, and hopefully even closer than that. We refer to [9, § F] for an estimate of the number of operations required for the verification.

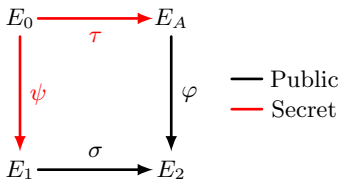


Fig. 1. The SQIsign/SQIsignHD identification protocol.

Content. The rest of this paper is organized as follows. In Section 2, we present the core idea of our paper: how to embed signature/response isogenies in higher dimension with Kani's lemma. Section 3 introduces algorithms for key generation, commitment and challenge whereas Section 4 presents the response and verification phase for FastSQIsignHD. A security analysis of FastSQIsignHD identification protocol is conducted in Section 5. Finally, we discuss the expected performance of the digital signature scheme derived from FastSQIsignHD in Section 6. To save space, some preliminaries, proofs and algorithmic details on RigorousSQIsignHD and higher dimensional isogenies are given in [9].

2 Representing the Response Isogeny Efficiently in Higher Dimension

In this section, we explore our main idea to improve SQIsign by embedding the signature isogeny inside an isogeny in higher dimension. We start by recalling how the signature is represented in the original SQIsign protocol in Section 2.1 and why this representation is slow to compute. Then, we introduce Kani's lemma and explain how to embed isogenies in higher dimension in Section 2.2. Finally, we apply this idea to provide another representation of the signature isogeny in SQIsign in Section 2.3.

2.1 State of the Art Isogeny Representation: a Slow Signature Process

With state of the art techniques prior to the attacks against SIDH, we could only efficiently represent isogenies of smooth degrees. That is why in the original versions of SQIsign [10, 13], the signature isogeny σ has degree a prime power ℓ^e and is represented as a chain of ℓ-isogenies.

To compute such a signature σ, the prover computes the ideal J associated to the isogeny path given by the secret key, commitment and challenge. They then apply a SigningKLPT algorithm to J, to return a random equivalent ideal $I \sim J$ of norm ℓ^e. Then, the prover converts I into an isogeny. This last computation is very costly because $\mathrm{nrd}(I) = \ell^e$ is close to $p^{15/4}$, while the accessible torsion points have much smaller order. The method introduced in [10] (and later improved in [13]) requires to cut J into several pieces in order to compute σ as a chain of isogenies. This complicated mechanism is by far the bottleneck in the signing algorithm.

In order to avoid this costly ideal to isogeny translation in SQIsignHD, we shall no longer require σ to have smooth degree and embed it in an isogeny of dimension 4 or 8 having smooth degree. This embedding will provide an efficient representation, and is faster to compute than the one in the original SQIsign. We shall also explain why this improves security in Section 5.

2.2 Embedding Isogenies in Higher Dimension with Kani's Lemma

In this section, we explain in more detail this idea of embedding isogenies in higher dimension. For that, we need a few definitions first.

Definition 2 (d-isogeny)**.** Let $\alpha : (A, \lambda_A) \longrightarrow (B, \lambda_B)$ be an isogeny between principally polarized abelian varieties. We say that α is a *d-isogeny* if $\widehat{\alpha} \circ \lambda_B \circ \alpha = [d]\lambda_A$, where $\widehat{\alpha} : \widehat{B} \longrightarrow \widehat{A}$ is the dual isogeny of α.

Equivalently, α is a d-isogeny if $\widetilde{\alpha} \circ \alpha = [d]_A$, where $\widetilde{\alpha} := \lambda_A^{-1} \circ \widehat{\alpha} \circ \lambda_B$ is the dual isogeny of α with respect to the principal polarisations λ_A and λ_B.

Definition 3 (Isogeny diamond)**.** Let $a, b \in \mathbb{N}^*$. An (a, b)-*isogeny diamond* is a commutative diagram of isogenies between principally polarized abelian varieties

$$\begin{array}{ccc} A' & \xrightarrow{\varphi'} & B' \\ {\scriptstyle\psi}\uparrow & & \uparrow{\scriptstyle\psi'} \\ A & \xrightarrow{\varphi} & B \end{array}$$

where φ and φ' are a-isogenies and ψ and ψ' are b-isogenies.

Lemma 4 (Kani)**.** *We consider an (a, b)-isogeny diamond as above, with $d := a + b$ prime to the characteristic of the base field of abelian varieties. Then, the isogeny $F : A \times B' \longrightarrow B \times A'$ given in matrix notation by*

$$F := \begin{pmatrix} \varphi & \widetilde{\psi'} \\ -\psi & \widetilde{\varphi'} \end{pmatrix}$$

is a d-isogeny with $d = a + b$, for the product polarisations.

If a and b are coprime, the kernel of F is

$$\ker(F) = \{(\widetilde{\varphi}(x), \psi'(x)) \mid x \in B[d]\}.$$

This lemma has first been proved in [19, Theorem 2.3]. We also give a proof in [9, § E.1].

Remark 2.1. The existence of $F : A \times B' \to B \times A'$, implies the existence of $\varphi : A \to B$. We can recover φ as $\pi \circ F \circ \iota$ where ι is the embedding morphism $x \in A \longmapsto (x, 0) \in A \times B'$ and π is the projection from $B \times A'$ to B. Hence, F is an efficient representation of φ.

2.3 Application of Kani's Lemma to SQIsign

Let us now see how we propose to use Kani's Lemma (Lemma 4) in SQIsignHD.

Signing in Dimension 4. The idea is to embed the signature $\sigma : E_1 \longrightarrow E_2$ in an isogeny of dimension 4. We consider the 2-dimensional q-isogeny $\Sigma := \mathrm{Diag}(\sigma, \sigma) : E_1^2 \longrightarrow E_2^2$, and for $a_1, a_2 \in \mathbb{Z}$ and $i \in \{1, 2\}$ the $(a_1^2 + a_2^2)$-isogeny

$$\alpha_i := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} \in \mathrm{End}(E_i^2).$$

Then, we have an isogeny diamond

$$\begin{array}{ccc} E_2^2 & \xrightarrow{\alpha_2} & E_2^2 \\ {\scriptstyle\Sigma}\uparrow & & \uparrow{\scriptstyle\Sigma} \\ E_1^2 & \xrightarrow{\alpha_1} & E_1^2 \end{array}$$

yielding an N-isogeny (with $N := q + a_1^2 + a_2^2$):

$$F := \begin{pmatrix} \alpha_1 & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}_2 \end{pmatrix} \in \mathrm{End}(E_1^2 \times E_2^2).$$

Notation 5. We shall denote $F(\sigma, a_1, a_2)$ when we want to specify the dependence of F on σ, a_1, a_2.

We choose the parameters q, a_1, a_2, so that $N = \ell^e$, with ℓ a small prime and $e \in \mathbb{N}^*$ big enough. Provided that q and ℓ are coprime, we know that

$$\ker(F) = \{(\widetilde{\alpha}_1(P), \Sigma(P)) \mid P \in E_1^2[\ell^e]\}, \tag{1}$$

by Lemma 4. Then, knowing $\ker(F)$ we can compute F as an ℓ-isogeny chain and obtain an efficient representation of σ, as explained in Remark 2.1.

It follows that our idea requires to compute $\ker(F)$, which becomes easy once we know how to evaluate σ on $E_1[\ell^e]$, by formula 1. The idea is to use the alternate isogeny path $\varphi \circ \tau \circ \widehat{\psi} : E_1 \to E_2$. Since the signature requires to compute the three isogenies φ, ψ, τ, it will not cost too much to use them in order to evaluate σ. There are several technicalities to make it work in practice (such as to making sure that this alternate path has degree prime to ℓ) but it is manageable (see [9, § A.5]).

Computing such a representation for the signature is simpler than in the original SQIsign protocol. This shifts the main computation effort to the verification, where the actual isogeny in dimension 4 must be computed.

Parameters. Even though we no longer impose $q = \deg(\sigma)$ to be smooth, we still impose conditions on q to make it work. We shall need $\ell^e - q$ to be a prime congruent to 1 modulo 4 in order to decompose it easily as a sum of two squares $\ell^e - q = a_1^2 + a_2^2$ by Cornacchia's algorithm [6]. This choice of q ensures its coprimality with ℓ, as required to compute $\ker(F)$. The exponent e is fixed to be as small as possible so that there always exists an isogeny $\sigma : E_1 \longrightarrow E_2$ of ℓ^e-*good* degree in the sense of the following definition. In practice, the smallest values for q are close to $\sqrt{p}$ (Section 4.2) so ℓ^e will be slightly bigger than $\sqrt{p}$.

Definition 6. We say that an integer q is ℓ^e-*good* when $\ell^e - q$ is a prime number congruent to 1 modulo 4.[6]

Remark 7 (The issue of the signature distribution)**.** Those restrictions on the degree q impact the distribution of signatures. The bound $\ell^e \simeq \sqrt{p}$ is also restrictive (see [9, Theorem 42]). For that reason, we need some plausible heuristic assumptions to prove the zero-knowledge property of our scheme. This can be

[6] One could improve slightly the scheme by defining ℓ^e-good integers as integers q such that $\ell^e - q = sq'$, with s a smooth integer whose prime factors are all congruent to 1 modulo 4 and q' is a prime congruent to 1 modulo 4. Indeed, all we really need is that $\ell^e - q$ is easy to factor so Cornacchia's algorithm can be applied efficiently. This alternate definition would improve a bit the search for ℓ^e-good integer, but we went for the simplest definition.

fixed by going to dimension 8 as long as $q < \ell^e$ and $\ell^e = \Omega(p^2)$. This way, we shall obtain a uniform distribution of signatures and a provably zero-knowledge scheme which is the purpose of our scheme in dimension 8 that we present below.

Signing in Dimension 8. By Lagrange's four square theorem [21], if $q < \ell^e$, there always exists $a_1, \cdots, a_4 \in \mathbb{Z}$ such that $q + a_1^2 + \cdots + a_4^2 = \ell^e$. We can find such a decomposition in polynomial time in e with Rabin and Shallit's algorithm [31] improved by Pollack and Treviño [30]. We then consider the endomorphisms

$$\alpha_i := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix} \in \mathrm{End}(E_i^4),$$

for $i \in \{1, 2\}$, which are $(a_1^2 + \cdots + a_4^2)$-isogenies, and the q-isogeny $\Sigma := \mathrm{Diag}(\sigma, \cdots, \sigma) : E_1^4 \longrightarrow E_2^4$. As previously, by Kani's lemma, we have the ℓ^e-isogeny

$$F := \begin{pmatrix} \alpha_1 & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}_2 \end{pmatrix} \in \mathrm{End}(E_1^4 \times E_2^4).$$

Similarly to dimension 4, we write $F(\sigma, a_1, \cdots, a_4)$ to highlight the dependence of F on $\sigma, a_1, \cdots, a_4$. To ensure the uniformity of the response, in dimension 8 we no longer restrict to the case q prime to ℓ. This means we might have to embed in dimension 8 a factor of σ of degree prime to ℓ instead of σ (see [9, § C] for details). As in dimension 4, can compute $\ker(F)$ by evaluating σ on $E_1[\ell^e]$ and then compute F as an ℓ-isogeny chain. This way, we can represent any signature isogeny σ of degree $q < \ell^e$, with the implications on the security proof that we mentioned before. However, computing isogenies in dimension 8 is much more costly than in dimension 4 (though, still polynomial), so we do not recommend to use this representation and only propose it in the alternate version RigorousSQIsignHD.

More generally, the same techniques allow, given an ideal I representing an isogeny of degree q, to give an efficient representation of the isogeny σ associated to I by the Deuring correspondance, even when q is not smooth (see [9, § A.3-4]).

Why not Signing in Dimension 2? The cost of computing an isogeny grows exponentially with the dimension [24, 25, 26]. For that reason, finding an efficient representation in dimension 2 could be fruitful for SQIsignHD. On the other hand, the higher the dimension, the lesser the constraints on the isogeny σ. We have already seen that going from dimension 4 to 8 relaxes the constraints on $q = \deg(\sigma)$. Unsurprisingly, the constraints on σ are tighter in dimension 2. So far, under those constraints, we have failed to provide an efficient and secure version of SQIsignHD. We leave this question to future works.

3 Key Generation, Commitment and Challenge

To evaluate σ on the ℓ^e-torsion, as required for the response computation, we apply the $\mathsf{EvalTorsion}_{\ell^f}$ procedure ([9, § A.5]) which uses the alternate path

$\varphi \circ \tau \circ \widehat{\psi} : E_1 \longrightarrow E_2$ formed by the challenge φ, secret key τ and commitment isogeny ψ along with their ideals I_φ, I_τ and I_ψ. These ideals are also necessary to compute the ideal I_σ.

For the $\mathsf{EvalTorsion}_{\ell^f}$ procedure to work, the degrees of φ, τ and ψ must be prime to ℓ. The ideals I_ψ and I_τ can be generated directly along with ψ and τ. However, the computation of I_φ uses the procedure $\mathsf{IsogenyToIdeal}$ ([9, § A.4]) which requires a precomputation in the key generation phase. Namely, the prover will need to generate an alternate secret path $\tau' : E_0 \longrightarrow E_A$ of degree $D_{\tau'}$ prime to D_φ along with the secret key $\tau : E_0 \longrightarrow E_A$. This will be explained in section 3.3.

3.1 Accessible Torsion and Choice of the Prime Characteristic

The choice of p is usually made to provide enough accessible torsion for our isogeny computations. In FastSQIsignHD, we can choose $p = c\ell^f \ell'^{f'} - 1$ with $\ell \neq \ell'$ two primes, $c \in \mathbb{N}^*$ small and $\ell^f \simeq \ell'^{f'} \simeq \sqrt{p}$, as in SIDH [18]. In practice, $\ell = 2$ and $\ell' = 3$ are the best choice.

We then require $D_\tau = D_\psi = \ell'^{2f'}$, $D_\varphi = \ell'^{f'}$ and $D_{\tau'} = \ell^{2f}$[7]. This choice ensures that D_τ, D_ψ and D_φ are prime to ℓ and that $D_{\tau'}$ is prime to D_φ, as needed. We also have $D_\tau, D_\psi, D_{\tau'} = \Theta(p)$, which guarantees (at least heuristically) that the public key E_A and the commitment E_1 are computationally indistinguishable from a uniformly random supersingular elliptic curve – which is essential to the security of FastSQIsignHD.

This choice of prime also provides enough accessible torsion to compute the ℓ^e-isogeny F representing the response σ in dimension 4, where $\ell^e > q := D_\sigma$. In fact, we even have much more than the minimum requirement since it will be enough to have $2f \geq e + 4$ (so $\ell^f = \Omega(p^{1/4})$) as will be explained in Sections 4.3 and 4.4 and Remark 4.2. This freedom is welcome anyway because it allows us to take ℓ^e slightly bigger than $\sqrt{p}$ to make sure that we can always find an ideal I of ℓ^e-good norm $q < \ell^e$ (see Section 4.2).

We finally discuss the security requirements regarding the size of p. The best known classical key recovery attacks are the meet-in-the-middle algorithm or the general Delfs and Galbraith attack [11] in the supersingular isogeny graph which both have a complexity in $\tilde{O}(\sqrt{p})$. Using Grover's algorithm [16], we reach a quantum complexity of $\tilde{O}(p^{1/4})$. Hence, to ensure a classical security level of λ bits and a quantum security level of $\lambda/2$ bits, we need to take $p = \Theta(2^{2\lambda})$, as in the original version of SQIsign [10].

We give below some concrete values of primes for NIST levels 1, 3 and 5.

NIST security level	Security parameter λ (bits)	Prime p
NIST-I	128	$13 \cdot 2^{126} \cdot 3^{78} - 1$
NIST-III	192	$5 \cdot 2^{193} \cdot 3^{122} - 1$
NIST-V	256	$11 \cdot 2^{257} \cdot 3^{163} - 1$

[7] Actually, we will not have exactly $D_\tau = D_\psi = \ell'^{2f'}$ but D_τ and D_ψ will be divisors of $\ell'^{2f'}$ close to $\ell'^{2f'}$. It will be the same for $D_{\psi'}$ (see Algorithm 1). We assume equality to simplify the exposition.

3.2 Challenge Generation

To ensure a soundness security level of λ bits, the challenge space needs to have size at least $2^\lambda \simeq \sqrt{p}$. We also need the challenge degree D_φ to be prime to ℓ to be able to push the points of order ℓ^f through ψ during the signing procedure. The challenge generation procedure $\mathsf{Challenge}_{D_\varphi}$ is the same in the fast and provably secure challenge generation procedure. It simply generates a random element $P \in E_A$ of order D_φ and computes φ of kernel $\langle P \rangle$. Only the degree D_φ changes; in FastSQIsignHD, we take $D_\varphi = \ell'^{f'}$.

3.3 Fast Key Generation and Commitment

We now present FastDoublePath (Algorithm 1) the main algorithmic block for the key generation and commitment of FastSQIsignHD. The goal of this algorithm is to generate two isogeny paths $\phi, \phi' : E_0 \longrightarrow E$ of degree dividing $\ell^{2f} \simeq p$ and $\ell'^{2f'} \simeq p$ respectively, computing the kernel ideals I_ϕ and $I_{\phi'}$ along the way. This algorithm is directly applicable to the key generation procedure FastKeyGen where we need to generate a double path to be able to compute the challenge kernel ideal I_φ (using [9, § A.4] and an ℓ-isogeny path of degree prime to ℓ') in order to apply the $\mathsf{EvalTorsion}_{\ell^f}$ procedure (with the ℓ'-isogeny path of degree prime to ℓ).

For the commitment FastCommit, we only need the ℓ'-isogeny path $\psi = \phi'$ but the algorithm is essentially the same, except that we do not compute ϕ and I_ϕ completely. This is the reason why we changed the side of the challenge: to save time in the commitment phase. Had we started the challenge φ from E_1 as in SQISign, we would have needed to compute a double isogeny path in the commitment phase. Instead, we precompute this double path during the key generation.

Note that generating isogenies of degree $\simeq p$ is essential for security reasons, in order to ensure that the codomain E is heuristically close to a random elliptic curve in the supersingular isogeny graph. To compute such long isogeny paths, however, we are limited by the accessible torsion in E_0 (we have access to the $\ell^f \ell'^{f'}$-torsion only). To circumvent this difficulty, we use pushforward isogenies, as defined in [10, § 4.1].

Definition 8. Let $\rho : E \longrightarrow E_1$ and $\theta : E \longrightarrow E_2$ be two isogenies with coprime degree. The *pushforward* of ρ via θ, denoted by $\rho' := [\theta]_*\rho$ is an isogeny $E_2 \longrightarrow E_3$ satisfying $\ker(\rho') = \theta(\ker(\rho))$.

Remark 9. θ and ρ satisfy $[\theta]_*\rho \circ \theta = [\rho]_*\theta \circ \rho$. In particular, $[\theta]_*\rho$ and $[\rho]_*\theta$ have the same codomain. If I and J are the ideals associated to ρ and θ respectively via the Deuring correspondence, we denote by $[J]_*I$ the *pushforward ideal* associated to $[\theta]_*\rho$. By [10, Lemma 3], the ideal $[J]_*I$ can be computed as follows: $[J]_*I = J^{-1} \cdot (I \cap J)$.

The Algorithm. The idea is to construct the isogenies ϕ and ϕ' (of degree dividing ℓ^{2f} and $\ell'^{2f'}$ respectively) by finding an endomorphism γ of degree dividing $\ell^{2f}\ell'^{2f'}$, and factoring it as $\gamma = \hat{\phi}' \circ \phi$. Since $\ell^{2f}\ell'^{2f'} = \Theta(p^2) = \omega(p)$, we can easily find $\gamma \in \mathcal{O}_0$ non divisible by ℓ or ℓ', of norm $\mathrm{nrd}(\gamma) = \ell^{2g}\ell'^{2g'}$ with $g \leq f$ close to f and $g' \leq f'$ close to f', using [22, Algorithm 4].

Since ℓ^{2f} (and $\ell'^{2f'}$) exceeds the available torsion, some "pushforward gymnastics" is required to compute the factorisation. We thus decompose $\varepsilon(\gamma) = \widehat{\rho_2} \circ \rho_1$ where ρ_1 and ρ_2 are isogenies $E_0 \longrightarrow E'$ of degree $\ell^g\ell'^{g'}$ and ε is an isomorphism $\mathcal{O}_0 \xrightarrow{\sim} \mathrm{End}(E_0)$. $\varepsilon(\gamma)$ being cyclic, according to the following lemma, ρ_1 and its associated kernel ideal K_1 are given by:

$$\ker(\rho_1) = \ker(\varepsilon(\gamma)) \cap E_0[\ell^g\ell'^{g'}] \quad \text{and} \quad K_1 = \mathcal{O}_0\gamma + \mathcal{O}_0\ell^g\ell'^{g'}.$$

Similarly, $\ker(\rho_2) = \ker(\widehat{\varepsilon(\gamma)}) \cap E_0[\ell^g\ell'^{g'}]$ and the associated kernel ideal is $K_2 = \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0\ell^g\ell'^{g'}$.

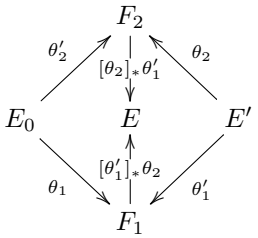
Lemma 10. *Let $\rho : E \longrightarrow E'$ be a cyclic isogeny decomposed into $\rho = \theta \circ \rho_1$. Then we have:*

(i) $\ker(\rho_1) = \ker(\rho) \cap E[d_1]$ *with* $d_1 := \deg(\rho_1)$.
(ii) *If ρ is a cyclic endomorphism ($E = E'$), then the kernel ideal of ρ_1 is $K_1 = \mathcal{O}\rho + \mathcal{O}d_1$, where $\mathcal{O} := \mathrm{End}(E)$.*

Proof. Since $\rho = \theta \circ \rho_1$ and $\deg(\rho_1) = d_1$, we clearly have $\ker(\rho_1) \subseteq \ker(\rho) \cap E[d_1]$. Since ρ is cyclic, there exists a generator $P \in E$ of $\ker(\rho)$ of order $d := \deg(\rho)$ and we have $\ker(\rho) \cap E[d_1] = \langle [d/d_1]P \rangle$, where $[d/d_1]P$ has order d_1, so we conclude that the inclusion is an equality by cardinality, since ρ_1 is separable. (i) follows.

To prove (ii), we remark that $E[\mathcal{O}\rho + \mathcal{O}d_1] = E[\rho] \cap E[d_1] = \ker(\rho_1)$, where the last equality was proved in (i). Then, we conclude that $K_1 = \mathcal{O}\rho + \mathcal{O}d_1$ by injectivity of the Deuring correspondence between left $\mathcal{O}$-ideals and isogenies of domain E [40, Proposition 42.2.16]. This completes the proof. □

Then, we can decompose ρ_1 and ρ_2 into $\rho_1 = \widehat{\theta'_1} \circ \theta_1$ and $\rho_2 = \widehat{\theta_2} \circ \theta'_2$ where the θ_i are isogenies of degree ℓ^g and the θ'_i are isogenies of degree $\ell'^{g'}$ for $i \in \{1, 2\}$, as in the following diagram:



The pushforward isogenies $[\theta'_1]_*\theta_2$ and $[\theta_2]_*\theta'_1$ have the same codomain E and degree ℓ^g and $\ell'^{g'}$ respectively. Hence, $\phi := [\theta'_1]_*\theta_2 \circ \theta_1$ and $\phi' := [\theta_2]_*\theta'_1 \circ \theta'_2$ are isogenies $E_0 \longrightarrow E$ of desired degrees ℓ^{2g} and $\ell^{2g'}$ respectively. By Lemma

Algorithm 1: $\mathsf{FastDoublePath}_{\ell^f,\ell'^{f'}}$

Data: A basis of $\mathcal{O}_0$ and an isomorphism $\varepsilon : \mathcal{O}_0 \xrightarrow{\sim} \mathrm{End}(E_0)$.
Result: Two cyclic isogenies $\phi : E_0 \longrightarrow E$ of degree dividing ℓ^{2f} and $\phi' : E_0 \longrightarrow E$ of degree dividing $\ell'^{2f'}$ and their respective kernel ideals J and J'.

1 Use [22, Algorithm 4] to find $\gamma \in \mathcal{O}_0$ non divisible by ℓ and ℓ' of norm $\mathrm{nrd}(\gamma) = \ell^{2g}\ell'^{2g'}$ with $g \leq f$ close to f and $g' \leq f'$ close to f';
2 Evaluate $\varepsilon(\gamma)$ and $\varepsilon(\overline{\gamma})$ on a basis of $E_0[\ell^g\ell'^{g'}]$ and solve discrete logarithm problems to compute $\mathcal{G}_1 := \ker(\varepsilon(\gamma)) \cap E_0[\ell^g\ell'^{g'}]$ and $\mathcal{G}_2 := \ker(\widehat{\varepsilon(\gamma)}) \cap E_0[\ell^g\ell'^{g'}]$;
3 Compute $\rho_i : E_0 \longrightarrow E'$ of kernel $\mathcal{G}_i$ for $i = 1, 2$;
4 Compute $\mathcal{H}_1 := \ker(\varepsilon(\gamma)) \cap E_0[\ell^g]$, $\mathcal{H}'_2 := \ker(\widehat{\varepsilon(\gamma)}) \cap E_0[\ell'^{g'}]$, $\mathcal{H}'_1 := \ker(\widehat{\rho_1}) \cap E'[\ell'^{g'}]$ and $\mathcal{H}_2 := \ker(\widehat{\rho_2}) \cap E'[\ell^g]$;
5 Compute θ_i of kernel $\mathcal{H}_i$ and θ'_i of kernel $\mathcal{H}'_i$ for $i = 1, 2$;
6 Compute $[\theta'_1]_*\theta_2$ and $[\theta_2]_*\theta'_1$ of kernels $\theta'_1(\ker(\theta_2))$ and $\theta_2(\ker(\theta'_1))$ respectively;
7 Let $\phi := [\theta'_1]_*\theta_2 \circ \theta_1$ and $\phi' := [\theta_2]_*\theta'_1 \circ \theta'_2$;
8 Let $J := \mathcal{O}_0\gamma + \mathcal{O}_0\ell^{2g}$ and $J' := \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0\ell'^{2g'}$;
9 Return ϕ, ϕ', J, J';

10, we can compute $\ker(\theta_1)$, $\ker(\theta'_2)$, $\ker(\theta'_1)$ and $\ker(\theta_2)$, and obtain the θ_i and θ'_i with Vélu's formulas [38]. We then compute $\ker([\theta'_1]_*\theta_2) = \theta'_1(\ker(\theta_2))$ and $\ker([\theta_2]_*\theta'_1) = \theta_2(\ker(\theta'_1))$ and use Vélu's formulas again. We then easily get ϕ and ϕ'.

Since $\varepsilon(\gamma) = \widehat{\rho_2} \circ \rho_1$ and $[\theta'_1]_*\theta_2 \circ \theta'_1 = [\theta_2]_*\theta'_1 \circ \theta_2$, we get that $\varepsilon(\gamma) = \hat{\phi}' \circ \phi$. Lemma 10 implies that the ideals $J := \mathcal{O}_0\gamma + \mathcal{O}_0\ell^{2g}$ and $J' := \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0\ell'^{2g'}$ are the respective kernel ideals of ϕ and ϕ'. Algorithm 1 follows.

Remark 3.1. The FastKeyGen procedure calls Algorithm 1 directly. For FastCommit, only ϕ' and J' are necessary, so we use a slightly modified version of Algorithm 1 where $\mathcal{H}_1$ (line 4), θ_1 (line 5), ϕ (line 7), and J (line 8) are not computed.

4 Response and Verification

The goal of this section is to present a precise description of the algorithmic building blocks required by our new signature scheme in dimension 4. We refer to [9, § C] for details on the dimension 8 version.

Throughout this section, we assume that the prover has generated two secret key paths $\tau, \tau' : E_0 \longrightarrow E_A$ of respective degrees $D_\tau = \ell'^{2f'}$ and $D_{\tau'} = \ell^{2f}$ and a secret commitment path $\psi : E_0 \longrightarrow E_1$ of degree $D_\psi = \ell'^{2f'}$. We also assume the prover has access to the challenge $\varphi : E_A \longrightarrow E_2$ of degree $D_\varphi = \ell'^{f'}$.

4.1 Overview of the Response Computation

In this section, we present the algorithm FastRespond used to compute the response in the FastSQIsignHD identification protocol (in dimension 4) and its verification counterpart FastVerify.

Those algorithms use the following sub-algorithms that will be introduced in this section (if not already):

- $\mathsf{IsogenyToIdeal}(\varphi, \tau', I_{\tau'})$ (presented in [9, § A.4]) takes as input a basis of $\mathrm{End}(E_0)$ that we can evaluate on points, an isogeny $\varphi : E_A \longrightarrow E_2$ of degree D_φ, an isogeny $\tau' : E_0 \longrightarrow E_A$ of degree prime to D_φ, its ideal $I_{\tau'} \subset \mathcal{O}_0$ and returns the kernel ideal I_φ of φ.
- $\mathsf{RandomEquivalentIdeal}_{\ell^e}$ takes as input an $\mathcal{O}_0$-left ideal J and returns an equivalent ideal I that is uniformly random among ideals of norm $\leq \ell^e$.
- $\mathsf{EvalTorsion}_{\ell^f}$ (presented in [9, § A.5]) evaluates a non-smooth degree isogeny on ℓ^f-torsion points knowing its kernel ideal and an alternate smooth degree path. Namely, it takes as input an ideal I connecting $\mathcal{O} \cong \mathrm{End}(E)$ and $\mathcal{O}' \cong \mathrm{End}(E')$, a basis (P_1, P_2) of $E[\ell^f]$, two isogenies $\rho_1 : E_0 \longrightarrow E$ and $\rho_2 : E_0 \longrightarrow E'$ of smooth degrees prime to ℓ, with their respective kernel ideals I_1 and I_2 and returns $(\phi_I(P_1), \phi_I(P_2))$, where $\phi_I : E \longrightarrow E'$ is the isogeny associated to I.
- $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ takes as input an ℓ^e-good integer q, integers a_1, a_2 such that $a_1^2 + a_2^2 + q = \ell^e$, a basis (P_1, P_2) of $E_1[\ell^f]$, $(\sigma(P_1), \sigma(P_2))$, where $\sigma : E_1 \longrightarrow E_2$ is a q-isogeny, and returns a chain of 4-dimensional ℓ-isogenies whose composition is $F(\sigma, a_1, a_2)$ as in Notation 5.
- $\mathsf{IsValid}_4$, with input $F, E_1, E_2, \ell^e, \ell^f$, checks if F is a valid output of $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ representing an isogeny $\sigma : E_1 \longrightarrow E_2$ in dimension 4.

The prover sends the image of two points P_1, P_2 forming a basis of $E_1[\ell^f]$ by σ and its degree q. The verifier can then use q to compute a_1, a_2 and compute $F(\sigma, a_1, a_2)$ with the $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ procedure. If the computation succeeds and is validated by the $\mathsf{IsValid}_4$ procedure, then the verification is complete. Algorithm 3 follows.

Remark 4.1 (On the ℓ^f-torsion basis). It is sufficient to send the data $(\sigma(P_1), \sigma(P_2), q)$ to the verifier as the basis (P_1, P_2) can be computed canonically knowing E_1 by classical compression techniques developed for SIDH [2, 42]. This decreases the communications size at a small computational cost. Later, with the compression/decompression algorithms (see Algorithms 6 and 7), we will see how to further compress this data.

Note that we use a basis of the ℓ^f-torsion with $2f \geq e + 4$ here because we might not have the ℓ^e-torsion accessible. We can still compute F with this partial information as explained in Section 4.3.

To respond, the prover starts by computing an ideal $I \sim \overline{I_\psi} \cdot I_\tau \cdot I_\varphi$ connecting $\mathcal{O}_1 \cong \mathrm{End}(E_1)$ to $\mathcal{O}_2 \cong \mathrm{End}(E_2)$ of ℓ^e-good norm q and prime to ℓ' with uniform distribution using $\mathsf{RandomEquivalentIdeal}_{\ell^e}$. The coprimality with

ℓ' is justified by security reasons (see Section 5.1). Then, the prover generates the basis (P_1, P_2) of $E_1[\ell^f]$ canonically and evaluates σ on it with $\mathsf{EvalTorsion}_{\ell^f}$ using I (kernel ideal of σ) and the paths $\psi : E_0 \longrightarrow E_1$ and $\varphi \circ \tau : E_0 \longrightarrow E_2$ of degrees prime to ℓ.

As input of Algorithms 2 and 3, we denote by:

- $\mathsf{FastSetup}$, the public parameters of FastSQIsignHD, $p = c\ell^f \ell'^{f'} - 1$, ℓ, ℓ', f, f', the exponent e and the elliptic curve $E_0/\mathbb{F}_p$;
- $\mathsf{SecretKey}$, the isogenies $\tau, \tau' : E_0 \longrightarrow E_A$ of degrees $D_\tau = \ell'^{2f'}$ and $D_{\tau'} = \ell^{2f}$ respectively along with their kernel ideals I_τ and $I_{\tau'}$;
- $\mathsf{CommitData}$, the isogeny $\psi : E_0 \longrightarrow E_1$ of degree $D_\psi = \ell'^{2f}$ and its kernel ideal I_ψ;
- $\mathsf{ChallData}$, the isogeny $\varphi : E_A \longrightarrow E_2$ of degree $D_\varphi = \ell'^{f'}$.

Algorithm 2: FastRespond

Data: FastSetup, SecretKey, CommitData and ChallData.
Result: $(\sigma(P_1), \sigma(P_2), q)$, where (P_1, P_2) is a canonically determined basis of $E_1[\ell^f]$ and $\sigma : E_1 \longrightarrow E_2$ is an isogeny of ℓ^e-good degree q prime to ℓ'.

1 $I_\varphi \longleftarrow \mathsf{IsogenyToIdeal}(\varphi, \tau', I_{\tau'})$;
2 $J \longleftarrow \overline{I_\psi} \cdot I_\tau \cdot I_\varphi$;
3 $I \longleftarrow \mathsf{RandomEquivalentIdeal}_{\ell^e}(J)$ and $q \longleftarrow \mathrm{nrd}(I)$;
4 If q is not ℓ^e-good or $q \wedge \ell' \neq 1$, go back to line 3;
5 Compute the canonical basis (P_1, P_2) of $E_1[\ell^f]$;
6 $(\sigma(P_1), \sigma(P_2)) \longleftarrow \mathsf{EvalTorsion}_{\ell^f}(I, P_1, P_2, \psi, \varphi \circ \tau, I_\psi, I_\tau \cdot I_\varphi)$;
7 Return $(\sigma(P_1), \sigma(P_2), q)$;

Algorithm 3: FastVerify

Data: FastSetup, E_1, E_2 and an output R from FastRespond.
Result: Determines if R is a valid response.

1 Try to parse $R := (R_1, R_2, q)$, where $R_1, R_2 \in E_2[\ell^f]$ and $q < \ell^e$ and return False if it fails;
2 If q is not ℓ^e-good or $q \wedge \ell' \neq 1$, return False;
3 Compute the canonical basis (P_1, P_2) of $E_1[\ell^f]$;
4 Find $a_1, a_2 \in \mathbb{Z}$ such that $a_1^2 + a_2^2 = \ell^e - q$ using Cornacchia's algorithm [6];
5 $F \longleftarrow \mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}(E_1, E_2, a_1, a_2, P_1, P_2, R_1, R_2)$;
6 **if** $F \neq$ False **then**
7 | Return $\mathsf{IsValid}_{4,\ell^e,\ell^f,\ell'^{f'}}(F, E_1, E_2, a_1, a_2)$;
8 **else**
9 | Return False.
10 **end**

4.2 Finding a Uniformly Random Tight Response Ideal

In this section, we present the algorithm $\mathsf{RandomEquivalentIdeal}_{\ell^e}$ taking a left $\mathcal{O}_0$-ideal J as input and returning an ideal I which is uniformly random among the ideals $I \sim J$ of norm $q < \ell^e$. By [10, Lemma 1], all the equivalent ideals $I \sim J$ are of the form $\chi_J(\alpha) := J\overline{\alpha}/\operatorname{nrd}(J)$ for some $\alpha \in J$ and α determines I up to multiplication by an element of $\mathcal{O}_0^\times$. Besides, the norm of $I = \chi_J(\alpha)$ is $q_J(\alpha) := \operatorname{nrd}(\alpha)/\operatorname{nrd}(J)$, so we need $q_J(\alpha) \leq \ell^e$.

Hence, to sample an ideal $I \sim J$ such that $\operatorname{nrd}(I) \leq \ell^e$ with uniform distribution is equivalent to sample $\alpha \in J \setminus \{0\}$ such that $q_J(\alpha) \leq \ell^e$ with uniform distribution. If we fix a basis of J, we can see q_J as a primitive positive definite integral quadratic form with four variables. By the following lemma, which is a simple generalization of [41, Lemma 3.3], we can sample uniformly $\alpha \in J$ such that $q_J(\alpha) \leq \ell^e$. $\mathsf{RandomEquivalentIdeal}_{\ell^e}$ calls this procedure to get $\alpha \in J$ uniform and rejects the result if $\alpha = 0$. Then the distribution of α is still uniform but in $J \setminus \{0\}$. The proofs of the two following lemmas can be found in [9, § E.2].

Lemma 11. *Let f be a primitive positive definite integral quadratic form in k variables and let $\rho > 0$. Then there exists an algorithm that samples uniformly random elements from the set*

$$\{x \in \mathbb{Z}^k \mid f(x) \leq \rho\}$$

in polynomial time in $\log(\rho)$ and the length of f (namely, the maximal number of bits of the coefficients of f). This algorithm runs in exponential time in k.

For $\mathsf{RandomEquivalentIdeal}_{\ell^e}(J)$ to terminate, we need to find $\alpha \in J \setminus \{0\}$ such that $q_J(\alpha) \leq \ell^e$. For such an α to exist, we need $\ell^e = \Omega(\sqrt{p})$ according to the following lemma (Lemma 12).

Lemma 12. *Let $\mathcal{O}$ be a maximal order and J be a left $\mathcal{O}$-ideal. Then there exists $\alpha \in J$ such that $q_J(\alpha) \leq 2\sqrt{2p}/\pi$.*

In the procedure $\mathsf{FastRespond}$, we reject the results of $\mathsf{RandomEquivalentIdeal}_{\ell^e}$ whose norm is not ℓ^e-good or divisible by ℓ'. If it terminates, this rejection sampling outputs ideals which are uniformly random among the targeted ones, as desired. However, we can only give a heuristic argument for the termination. Assuming that $q_J(\alpha)$ behaves like a random integer, we should expect to find a suitable $\alpha \in J$ with probability $O(1/\log(p))$. Hence, taking ℓ^e a few bits over $\sqrt{p}$ might be sufficient. For that reason, in our choice of parameters, we only have accessible ℓ^f-torsion with $\ell^f < \sqrt{p} < \ell^e$ (see Section 3.1). Proving formally that we can always find an ℓ^e-good value of $q_J(\alpha)$ would certainly require to increase ℓ^e by a lot. As [35] indicates, we should expect lower bounds close to $\ell^e = \omega(p^2)$, causing a huge efficiency loss.

4.3 Dividing the Higher Dimensional Isogeny Computation in Two

As explained in Section 4.2, we do not necessarily have enough accessible torsion to compute the whole kernel of the higher dimensional representation of the

response F. In this section, we explain in plain generality how to circumvent this difficulty. Hence, the following discussion applies to both dimension 4 and 8. Let us keep the notations of Section 2.2. Recall that we have the following isogeny

$$F := \begin{pmatrix} \varphi & \widetilde{\psi'} \\ -\psi & \widetilde{\varphi'} \end{pmatrix}, \quad \text{with} \quad \ker(F) = \{(\widetilde{\varphi}(x), \psi'(x)) \mid x \in B[d]\}.$$

To compute F, we need to evaluate $\widetilde{\varphi}$ and ψ' on $B[d]$, so we need to have accessible d-torsion. However, we assume that we only have d'-accessible torsion with $d'|d$.

The idea is to decompose $F = F_2 \circ F_1$ where $F_1 : \mathcal{A} := A \times B' \longrightarrow \mathcal{C}$ and $F_2 : \mathcal{C} \longrightarrow \mathcal{B} := B \times A'$ are respectively d_1 and d_2-isogenies such that $d_1, d_2 | d'$ and to use the following proposition (proved in [9, § E.3]) to compute F_1 and $\widetilde{F_2}$ to infer F.

Proposition 13. *Suppose d prime to p so that F is separable. Then:*

(i) *We can always decompose $F = F_2 \circ F_1$, as above.*
(ii) $\ker(F_1) \subseteq \ker(F) \cap \mathcal{A}[d_1]$.
(iii) $\ker(\widetilde{F_2}) \subseteq \ker(\widetilde{F}) \cap \mathcal{B}[d_2] = F(\mathcal{A}[d]) \cap \mathcal{B}[d_2]$.
(iv) *When $\ker(F)$ has rank $g := \dim(\mathcal{A})$, those inclusions are equalities.*

In SQIsignHD, $d_1 = \ell^{e_1}$ and $d_2 = \ell^{e_2}$ with $e = e_1 + e_2$ and we have accessible ℓ^f-torsion such that $f \geq e_1, e_2$. Since $\ker(F)$ has maximal rank $g = 4$ (or 8), we have by point (iv) of the above proposition

$$\ker(F_1) = \ker(F)[\ell^{e_1}] = \{(\widetilde{\alpha_1}(P), \Sigma(P)) \mid P \in E_1^{g/2}[\ell^{e_1}]\}$$

and similarly, $\ker(\widetilde{F_2}) = \ker(\widetilde{F})[\ell^{e_2}] = \{(\alpha_1(P), -\Sigma(P)) \mid P \in E_1^{g/2}[\ell^{e_2}]\}$, with the notations of Section 2.3.

In [9, § F], we give an overview of the higher dimensional isogeny computation required in the procedures $\mathsf{RepresentIsogeny}_{g,\ell^e,\ell^f}$ of our SQIsignHD scheme. We provide a proof of concept `sagemath` implementation in dimension 4. Optimizing this implementation in a low level programming language is left for future works.

4.4 Computing the Response Isogeny Representation

We finally give algorithms to compute the signature representation in dimension 4 using all the ideas presented in Section 4.3 and [9, § F]. We refer to $\mathsf{KernelToIsogeny}_{g,\ell^e}(\mathscr{B}_0)$ as the algorithm computing an ℓ-isogeny chain in dimension g given a basis $\mathscr{B}_0$ of its kernel. We refer to [9, § F] for more details on this algorithm.

In dimension 4, $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ (Algorithm 4) computes basis of $\ker(F_1)$ and $\ker(\widetilde{F_2})$ with $F := F_2 \circ F_1$, as in Section 4.3. Then, it calls $\mathsf{KernelToIsogeny}_{4,\ell^e}$ to obtain F_1 and $\widetilde{F_2}$ as isogeny chains. The ideas are the same in dimension 8.

Algorithm 4: $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$

Data: E_1, E_2, $a_1, a_2 \in \mathbb{Z}$, a basis (P_1, P_2) of $E_1[\ell^f]$ and $(\sigma(P_1), \sigma(P_2))$, where $\sigma : E_1 \longrightarrow E_2$ is a q-isogeny with $a_1^2 + a_2^2 + q = \ell^e$.
Result: An ℓ^{e_1}-isogeny $F_1 : E_1^2 \times E_2^2 \longrightarrow \mathcal{C}$ and a ℓ^{e_2}-isogeny $\widetilde{F_2} : E_1^2 \times E_2^2 \longrightarrow \mathcal{C}$ such that $F(\sigma, a_1, a_2) = F_2 \circ F_1$, with $e_1, e_2 \leq f$ and $e_1 + e_2 = e$.

$e_2 \longleftarrow \lceil e/2 \rceil, e_1 \longleftarrow e - e_2$;
$Q_i \longleftarrow [\ell^{f-e_1}]P_i$, $R_i \longleftarrow [\ell^{f-e_1}]\sigma(P_i)$, $Q'_i \longleftarrow [\ell^{f-e_2}]P_i$, $R'_i \longleftarrow [\ell^{f-e_2}]\sigma(P_i)$ for $i \in \{1, 2\}$;
$\mathscr{B}_0 \longleftarrow \big(([a_1]Q_i, [a_2]Q_i, R_i, 0)_{i\in\{1,2\}}, (-[a_2]Q_i, [a_1]Q_i, 0, R_i)_{i\in\{1,2\}}\big)$;
$\mathscr{C}_0 \longleftarrow (([a_1]Q'_i, -[a_2]Q'_i, -R'_i, 0)_{i\in\{1,2\}}, ([a_2]Q'_i, [a_1]Q'_i, 0, -R'_i)_{i\in\{1,2\}})$;
if *$\mathscr{C}_0$ and $\mathscr{B}_0$ are valid kernels of ℓ^{e_1} and ℓ^{e_2}-isogenies* **then**
  $F_1 \longleftarrow \mathsf{KernelToIsogeny}_{4,\ell^{e_1}}(\mathscr{B}_0)$;
  $\widetilde{F_2} \longleftarrow \mathsf{KernelToIsogeny}_{4,\ell^{e_2}}(\mathscr{C}_0)$;
  Return F_1 and $\widetilde{F_2}$;
else
  Return False;
end

Proposition 14. *Algorithm 4 is correct. Namely, Algorithm 4 returns $F_1, \tilde{F}_2$ such that $F_2 \circ F_1 = F(\sigma, a_1, a_2)$ on entry $a_1, a_2, P_1, P_2, \sigma(P_2)$, $\sigma(P_2)$, where $\sigma : E_1 \longrightarrow E_2$ is a q-isogeny with $a_1^2 + a_2^2 + q = \ell^e$.*

Proof. See [9, § E.4]. □

Remark 4.2. To make sure we have enough accessible torsion, we need $f \geq e_1, e_2$, so that $2f \geq e$. Actually, for $\mathsf{KernelToIsogeny}_{4,\ell^{e_i}}$ to work (with theta coordinates of level 2), we need $4\ell^{e_i}$-torsion points (see [9, § F.3]). Then, when $\ell = 2$, we have $f \geq e_i + 2$, so $2f \geq e + 4$.

4.5 Verification

We describe the verification procedure $\mathsf{IsValid}_4$ taking as input the isogenies F_1 and $\widetilde{F_2}$ outputted by $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ and determining if they represent an isogeny $\sigma : E_1 \longrightarrow E_2$ of degree q. The idea is to check if F_1 and $\widetilde{F_2}$ have the same codomain (computed as principally polarized abelian varieties) and then evaluate $F_2 \circ F_1$ on some points to check that the degree is correct.

The following results (proved in [9, § E.5]) ensure that our verification procedure is correct. In [9, § C], we provide algorithms in dimension 8 achieving similar correctness results.

Proposition 15. *Algorithm 5 is correct. Namely, when given $E_1, E_2, a_1, a_2, F_1, \tilde{F}_2$, if Algorithm 5 returns* True*, then $F_2 \circ F_1$ is an efficient representation of an isogeny $\sigma : E_1 \longrightarrow E_2$ of degree $q = \ell^e - a_1^2 - a_2^2$.*

Corollary 16. *The verification procedure* FastVerify *(Algorithm 3) is correct. Namely, on input (R_1, R_2, q),* FastVerify *returns* True *if and only if (R_1, R_2, q)*

Algorithm 5: $\mathsf{IsValid}_{4,\ell^e,\ell^f\ell'^{f'}}$

Data: Elliptic curves E_1, E_2, integers $a_1, a_2 \in \mathbb{Z}$ and the output $(F_1, \widetilde{F_2})$ of $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}(E_1, E_2, a_1, a_2, *, *, *, *)$.
Result: Determines if $F_2 \circ F_1$ is an efficient repersentation of an isogeny $\sigma : E_1 \longrightarrow E_2$ of degree $q := \ell^e - a_1^2 - a_2^2$.

```
Let (C_1, λ_1) and (C_2, λ_2) be the respective codomains of F_1 and F̃_2;
if (C_1, λ_1) ≠ (C_2, λ_2) then
    Return False;
else
    Find a point Q ∈ E_1 of order ℓ^f ℓ'^{f'};
    Compute compute F_2 as the dual of F̃_2 and T ⟵ F_2 ∘ F_1(Q, 0, 0, 0);
    if T = ([a_1]Q, −[a_2]Q, *, 0) then
        Return True;
    else
        Return False;
    end
end
```

defines an efficient representation of an isogeny $\sigma : E_1 \longrightarrow E_2$ of degree q, where q is ℓ^e-good and prime to ℓ'.

5 Security Analysis

In this section, we prove that the SQIsignHD identification protocol is secure, namely that it is complete, knowledge sound and honest-verifier zero knowledge. Recall that by [39, Theorem 7], it is sufficient to ensure that our signature scheme obtained by Fiat-Shamir transform is universally unforgeable under chosen message attacks in the random oracle model.

Completeness means that a honest execution of the protocol is always accepted by the verifier. This is true by Proposition 14 and by construction of IsValid. Knowledge soundness means that an attacker can only "guess" a response with very low probability. It is proven under the assumption that computing an endomorphism in a supersingular elliptic curve is hard, a well known difficult problem in isogeny based cryptography.

The honest-verifier zero-knowledge property implies that the response does not leak any information on the secret key τ. More precisely, we can simulate transcripts of the identification protocol without using the secret key with the same distribution as real transcripts. To construct such a simulator of SQIsignHD, we need access to an oracle evaluating isogenies of non-smooth degrees. In RigorousSQIsignHD, this oracle is very generic and we do not need any additional hypothesis to prove the zero-knowledge property (hence the name of this version). On the contrary, in FastSQIsignHD, the oracle definition is *ad hoc* and we need an additional heuristic assumption to prove the zero-knowledge property. However, it is very unlikely to build an attack on this assumption as we

argue in Section 5.3 and both oracles do not undermine the knowledge soundess. As previously, this section mainly focuses on FastSQIsignHD and refer to [9, § D] for a complete security analysis of RigorousSQIsignHD.

5.1 Knowledge Soundness

The proof that FastSQIsignHD is knowledge sound is a straightforward special soundess argument identical to the original version of SQIsign [10, Theorem 1]. Namely, we prove that given two transcripts with the same commitment but disctinct challenges, we can find an endomorphism in E_A. This *special soundness* property is sufficient to prove that SQIsignHD satisfies *knowledge soundness* [17, Theorem 6.3.2]. However, note that we have to require the prime ideal norm q to be not only ℓ^e-good but also prime to ℓ' in order to complete the proof.

Proposition 17. *Under the assumption that $q = \deg(\sigma)$ is always prime to ℓ', the* FastSQIsignHD *identification protocol satisfies special soundness. Namely, given two transcripts $(E_1, \varphi, R_1, R_2, q)$ and $(E_1, \varphi', R_1', R_2', q')$ with the same commitment E_1 but different challenges $\varphi \neq \varphi'$, we can extract an efficient representation of a non-scalar endomorphism $\alpha \in \mathrm{End}(E_A)$.*

Proof. Let $(E_1, \varphi, R_1, R_2, q)$ and $(E_1, \varphi', R_1', R_2', q')$ be two FastSQIsignHD transcripts with the same commitment E_1 but different challenges $\varphi \neq \varphi'$. Then, by Corollary 16, (R_1, R_2, q) and (R_1', R_2', q') define efficient representations of isogenies $\sigma : E_1 \longrightarrow E_2$ and $\sigma' : E_1 \longrightarrow E_2'$ of degrees q and q' respectively which are ℓ^e-good and comprime with ℓ'. Knowing $(R_1, R_2) = (\sigma(P_1), \sigma(P_2))$, where (P_1, P_2) is a canonical basis of $E_1[\ell^f]$, we can also find $a_1, a_2 \in \mathbb{Z}$ such that ${a_1}^2 + {a_2}^2 + q = \ell^e$ and apply $\mathsf{RepresentIsogeny}_{4,\ell^e,\ell^f}$ to compute $F := F(\sigma, a_1, a_2)$ by Proposition 14. Then, F provides an efficient representation of $\widehat{\sigma}$.

Hence, we know an efficient representation of $\alpha := \widehat{\varphi'} \circ \sigma' \circ \widehat{\sigma} \circ \varphi \in \mathrm{End}(E_A)$. We now prove that α is not scalar. Indeed, if it was, we would have $\alpha = [\lambda]$ for some $\lambda \in \mathbb{Z}$ and $qq'\ell'^{2f'} = \lambda^2$ where $q := \deg(\sigma)$ and $q' := \deg(\sigma')$ are prime to ℓ'. Hence, $\lambda = \ell'^{f'}\lambda'$ with $\lambda' \in \mathbb{Z}$ prime to ℓ' ($\lambda'^2 = qq'$). It follows that $[q']\widehat{\sigma}\circ\varphi = [\lambda']\widehat{\sigma'}\circ\varphi'$. Since q, q' and λ' are prime to ℓ', we get that $\ker(\varphi) = \ker(\varphi')$ *i.e.* $\varphi = \varphi'$ up to post-composition by an automorphism. Contradiction. This completes the proof. □

For RigorousSQIsignHD, our special soundness argument does not apply because we have no guarantee on q in general. For that reason, we need to come back to the formal definition of knowledge soundess given in [17, Definition 6.3.1]. This analysis is conducted in [9, § D.1].

The previous proof of knowledge would be trivial if it was easy to find an endomorphism. Fortunately, this is a well-known hard problem in isogeny-based cryptography.

Problem 18 (Supersingular Endomorphism Problem)**.** Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find an efficient representation of a non-scalar endomorphism $\alpha \in \mathrm{End}(E)$.

This problem is very similar to [10, Problem 1], except that we do not require the endomorphism to have smooth degree. This does not seem to make the problem easier since the endomorphisms solution to this can be evaluated (which was the reason why smoothness was imposed in the first place). The supersingular endomorphism ring problem (Problem 19) reduces to Problem 18. Problem 19 is notoriously hard and it has been proven it is equivalent to path finding in the supersingular ℓ-isogeny graph [41]. The heuristic reduction from Problem 19 to 18 is given by [12, Algorithm 8]. Basically, if we have an oracle finding endomorphisms of E, we call this oracle until we have found enough endomorphisms to generate $\mathrm{End}(E)$.

Problem 19 (Supersingular Endomorphism Ring Problem)**.** Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find four endomorphisms of E (that we can evaluate) forming a $\mathbb{Z}$-basis of $\mathrm{End}(E)$.

5.2 Heuristic Zero-Knowledge Property

The proof of the zero-knowledge property of SQIsignHD uses an oracle generating isogenies of non-smooth degree. To our knowledge, there is no efficient algorithm implementing such an oracle. Nonetheless, it is believed that access to such an oracle does not affect the hardness of the underlying problem (the endomorphism ring problem, see Section 5.3). In RigorousSQIsignHD, the definition of such an oracle is very natural. In FastSQIsignHD, we add (mild) conditions on the degree to account for the computational constraints imposed by the method in dimension 4. These degree constraints are the main reason why the signatures are represented in dimension 8 instead of 4 in RigorousSQIsignHD. Our proof is limited to FastSQIsignHD in this section and we refer to [9, § D.2] for an analysis of RigorousSQIsignHD.

Definition 20. A *random uniform good degree isogeny oracle* (RUGDIO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and returning an efficient representation of a random isogeny $\sigma : E \longrightarrow E'$ of ℓ^e-good degree prime to ℓ', such that: **(i)** the distribution of E' is uniform in the supersingular isogeny graph and **(ii)** the conditional distribution of σ given E' is uniform among isogenies $E \longrightarrow E'$ of ℓ^e-good degree prime to ℓ'.

In addition to the constraint on the degree of the RUGDIO output, we add constraints on the distributions of isogenies. Those constraints are necessary to construct a simulator of FastSQIsignHD. We already justified that these constraints can be mathematically satisfied, namely that for all supersingular elliptic curves E and E', there exists $\sigma : E \longrightarrow E'$ of ℓ^e-good norm. As explained in Section 4.2, taking ℓ^e slightly bigger than $\sqrt{p}$ by a few bits is heuristically sufficient. Note that to prove the zero-knowledge property, we not only need access to a RUGDIO, but also to make a heuristic assumption on the distribution of the commitment E_1. This assumption is no longer necessary in RigorousSQIsignHD.

Theorem 21. *Assume that the commitment E_1 is computationally indistinguishable from an elliptic curve chosen uniformly at random in the supersingular*

isogeny graph. Then, the FastSQIsignHD *identification protocol is computationally honest-verifier zero knowledge in the RUGDIO model.*

In other words, under this assumption, there exists a random polynomial time simulator $\mathcal{S}$ with access to a RUGDIO that simulates transcripts (E_1, φ, R) with a computationally indistinguishable distribution from the transcripts of the FastSQIsignHD *identification protocol.*

Proof. First, we explain how to construct the simulator $\mathcal{S}$. The simulator starts by generating a challenge $\varphi' : E_A \longrightarrow E_2'$. Then, it applies the RUGDIO on entry E_2' to get an efficient representation of a dual response isogeny $\widehat{\sigma'} : E_2' \longrightarrow E_1'$. We can use this efficient representation to evaluate $\widehat{\sigma'}$ on $E_2'[\ell^f]$ and obtain its degree in polynomial time in $\log(p)$[8]. Then, as explained in the proof of Proposition 17, we can compute a dimension 4 isogeny representation of $\widehat{\sigma'}$, which is also to an efficient representation of σ'. Hence, we can compute $R' := (\sigma'(P_1), \sigma'(P_2), q')$ in polynomial time, where (P_1, P_2) is a canonical basis of $E_1'[\ell^f]$ and $q' := \deg(\sigma')$.

We now prove that the transcripts (E_1', φ', R') of $\mathcal{S}$ are statistically indistinguishable from the transcripts (E_1, φ, R) of the FastSQIsignHD identification protocol. By construction, φ and φ' have the same distribution. Given E_2', by the definition of the RUGDIO, E_1' is uniformly random in the supersingular isogeny graph. Besides, E_1 is statistically close to uniformly random as well by assumption.

Finally, conditionally to E_1' and E_2', $\widehat{\sigma'}$ (represented by R') is uniformly random among the isogenies $E_2' \longrightarrow E_1'$ of ℓ^e-good degree prime to ℓ' by the definition of the RUGDIO. The dual map $\phi \longmapsto \widehat{\phi}$ being a bijection preserving the degree, conditionally to E_1' and E_2', σ' is also uniformly random among the isogenies $E_2' \longrightarrow E_1'$ of ℓ^e-good degree prime to ℓ'. By construction (see Section 4.2), conditionally to E_1 and E_2, σ has the same distribution. This completes the proof. □

It remains to justify that the commitment E_1 is computationally indistinguishable from an elliptic curve chosen uniformly at random in the supersingular isogeny graph. While RigorousCommit satisfies statistical indistinguishability, the variant FastCommit relies on heuristics. Consider the distributions on E_1 induced by the following procedures

1. Return the output E_1 of FastCommit.
2. Generate a uniformly random cyclic endomorphism γ of E_0 of degree $\ell^{2f}\ell'^{2f'}$. Factor it as $\gamma = \hat{\phi}' \circ \phi$ with $\deg(\phi) = \ell^{2f}$. Return the codomain E_1 of ϕ.
3. Generate a uniformly random cyclic isogeny ϕ from E_0 of degree ℓ^{2f}. Let E_1 be its codomain; let m be the number of cyclic isogenies $\phi' : E_0 \to E_1$ of degree $\ell'^{2f'}$. Return E_1 with probability m/M (for some fixed upper bound M on m, for instance $M = (\ell' + 1)\ell'^{2f'-1}$), otherwise resample.
4. Generate a uniformly random cyclic isogeny ϕ from E_0 of degree ℓ^{2f}; return its codomain E_1.

[8] We can compute the norm of σ' on $E_2'[m]$ (which is $\deg(\sigma') \mod m$) for a bunch of small primes m and apply the Chinese remainder theroem.

5. Return a uniformly random elliptic curve E_1.

We argue that each distribution from the list is somewhat close to the next. The difference between 1 and 2 is that in FastCommit, the endomorphism γ is not truly uniform: they follow a distribution biased by the fact that some intermediate result should be easy to factor. Since this property appears somewhat decorrelated from the final distribution of γ it seems plausible to argue that the distribution of γ in 1 is close to the one in 2. The distributions 2 and 3 are actually identical: distribution 3 simulated distribution 2 by rejection sampling. The difference between 3 and 4 is that m is not necessarily a (positive) constant; it is however heuristically expected to be almost a constant: there are about $(\ell'-1)\ell'^{2f'-1}$ possible paths, and about $p/12$ vertices, so we expect about $m \approx 12(\ell'-1)\ell'^{2f'-1}/p$ distinct paths to any fixed vertex. The difference between 4 and 5 is similar, but reasoning about ℓ-paths instead of ℓ'-paths.

Note that the differences at some of these steps are statistically significant. We only argue that they are not computationally detectable, at least when the endomorphism rings are not known.

5.3 On Hardness of the Supersingular Endomorphism Problem with Access to an Auxiliary Oracle

The FastSQIsignHD identification protocol is sound assuming the hardness of the supersingular endomorphism problem 18, and zero-knowledge with respect to a simulator that has access to a RUGDIO (or a RADIO for RigorousSQIsignHD, as defined in [9, § D.2]). For the resulting signature scheme to be secure, one therefore needs to assume that the supersingular endomorphism problem remains hard even when given access to a RUGDIO.

While it currently seems out of reach to prove that the supersingular endomorphism problem is equivalent to the variant with RUGDIO access, let us argue that the RUGDIO indeed does not help. We focus the following discussion on the RUGDIO, but the same arguments apply to the RADIO despite the slightly different distribution.

The RUGDIO allows to generate random isogenies with a chosen domain E. Note that this task is already known to be easy, with isogenies of smooth degree. The RUGDIO only lifts this smoothness restriction and replaces it with other restrictions (ℓ^e-good and prime to ℓ'): it allows to generate random isogenies whose degrees have large prime factors. It does not allow to reach more target curves, nor does it give more control on which specific target to hit: the target curve is uniformly distributed in the supersingular graph, which was already possible with smooth isogenies.

Smoothness of random isogenies has never been an inconvenience in finding endomorphisms. In fact, the current fastest algorithms for this problem only require very smooth degree isogenies, typically a power of 2. The reason is the following: the purpose of constructing a random isogeny from a fixed source is to reach a random target. As very smooth isogenies (even 2-smooth) are sufficient for optimal randomisation, there is no incentive to involve much larger prime

factors. More specifically, the best known strategies to solve the supersingular endomorphism problem [11] have classical time complexity $\tilde{O}(\sqrt{p})$ (and quantum time complexity $\tilde{O}(p^{1/4})$ with a Grover argument [16]) and essentially perform a meet-in-the-middle search in the supersingular isogeny graph. Access to a RUGDIO would allow to use isogenies of a different shape in the search, but would not speed it up, as the probability to find isogenies with matching codomains stays the same. Another illustration that having access to non-smooth degree isogenies does not help is the fact that the discovery of the $\sqrt{\text{élu}}$ algorithm [3] (which dramatically improved the complexity of computing prime degree isogenies) did not affect the state-of-the-art of the supersingular endomorphism problem.

The above arguments support that random isogenies of non-smooth degrees are not more helpful than random isogenies of smooth degrees. Now, one may be concerned that the encoding of the output of the RUGDIO may leak more information than it should. Non-smooth degree isogenies are represented as a component of a higher dimensional isogeny (Section 2.2). This representation is universal, in the sense that any efficient representation of an isogeny can be efficiently rewritten in this form. In particular, this encoding contains no more information than any other efficient representation of the same isogeny.

6 The SQIsignHD Digital Signature Scheme

The SQIsignHD identification protocol that we presented yields a digital signature scheme via the Fiat-Shamir transform. The security of the transform of both versions FastSQIsignHD and RigorousSQIsignHD follows from the analysis conducted in Section 5 and [9, § D], so the digital signature is also secure under the same computational assumptions. Namely, we have seen it is universally unforgeable under chosen message attacks in the random oracle and RADIO or RUGDIO model, assuming the hardness of the endomorphism ring problem. In this section, we present the performance of the signature scheme obtained from FastSQIsignHD.

6.1 Compactness

As explained before, the signature is made of the data $(E_1, q, \sigma(P_1), \sigma(P_2))$, with $q < \ell^e$, $\sigma : E_1 \longrightarrow E_2$ a q-isogeny and (P_1, P_2) a basis of $E_1[\ell^f]$ determined canonically.

E_1 can be entirely determined by its j-invariant $j(E_1) \in \mathbb{F}_{p^2}$. Since any element of $\mathbb{F}_{p^2}$ can be represented by 2 integers in $[\![0\ ;\ p-1]\!]$, storing $j(E_1)$ takes approximately $2\log_2(p) \simeq 4\lambda$ bits, given that $p = \Theta(2^{2\lambda})$ (where λ is the security level). Similarly, $q < \ell^e \simeq \sqrt{p}$, so q is an integer of $1/2\log_2(p) \simeq \lambda$ bits.

The points $\sigma(P_1)$ and $\sigma(P_2)$ need not be represented explicitely with coordinates in $\mathbb{F}_{p^2}$. They can be compressed. Indeed, if we generate a canonical basis (Q_1, Q_2) of $E_2[\ell^f]$, then we may write $\sigma(P_1) = a_1Q_1 + b_1Q_2$ and $\sigma(P_2) = a_2Q_1 + b_2Q_2$ with $a_1, b_1, a_2, b_2 \in \mathbb{Z}/\ell^f\mathbb{Z}$. Storing the scalars a_1, b_1, a_2, b_2 requires $4f$ bits (assuming $\ell = 2$, which will be the case in practice).

Actually, we can gain f bits by omitting one of the scalars a_1, b_1, a_2, b_2 if we use the Weil pairing. Indeed, we have

$$e_{\ell^f}(\sigma(P_1), \sigma(P_2)) = e_{\ell^f}(P_1, P_2)^q.$$
$$e_{\ell^f}(\sigma(P_1), \sigma(P_2)) = e_{\ell^f}(a_1Q_1 + b_1Q_2, a_2Q_1 + b_2Q_2) = e_{\ell^f}(Q_1, Q_2)^{a_1b_2 - b_1a_2}.$$

Since (P_1, P_2) and (Q_1, Q_2) are basis of $E_1[\ell^f]$ and $E_2[\ell^f]$ respectively, $e_{\ell^f}(P_1, P_2)$ and $e_{\ell^f}(Q_1, Q_2)$ are both primitive ℓ^f-th root of unity. Hence, we may find $k \in (\mathbb{Z}/\ell^f\mathbb{Z})^\times$ such that $e_{\ell^f}(P_1, P_2) = e_{\ell^f}(Q_1, Q_2)^k$, and we must have

$$a_1b_2 - b_1a_2 \equiv kq \mod \ell^f \tag{2}$$

Since $\ell^f | p - 1$, the ℓ^f-th Weil pairing takes values in $\mathbb{F}_p^*$, so we find k easily by solving a discrete logarithm problem in a subgroup of order ℓ^f of $\mathbb{F}_p^*$ by Pohlig-Hellman [29] techniques (which apply since $p - 1$ is smooth).

Since q is prime to ℓ, $\sigma(P_1)$ have order ℓ^f so either a_1 or b_1 is invertible modulo ℓ^f. If a_1 is invertible, we can recover b_2 from the other scalars using equation 2 and we can recover a_2 otherwise. Hence we only need 3 scalars among 4.

We can make the representation of $\sigma(P_1)$ and $\sigma(P_2)$ even more compact. Indeed, by Remark 4.2 the ℓ^e-isogeny F representing σ can be computed as long as $2f \geq e + 4$. But in FastSQIsignHD, $f \simeq e \simeq \lambda$ so we may use points of ℓ^{f_1}-torsion with $f_1 := \lceil e/2 \rceil + 3$ instead of points of ℓ^f-torsion. This reduces the storage cost of $\sigma(P_1)$ and $\sigma(P_2)$ from $3f \simeq 3\lambda$ to $3f_1 \simeq 3/2\lambda$.

On the whole, we can represent the signatures with $s = 13/2\lambda + O(\log(\lambda))$ bits if we use the compression and decompression algorithms given by Algorithms 6 and 7, breaking the previous record of SQIsign. Indeed, in SQIsign, the kernels of the signature isogeny σ (of degree $p^{15/4}$) and of the dual of the challenge (of degree $\sqrt{p}$) need to be transmitted so we get a signature of size $s = 17/2\lambda + O(\log(\lambda))$ at least.

Example: For NIST-I security level ($\lambda = 128$ bits), we can choose the parameters $p = 13 \cdot 2^{126}3^{78} - 1$, $e = 142$ and $f_1 = 73$. The total signature size in SQIsignHD is $2\lceil \log_2(p) \rceil + e + 3f_1 + 1 = 870$ bits or 109 bytes. SQIsign signatures took 177 bytes in the NIST-I implementation [37].

We can use the same techniques in dimension 8 but we output signatures of size $s = 14\lambda + O(\log(\lambda))$ instead of $13/2\lambda + O(\log(\lambda))$ since e is bigger ($\ell^e = \Theta(p^2)$). Details may be found in [9, § C.3].

6.2 Time Efficiency

Low Signing Time (and Key Generation Time). In the latest version of SQIsign [13], the signature time was dominated by the computation of 30 T-isogenies with $T \simeq p^{5/4}$. Each T-isogeny is rather slow as T typically has prime factors as large as a few thousands. FastSQIsignHD signing only requires a handful of ℓ^f and $\ell'^{f'}$-isogenies, where typically $\ell = 2$ and $\ell' = 3$. Computing these isogenies is orders of magnitude faster than a SQIsign [13] signature.

Algorithm 6: Compression

Data: $E_1, E_2, q, P_1, P_2, \sigma(P_1), \sigma(P_2)$, where $q < \ell^e$, $\sigma : E_1 \longrightarrow E_2$ a q-isogeny and (P_1, P_2) a basis of $E_1[\ell^{f_1}]$ determined canonically ($f_1 := \lceil e/2 \rceil + 3$).
Result: A word of length $2\lceil \log_2(p) \rceil + e + 3f_1$ bits (assuming $\ell = 2$).

Compute $j(E_1) \in \mathbb{F}_{p^2}$;
Let ζ be a canonical generator of $\mathbb{F}_{p^2}$. Write $\zeta := n_1 + n_2\zeta$ where $n_1, n_2 \in \mathbb{F}_p$ are represented by integers in $[\![0\ ;\ p-1]\!]$ of length $\lceil \log_2(p) \rceil$ bits each;
Compute the canonical basis (Q_1, Q_2) of $E_2[\ell^{f_1}]$;
Find $k \in (\mathbb{Z}/\ell^{f_1}\mathbb{Z})^\times$ such that $e_{\ell^{f_1}}(P_1, P_2) = e_{\ell^{f_1}}(Q_1, Q_2)^k$;
Find $a_1, b_1, a_2, b_2 \in \mathbb{Z}/\ell^{f_1}\mathbb{Z}$ such that $\sigma(P_1) = a_1Q_1 + b_1Q_2$ and $\sigma(P_2) = a_2Q_1 + b_2Q_2$;
if $\ell \not| a_1$ **then**
  Return $\|n_1\|n_2\|q\|a_1\|b_1\|b_2\|$;
else
  Return $\|n_1\|n_2\|q\|a_1\|b_1\|a_2\|$;
end

We have implemented FastSQIsignHD in C, based on the implementation of SQIsign [37]. The signature takes an average time of 28 ms at the 128 bits security level on an Intel(R) Core(TM) i5-1335U 4600MHz CPU (average over 1000 signature computations). Key generation takes 70 ms on average on the same CPU. While signature is already close to ten times faster than the fastest implementations of SQIsign [13, 23], we refrain from reporting a detailed clock-cycle comparison, as the bottleneck of our implementation has shifted from isogeny computations to a variety of steps which have not been optimised (as they were negligible in former SQIsign implementations). Most notably, about 29% of the FastSQIsignHD signing time is spent computing two discrete logarithms; a 4.8× speedup is reported in [23] using Tate pairings for comparable discrete logarithm computations. Another 20% is spent solving a quaternion norm equation [22, Algorithm 4], a step that has not been the focus of much attention. In addition, contrary to former implementations of SQIsign, our implementation is purely in C, with no assembly-optimised field arithmetic. Providing a completely optimized implementation is left for future works, yet this first implementation is already compelling.

Verification Time. This efficiency gain in the signature is made at the expense of the verification time where a 4-dimensional ℓ^e-isogeny has to be computed. Of course ℓ-isogenies in dimension 4 are expected to be slower to compute than in dimension 1. Nonetheless, we only have to compute two chains of ℓ-isogenies of length $1/4 \log_\ell(p)$, whereas the verifier had to compute an ℓ-isogeny chain of size $15/4 \log_\ell(p)$ in the last version of SQIsign [13]. Furthermore, our choice of parameters allows for more efficient field arithmetic.

Our experimental `sagemath` implementation provides an upper bound on the verification time: for 128 bits of security the verification takes around 600 ms on the same CPU as above. We expect this time to be significantly reduced in the

Algorithm 7: Decompression

Data: A word w of $2\lceil \log_2(p) \rceil + e + 3f_1$ bits ($\ell := 2, f_1 := \lceil e/2 \rceil + 3$), the public key E_A, a message m, and hash functions Φ and H used to generate the challenge in the Fiat-Shamir transform (see [9, § A.1]).

Result: $E_1, E_2, q, P_1, P_2, \sigma(P_1), \sigma(P_2)$, where $q < \ell^e$, $\sigma : E_1 \longrightarrow E_2$ is a q-isogeny and (P_1, P_2) is a basis of $E_1[\ell^{f_1}]$ determined canonically.

Parse $\|n_1\|n_2\|q\|a_1\|b_1\|c_2\| \longleftarrow w$;
Set $j \longleftarrow n_1 + n_2\zeta$, where ζ is the canonical generator of $\mathbb{F}_{p^2}$;
Compute E_1 of j-invariant $j(E_1) = j$;
Recover the commitment $\varphi \longleftarrow \Phi(E_1, H(E_1, m))$. Let E_2 be the codomain of φ;
Compute the canonical basis (P_1, P_2) of $E_1[\ell^{f_1}]$ and the canonical basis (Q_1, Q_2) of $E_2[\ell^{f_1}]$;
Find $k \in (\mathbb{Z}/\ell^{f_1}\mathbb{Z})^\times$ such that $e_{\ell^{f_1}}(P_1, P_2) = e_{\ell^{f_1}}(Q_1, Q_2)^k$;
if $\ell \nmid a_1$ **then**
    $a_2 \longleftarrow c_2$;
    Find $b_2 \in \mathbb{Z}/\ell^{f_1}\mathbb{Z}$ such that $a_1 b_2 - b_1 a_2 \equiv kq \mod \ell^{f_1}$;
else
    $b_2 \longleftarrow c_2$;
    Find $a_2 \in \mathbb{Z}/\ell^{f_1}\mathbb{Z}$ such that $a_1 b_2 - b_1 a_2 \equiv kq \mod \ell^{f_1}$;
end
Return $E_1, E_2, q, P_1, P_2, a_1Q_1 + b_1Q_2, a_2Q_1 + b_2Q_2$;

future: new optimisations remain to be implemented, and more importantly a low level implementation should lead to a significant gain. Currently, the time spent on verification is as follows: around 60 ms for the challenge computation[9], 510 ms for the two dimension 4 2^{71}-isogenies giving F, and 30 ms for the image of a point through F.

References

[1] K. Ahrens. *Sieving for large twin smooth integers using single solutions to Prouhet-Tarry-Escott*. Cryptology ePrint Archive, Paper 2023/219. 2023. URL: https://eprint.iacr.org/2023/219.

[2] R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi. "Key compression for isogeny-based cryptosystems". In: *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*. Xi'an, China: ACM, 2016, pp. 1–10.

[9] Which essentially consists in computing a chain of 3-isogenies in dimension 1. The signature also needs to compute similar isogenies, and in this case the low level C implementation only takes 1-2 ms, which shows the potential of improvements of writing a low level implementation of the verification.

[3] D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. "Faster computation of isogenies of large prime degree". In: *Open Book Series, Proceedings of the Fourteenth Algorithmic Number Theory Symposium – ANTS XIV* 4.1 (2020), pp. 39–55.

[4] G. Bruno, M. C.-R. Santos, C. Costello, J. K. Eriksen, M. Naehrig, M. Meyer, and B. Sterner. *Cryptographic Smooth Neighbors*. Cryptology ePrint Archive, Paper 2022/1439. 2022. URL: https://eprint.iacr.org/2022/1439.

[5] W. Castryck and T. Decru. "An Efficient Key Recovery Attack On SIDH". In: *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Lyon, France: Springer-Verlag, 2023, pp. 423–447. ISBN: 978-3-031-30588-7. URL: https://doi.org/10.1007/978-3-031-30589-4_15.

[6] G. Cornacchia. "Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} C_h x^{n-h} y^h = P$". In: *Giornale di matematiche di Battaglini* 46 (1908), pp. 33–90.

[7] C. Costello, M. Meyer, and M. Naehrig. "Sieving for Twin Smooth Integers with Solutions to the Prouhet-Tarry-Escott Problem". In: *Advances in Cryptology – EUROCRYPT 2021*. Ed. by A. Canteaut and F.-X. Standaert. Cham: Springer International Publishing, 2021, pp. 272–301. ISBN: 978-3-030-77870-5.

[8] J.-M. Couveignes. *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Report 2006/291. 2006. URL: https://eprint.iacr.org/2006/291.

[9] P. Dartois, A. Leroux, D. Robert, and B. Wesolowski. *SQIsignHD: New Dimensions in Cryptography*. Cryptology ePrint Archive, Paper 2023/436. 2023. URL: https://eprint.iacr.org/2023/436.

[10] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. "SQIsign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *Advances in Cryptology – ASIACRYPT 2020*. Ed. by S. Moriai and H. Wang. Cham: Springer International Publishing, 2020, pp. 64–93. ISBN: 978-3-030-64837-4.

[11] C. Delfs and S. D. Galbraith. "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$". In: *Designs, Codes and Cryptography* 78.2 (2016), pp. 425–440. DOI: 10.1007/s10623-014-0010-1.

[12] K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison, and C. Petit. "Supersingular Isogeny Graphs and Endomorphism Rings: Reductions and Solutions". In: *Advances in Cryptology – EUROCRYPT 2018*. Ed. by J. B. Nielsen and V. Rijmen. Cham: Springer International Publishing, 2018, pp. 329–368. ISBN: 978-3-319-78372-7.

[13] L. D. Feo, A. Leroux, P. Longa, and B. Wesolowski. "New Algorithms for the Deuring Correspondence - Towards Practical and Secure SQISign Signatures". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by

C. Hazay and M. Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 659–690. URL: https://doi.org/10.1007/978-3-031-30589-4%5C_23.

[14] A. Fiat and A. Shamir. "How To Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Advances in Cryptology — CRYPTO' 86*. Ed. by A. M. Odlyzko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194. ISBN: 978-3-540-47721-1.

[15] S. D. Galbraith, C. Petit, and J. Silva. "Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems". In: *J. Cryptol.* 33.1 (Jan. 2020), pp. 130–175. ISSN: 0933-2790. URL: https://doi.org/10.1007/s00145-019-09316-0.

[16] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866.

[17] C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. 1st. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN: 3642143024.

[18] D. Jao and L. De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography*. Ed. by B.-Y. Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 19–34. ISBN: 978-3-642-25405-5.

[19] E. Kani. "The number of curves of genus two with elliptic differentials". In: *Journal für die reine und angewandte Mathematik* 1997.485 (1997), pp. 93–122. DOI: 10.1515/crll.1997.485.93.

[20] D. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. "On the quaternion - isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17 (June 2014). DOI: 10.1112/S1461157014000151.

[21] J. L. de Lagrange. "Démonstration d'un théoreme d'arithmétique". In: *Nouveau Mémoire de l'Académie Royale des Sciences de Berlin* (1770), pp. 123–133.

[22] A. Leroux. *Quaternion algebras and isogeny-based cryptography*. 2022. URL: http://www.lix.polytechnique.fr/Labo/Antonin.LEROUX/manuscrit_these.pdf.

[23] K. Lin, W. Wang, Z. Xu, and C.-A. Zhao. *A Faster Software Implementation of SQISign*. Cryptology ePrint Archive, Paper 2023/753. 2023. URL: https://eprint.iacr.org/2023/753.

[24] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.5 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243.

[25] D. Lubicz and D. Robert. "Computing separable isogenies in quasi-optimal time". In: *LMS Journal of Computation and Mathematics* 18.1 (2015), pp. 198–216. DOI: 10.1112/S146115701400045X.

[26] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". In: vol. 9. 1. Springer, 2023. DOI: 10.1007/s40993-022-00407-9.
[27] L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. "A Direct Key Recovery Attack on SIDH". In: Springer-Verlag, 2023.
[28] *PARI/GP version 2.13.4*. available from http://pari.math.u-bordeaux.fr/. The PARI Group. Univ. Bordeaux, 2022.
[29] S. C. Pohlig and M. E. Hellman. "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance". In: *IEEE Transactions on information theory* 24.1 (Jan. 1978), pp. 106–110.
[30] P. Pollack and E. Treviño. "Finding the Four Squares in Lagrange's Theorem". In: *Integers* 18A (2018), A15.
[31] J. O. Rabin M. O.; Shallit. "Randomized Algorithms in Number Theory". In: *Communications on Pure and Applied Mathematic* 39.S1 (1986), S239–S256. DOI: 10.1002/cpa.3160390713.
[32] D. Robert. *Evaluating isogenies in polylogarithmic time*. Cryptology ePrint Archive, Paper 2022/1068. 2022. URL: https://eprint.iacr.org/2022/1068.
[33] D. Robert. "Breaking SIDH in Polynomial Time". In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by C. Hazay and M. Stam. Cham: Springer Nature Switzerland, 2023, pp. 472–503. ISBN: 978-3-031-30589-4.
[34] A. Rostovtsev and A. Stolbunov. *Public-Key Cryptosystem Based On Isogenies*. Cryptology ePrint Archive, Report 2006/145. 2006. URL: https://eprint.iacr.org/2006/145.
[35] J. Rouse and K. Thompson. *Quaternary quadratic forms with prime discriminant*. 2022. arXiv: 2206.00412 [math.NT].
[36] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.0)*. 2023. URL: https://www.sagemath.org.
[37] The SQIsign team. *SQIsign*. 2023. URL: https://www.sqisign.org.
[38] J. Vélu. "Isogénies entre courbes elliptiques". In: *Comptes-rendus de l'Académie des Sciences* 273 (July 1971). Available at https://gallica.bnf.fr, pp. 238–241.
[39] D. Venturi and A. Villani. *Zero-Knowledge Proofs and Applications*. May 2015. URL: http://danieleventuri.altervista.org/files/zero-knowledge.pdf.
[40] J. Voight. *Quaternion algebras*. v.0.9.23. Aug. 2020. URL: https://math.dartmouth.edu/~jvoight/quat.html.
[41] B. Wesolowski. "The supersingular isogeny path and endomorphism ring problems are equivalent". In: *FOCS 2021 - 62nd Annual IEEE Symposium on Foundations of Computer Science*. Denver, Colorado, United States, Feb. 2022. URL: https://hal.archives-ouvertes.fr/hal-03340899.
[42] G. Zanon, M. A. Simplicio, G. Pereira, J. Doliskani, and P. L. Barreto. "Faster isogeny-based compressed key agreement". In: *Post-Quantum Cryptography*. Springer International Publishing, 2018, pp. 248–268.