SQIsign2D–West

The Fast, the Small, and the Safer

Andrea Basso[1,2][0000−0002−3270−1069], Pierrick Dartois[3,4][0009−0008−2808−9867], Luca De Feo[2][0000−0002−9321−0773], Antonin Leroux[5,6][0009−0002−3737−0075], Luciano Maino[1][0009−0005−4495−5102], Giacomo Pope[1,7], Damien Robert[3,4][0000−0003−4378−4274], and Benjamin Wesolowski[8][0000−0003−1249−6077]

[1] University of Bristol, Bristol, United Kingdom
[2] IBM Research Europe, Zürich, Switzerland
[3] Univ. Bordeaux, CNRS, INRIA, IMB, UMR 5251, F-33400 Talence, France
[4] INRIA, IMB, UMR 5251, F-33400, Talence, France
[5] DGA-MI, Bruz, France
[6] IRMAR - UMR 6625, Université de Rennes, France
[7] NCC Group, Cheltenham, United Kingdom
[8] ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France

Abstract. We introduce SQIsign2D–West, a variant of SQIsign using two-dimensional isogeny representations.
SQIsignHD introduced four- and eight-dimensional isogeny representations to improve signing times and provable security of SQIsign, at the cost of slower verification. It left open the question of leveraging two-dimensional representations, which we solve here by introducing new algorithmic tools. These lead to a "best-of-both-worlds" scheme: our signing times are only 2× to 3× slower than SQIsignHD but 10× to 15× faster than SQIsign, our security proof rigorously reduces to an assumption similar to the one behind SQIsignHD, and our verification times are the fastest among all present variants of SQIsign. Additionally, like SQIsignHD, SQIsign2D–West favourably scales to high levels of security. Concretely, for NIST level I we achieve signing times of 80 ms and verifying times of 4.5 ms, using optimised arithmetic for the `x86_64` architecture. For NIST level V, we achieve 470 ms for signing and 31 ms for verifying.

Keywords: Isogenies · Post-quantum · Signatures.

1 Introduction

SQIsign [14,9] is a signature scheme based on the conjectured hardness of computing endomorphism rings of supersingular curves. A candidate in the NIST post-quantum cryptography standardisation process, it features the smallest combined size of public key and signature, but it also exhibits one the slowest performances among all candidates.

The SIDH attacks [8,30,39] shook the foundations of isogeny-based cryptography by showing that any isogeny can be efficiently recovered from its

Table 1. Parameter sizes and performance of SQIsign2D–West. Average running times computed using an Intel Xeon Gold 6338 (Ice Lake, 2GHz) using finite field arithmetic optimised for the x64 architecture, turbo boost disabled. See Section 6 for details.

	Sizes (bytes)		Timings (ms)		
	Public key	Signature	Keygen	Sign	Verify
NIST I	66	148	30	80	4.5
NIST III	98	222	85	230	14.5
NIST V	130	294	180	470	31.0

evaluation on a sufficiently large torsion subgroup. Although they marked the end of SIDH/SIKE [25,24] and related schemes, it was not long before the same technique was put to constructive use, notably in the encryption schemes FESTA [4] and QFESTA [31], and in the SQIsignHD [11] variant of SQIsign. The key to all these applications is to *embed* an isogeny of elliptic curves into an isogeny between *higher-dimensional abelian varieties.* The number of dimensions used for the embedding is a key parameter for efficiency: Robert [38] shows that 8 dimensions are always enough, however the cost of representing the higher-dimensional objects grows *exponentially* with the dimension, thus all practical constructions strive to limit the embedding dimension. For example, FESTA and QFESTA manage to restrict themselves to two-dimensional isogenies.

In the same vein, SQIsignHD consists of two sub-variants. The first, RigorousSQIsignHD, uses eight-dimensional isogenies and strives for the best possible provable security but is deemed unpractical. The second, FastSQIsignHD, uses four-dimensional isogenies and compromises on the security proof to achieve the best possible efficiency: the result is a signature scheme with smaller signatures than SQIsign, similarly sized public keys, and significantly faster signing times, but, realistically, *slower verification* owing to the complexity of the four-dimensional representation.

Our contributions. The question of whether it is possible to obtain an improvement over SQIsign by using only two-dimensional isogenies was left open in [11], where a short paragraph commented on the apparent difficulty of this task. We answer this question in the affirmative by introducing SQIsign2D–West.

To achieve this we introduce new tools for computing higher-dimensional isogeny representations in the context of supersingular elliptic curves:

- An algorithm, a simple extension of [31, Algorithm 2], to evaluate a random elliptic isogeny of given degree by embedding it in a two-dimensional isogeny;
- An algorithm, inspired by [34], to translate a quaternion ideal into a two-dimensional representation of the corresponding elliptic curve isogeny. Combined with an algorithm to sample uniformly random quaternion ideals of given norm, it lets the signer uniformly generate isogenies to be transmitted to the verifier.

We give concrete parametrisations of SQIsign2D–West for NIST security levels I, III and V, and implement them, using both generic and optimised modular arithmetic. With key and signature sizes as reported in Table 1, it is comparable to SQIsignHD in terms of bandwidth. Our benchmarks highlight a consistent improvement over SQIsign across the whole spectrum, slightly slower signing performance than FastSQIsignHD but much faster than SQIsign, and *the fastest verification* among all variants of SQIsign. Because prime characteristics in the shape required by SQIsign2D–West are abundant, our variant, unlike SQIsign, does not need a costly search for a "SQIsign-friendly" prime and thus scales seamlessly to high security levels.

Our security proof shows that the security of SQIsign2D–West reduces to the problem of computing the endomorphism ring of a random supersingular curve, in a security model where the attacker is given (classical) access to an oracle computing (higher-dimensional representations of) uniformly random isogenies from a given curve. Hence, compared to SQIsignHD, SQIsign2D–West manages to blend the efficiency gains of FastSQIsignHD with security guarantees similar to RigorousSQIsignHD.

The algorithmic tools we introduce are very flexible, and we considered several variants with different trade offs between provable security and speed. In the main text, we focus on the most secure variant: our security proof follows the blueprint of RigorousSQIsignHD and achieves a reduction to the endomorphism ring problem, provided an isogeny-sampling oracle. By contrast the proof of unforgeability for SQIsign essentially assumes that the signing oracle does not leak information on the secrets. Nevertheless, if one is ready to accept heuristic security (roughly similar to the heuristics used in FastSQIsignHD, so still cleaner than the heuristics of SQIsign), it is possible to modify SQIsign2D–West to obtain even faster signing. We describe this variant in [3, Appendix B]

Related Work. Besides SQIsignHD, there is a growing interest in finding more efficient variants of SQIsign. The recent work AprèsSQI [41] achieves promising savings in verification, while keeping the general structure of SQIsign the same (in particular, AprèsSQI does not use higher dimensional isogenies). The key idea is to use larger extensions of the base field to access more small-order points of the curves, and thus more easy-to-compute isogenies. Nevertheless, because it does not change the overall structure, AprèsSQI suffers from the same problems as SQIsign when it comes to scaling: suitable primes are difficult to find and negatively impact the performance of high security levels.

While preparing this work, we were informed of three concurrent projects with similar objectives. What they have in common is the use of two-dimensional isogeny representations and prime characteristics of similar shape. In particular, they all scale to higher security levels more favourably than SQIsign. We briefly discuss the differences with our work below.

1. In [32], Nakagawa and Onuki first introduce an algorithm to translate ideals to isogenies relying on the computation of two-dimensional isogenies. This algorithm is reminiscent of the techniques used in [19]; in particular, it is significantly different from the one we introduce in Section 3.2. Then, they apply the algorithm to SQIsign. Their proof-of concept implementation in Julia suggests an improvement over SQIsign for key generation and signing,

especially at higher security levels. The proof of security, however, remains heuristic.

2. In [33], Nakagawa and Onuki design SQIsign2D-East, a version of SQIsign where verification requires a computation of a two-dimensional isogeny. This idea shares many similarities with the heuristic version we describe in [3, Appendix B]. At the time of writing, we were not provided an implementation, but we expect SQIsign2D-East to have performance similar to our heuristic version. The main difference between this work and ours is the rigorous proof of security of SQIsign2D–West, which appears difficult to emulate with SQIsign2D-East.

   Very recent work [7] shows that the version of SQIsign2D-East described in [33] did not reach the security levels claimed. The authors of [7] also present a new version of SQIsign2D-East that thwarts their attack. We highlight that our security proof rules out the existence of a similar attack against SQIsign2D-West.
3. In [19], Duparc and Fouotsa introduce another version of SQIsign, called SQIPrime. SQIPrime is the closest to SQIsignHD of all the variants, the main difference being the type of challenge used in the identification protocol. The authors describe two versions, one using two-dimensional isogeny representations and another using four-dimensional ones. The security of either is close to FastSQIsignHD, and thus less rigorous than ours. No implementation of SQIPrime is available at the time, but we expect the four-dimensional variant to perform similarly to SQIsignHD, and the two-dimensional variant to perform similarly to SQIsign2D-East/West, albeit with larger keys and signatures.

For conciseness, from now on we will use SQIsign2D to refer to our protocol, only using SQIsign2D–West when it is needed to distinguish it from other variants.

Plan. We start by reviewing some mathematical background and the fundamentals of SQIsign and its variants in Section 2. In Section 3 we introduce our new algorithms to compute two-dimensional representations of isogenies. Building on these we give in Section 4 a detailed description of the SQIsign2D identification protocol, and provide a formal proof of its security in Section 5. Finally in Section 6 we describe our implementation of SQIsign2D–West and of its heuristic variant, and report on their performance. For space reasons, we describe the aforementioned heuristic variant in [3, Appendix A].

Acknowledgements. The authors were supported by the Agence Nationale de la Recherche under grant ANR-19-CE48-0008 (CIAO), the EPSRC via grant EP/R012288/1 under the RISE (http://www.ukrise.org) programme and the Centre for Doctoral Training (CDT) in Trust, Identity, Privacy and Security in Large-scale Infrastructures, the France 2030 program under grant ANR-22-PETQ-0008 (PQ-TLS), the European Union (ERC, AGATHA CRYPTY, 101116169), and the Swiss National Science Foundation under grant no. 213766 (CryptonIs). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research

Council. Neither the European Union nor the granting authority can be held responsible for them.

2 Preliminaries

In this section, we recall some background knowledge about the Deuring correspondence and isogenies between products of two elliptic curves. We assume some familiarity with elliptic curves and their isogenies and refer to [42,13] for more information.

2.1 The Deuring correspondence

We now give a brief summary of the theory of the Deuring correspondence, following the approach in [29, Chapter 2]. Let $p > 3$ be a prime $\equiv 3 \pmod 4$ and let $\mathcal{B}_{p,\infty}$ be the unique quaternion algebra ramified at p and ∞, i.e. $\mathcal{B}_{p,\infty} = \mathbb{Q}\langle i, j\rangle$, where $i^2 = -1$ and $j^2 = -p$. Given a fractional ideal I, we define its left order as $\mathcal{O}_L(I) = \{\alpha \in \mathcal{B}_{p,\infty} \mid \alpha I \subset I\}$; similarly, one can define its right order $\mathcal{O}_R(I)$.

In [17], Deuring showed an equivalence between maximal orders in $\mathcal{B}_{p,\infty}$ and supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. From now on, we will refer to this equivalence as the *Deuring correspondence*. Under this correspondence, an isogeny $\varphi\colon E_1 \to E_2$ corresponds to a fractional ideal I_φ, where $\mathcal{O}_L(I_\varphi) \cong \mathrm{End}(E_1)$ and $\mathcal{O}_R(I_\varphi) \cong \mathrm{End}(E_2)$. Moreover, $\deg(\varphi) = \mathrm{nrd}(I_\varphi)$.

Given two isogenies $\varphi_1 : E \to E_1$ and $\varphi_2 : E \to E_2$ of coprime degrees, we denote by $[\varphi_1]_*\varphi_2 : E_1 \to E'$ the pushforward isogeny of φ_2 under φ_1, i.e. $\ker([\varphi_1]_*\varphi_2) = \varphi_1(\ker(\varphi_2))$. Equivalently, we define the pushforward of I_{φ_2} under I_{φ_1} as the ideal corresponding to the isogeny $[\varphi_1]_*\varphi_2$. We give a succinct summary of the Deuring correspondence in the following table.

Supersingular elliptic curves	Quaternions
$j(E)$ or $j(E)^p$ supersingular	$\mathcal{O} \cong \mathrm{End}(E)$ maximal order in $\mathcal{B}_{p,\infty}$
$\varphi : E \to E'$	left $\mathcal{O}$-ideal and right $\mathcal{O}'$-ideal I_φ
$\varphi, \psi : E \to E'$	$I_\varphi \sim I_\psi$ $(I_\psi = I_\varphi \alpha)$
$\widehat{\varphi} : E' \to E$	$\overline{I_\varphi}$
$\varphi \circ \psi$	$I_\psi \cdot I_\varphi$
$\theta \in \mathrm{End}(E)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$\mathrm{nrd}(I_\varphi)$

A problem we will face in the following sections is to compute the ideal associated to a given kernel generator. To be more precise, we are given an isogeny $\varphi\colon E_0 \to E$, where we know $\mathcal{O}_0 \cong \mathrm{End}(E_0)$ and its associated ideal I_φ. We also have a point $K \in E$ of smooth order D coprime to $\deg(\varphi)$, which describes the kernel of an isogeny $\psi\colon E \to E'$. Our goal is to compute I_ψ, the ideal corresponding to ψ.

We can accomplish this goal using the algorithm [11, Algorithm 9]. In particular, we first push $\mathcal{O}_0$ under φ via [11, Algorithm 8] and then use [11, Algorithm 9] to retrieve I_ψ. In our use case, we want to avoid running [11, Algorithm 8] and [11, Algorithm 9] on the fly but rather allow some precomputations. Let (P, Q) be a basis $E[D]$ and write K as $[a]P + [b]Q$. In [11, Algorithm 9, Line 1], we have to evaluate a basis $(\beta_1, \beta_2, \beta_3, \beta_4)$ of the right order of I_φ at K. This is equivalent to evaluating $(\beta_1, \beta_2, \beta_3, \beta_4)$ at the basis (P, Q) and then retrieving $\beta_i(K)$ as $[a]\beta_i(P) + [b]\beta_i(Q)$.

In what follows, we use the notation $\{\beta_i(P), \beta_i(Q)\}_{i=1,\dots,4}$ to mean that we have evaluated a basis $(\beta_1, \beta_2, \beta_3, \beta_4)$ of the right order of I_φ at (P, Q) via [11, Algorithm 9]. Additionally, we say that we use the datum $\{\beta_i(P), \beta_i(Q)\}_{i=1,\dots,4}$ to compute I_ψ when we evaluate $(\beta_1, \beta_2, \beta_3, \beta_4)$ at K as $([a]\beta_i(P) + [b]\beta_i(Q))_{i=1,\dots,4}$ and then run the rest of [11, Algorithm 9] to obtain I_ψ.

2.2 Kani's Lemma

Throughout this document we will encounter several different ways to represent isogenies of elliptic curves. We abstract the details into the concept of *isogeny representation*, which essentially says that representing an isogeny is having an efficient algorithm to evaluate it.

Definition 1. *Let $\mathbb{F}_q$ be a finite field. An* isogeny evaluator *$\mathscr{E}$ is a pair of polynomial-time algorithms:*

- *$\mathscr{E}$.**valid**(D) taking as input a string $D \in \{0,1\}^*$ and outputting either a symbol $\perp$ or a triple (E, E', d); in the latter case, E and E' are elliptic curves defined over $\mathbb{F}_q$ and there exists an isogeny $\varphi : E \to E'$ of degree d.*
- *$\mathscr{E}$.**eval**(D, P) taking as input a string $D \in \{0,1\}^*$ and a point $P \in E(\mathbb{F}_{q^k})$; if $\mathscr{E}$.**valid**$(D) = (E, E', d)$ it outputs the image point $\varphi(P) \in E'(\mathbb{F}_{q^k})$, otherwise the output is undefined.*

*In the case that D is of size polynomial in $\log(q)$ and $\log(d)$ and that $\mathscr{E}$.**valid**(D) does not output $\perp$, the string D is called an* efficient representation *of φ (for the evaluator $\mathscr{E}$).*

The article [38] shows that any isogeny can be efficiently represented as the datum of its evaluation on a suitably chosen set of points, then gives an efficient algorithm, akin to an *interpolation-evaluation* algorithm, which, on input an arbitrary point x and the evaluation datum, outputs the value of the isogeny at x. We will only need a special case of this construction, embedding an arbitrary dimension-one n-isogeny into a two-dimensional 2^e-isogeny where $2^e > n$. Let us first recall the notion of (d_1, d_2)-isogeny diamond.

Definition 2. *A (d_1, d_2)-isogeny diamond is a commutative diagram of isogenies:*

$$\begin{tikzcd}
E_0 \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] \arrow[dr, phantom, "\circlearrowright"] & E_1 \arrow[d, "\varphi_2'"] \\
E_2 \arrow[r, "\varphi_1'"'] & E_{12}
\end{tikzcd}$$

where $\varphi_1 : E_0 \to E_1$ and $\varphi'_1 : E_2 \to E_{12}$ are d_1-isogenies, $\varphi_2 : E_0 \to E_2$ and $\varphi'_2 : E_1 \to E_{12}$ are d_2-isogenies.

Remark 3. If d_1 is coprime to d_2, then an isogeny diamond as above is the same thing as a pushforward square from φ_1, φ_2 or a pullback square from φ'_1, φ'_2.

We can now state Kani's Lemma, which is contained in [26, Section 2, Proof of Theorem 2.3].

Theorem 4 (Kani's Lemma). *Let d_1 and d_2 be two coprime positive integers. Given a (d_1, d_2)-isogeny diamond, the isogeny $\Phi : E_0 \times E_{12} \to E_1 \times E_2$ given matricially by*

$$\Phi = \begin{pmatrix} \varphi_1 & \hat{\varphi}'_2 \\ -\varphi_2 & \hat{\varphi}'_1 \end{pmatrix}$$

is a $(d_1 + d_2)$-isogeny between these products of elliptic curves with their principal product polarisation. The kernel of Φ is given by

$$\operatorname{Ker} \Phi = \{(\hat{\varphi}_1(P), \varphi'_2(P)) \mid P \in E_1[d_1 + d_2]\}.$$

Proof. We compute $\tilde{\Phi} \circ \Phi = \begin{pmatrix} \tilde{\varphi}_1 & -\tilde{\varphi}_2 \\ \varphi'_1 & \varphi'_2 \end{pmatrix} \begin{pmatrix} \varphi_1 & \tilde{\varphi'_1} \\ -\varphi_2 & \tilde{\varphi'_2} \end{pmatrix} = \begin{pmatrix} d_1 + d_2 & 0 \\ 0 & d_1 + d_2 \end{pmatrix}$ which shows that Φ is a $(d_1 + d_2)$-isogeny for the product polarisations.

The kernel of Φ, of cardinality $(d_1 + d_2)^2$, is given by the image of $\tilde{\Phi}$ on $(E_1 \times E_2)[d_1 + d_2]$. If d_1 is coprime to d_2, the restriction of $\tilde{\Phi}$ to $E_1[d_1] \times 0_{E_2}$ is injective so its image already spans the full kernel: $\operatorname{Ker} \Phi = \{(\tilde{\varphi}_1(P), \varphi'_2(P)) \mid P \in E_1[d_1 + d_2]\}$. The second equality follows by symmetry, and the third by plugging $P = \varphi_1(P_0)$ with $P_0 \in E_0[d_1 + d_2]$. □

If $\varphi_1 : E_0 \to E_1$ is an isogeny of odd degree d, then if we can construct an arbitrary isogeny $\varphi'_2 : E_1 \to E_{12}$ of degree $2^e - d$, then we can apply Theorem 4 to construct a 2^e-isogeny $\Phi : E_0 \times E_{12} \to E_1 \times E_2$, where $\varphi_2 : E_0 \to E_2$ is given by the pullback of φ'_2 by φ_1.

If our curves have their 2^e-torsion rational (since we work on Kummer lines of supersingular curves over $\mathbb{F}_{p^2}$ this is equivalent to $2^e \mid p \pm 1$), and we know how $\varphi'_2 \circ \varphi_1$ acts on the 2^e-torsion of E_0, we can recover $\operatorname{Ker} \Phi$ efficiently: $\operatorname{Ker} \Phi = \{(dP, \varphi'_2 \circ \varphi_1(P)) \mid P \in E_0[2^e]\}$. We can then use [12] to evaluate Φ efficiently on an arbitrary point of $E_0 \times E_{12}$, this allows to evaluate φ on a point $P \in E_0$ via $\Phi((P, 0_{E_{12}}) = (\varphi_1(P), 0_{E_2}))$. We remark that we do not need to know φ_2 to be able to evaluate Φ. Thus Φ, or more precisely, given a basis (P_2, Q_2) of $E_2[2^e]$, the two generators $(dP_2, \varphi'_2 \circ \varphi_1(P_2))$, $(dP_2, \varphi'_2 \circ \varphi_1(P_2))$ of its kernel, encodes an efficient representation of φ_1.

Of course, if we know how φ_1 acts on $E_0[2^e]$, and we also know how φ'_2 acts on $E_1[2^e]$, then we recover how $\varphi'_2 \circ \varphi_1$ acts on $E_0[2^e]$ and evaluate Φ.

More generally, if we know how φ_1 acts on $E_0[2^e]$, it suffices to be able to construct a isogeny of degree $2^e - d$ starting or ending on E_0, or starting or ending on E_1, to be able to embed efficiently φ_1 into a 2^e-isogeny Φ in dimension 2.

If φ_1 has even degree, we can factorise it as a product of an isogeny of degree 2^t, which can be efficiently evaluated given its kernel, followed by an isogeny of odd degree d, to which we can apply the strategy above.

Remark 5. Kani's Lemma extends to abelian varieties [39, Section 3.2], this is the version used in SQIsignHD to build a response embedded into a dimension four isogeny.

2.3 The SQIsign family

SQIsign and SQIsignHD. SQIsign is a digital signature scheme obtained via the Fiat-Shamir transform [21] of an identification protocol. This protocol is built on the Deuring correspondence between quaternion ideals and isogenies. SQIsign and SQIsignHD mainly differ in the way of making the Deuring correspondence effective. While SQIsign only works with smooth degree isogenies between supersingular elliptic curves, SQIsignHD uses four-dimensional isogenies in the verification process. In the following, we present the main building blocks of SQIsign (and SQIsignHD) identification protocol which will be used in SQIsign2D.

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 $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}$ and publishes E_{pk} as its public key.

Commitment. The prover generates a random secret isogeny $\varphi_{\mathsf{com}} : E_0 \to E_{\mathsf{com}}$ and sends E_{com} to the verifier as its commitment. For the identification protocol to be zero-knowledge (and the derived signature scheme to be secure), E_{com} has to be computationally indistinguishable from a uniformly random elliptic curve in the supersingular isogeny graph.

Challenge. The verifier generates and sends to the prover a random isogeny $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ of smooth degree sufficiently large for φ to have high entropy. The challenge space should have size $\Omega(2^\lambda)$ to ensure λ bits of (soundness) security.

Response. The prover generates and transmits to the verifier an *efficient representation* (as defined in Definition 1) of an isogeny $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ which does not backtrack through φ_{chl} (i.e. $\widehat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}}$ is cyclic).

Verification. The verifier checks that the response returned by the prover correctly represents an isogeny $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ and checks that this isogeny does not backtrack through φ_{chl}. The diagram in Fig. 1 illustrates the relationship between the various isogenies computed by the protocol.

To compute such an efficient representation of φ_{rsp} (that will be called φ_{rsp} by abuse of notations), the prover uses the Deuring correspondence. Returning $\varphi_{\mathsf{rsp}} = \varphi_{\mathsf{chl}} \circ \varphi_{\mathsf{sk}} \circ \widehat{\varphi}_{\mathsf{com}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ would make the scheme insecure. However,

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\varphi_{\mathsf{sk}}"'] \arrow[d, dashed, "\varphi_{\mathsf{com}}"] & E_{\mathsf{pk}} \arrow[d, "\varphi_{\mathsf{chl}}"] \\
E_{\mathsf{com}} \arrow[r, "\varphi_{\mathsf{rsp}}"] & E_{\mathsf{chl}}
\end{tikzcd}$$

— Public
- - Secret

Fig. 1. The SQIsign/SQIsignHD identification protocol. Dashed red lines represent secrets.

the prover can translate $\varphi_{\mathsf{chl}} \circ \varphi_{\mathsf{sk}} \circ \widehat{\varphi}_{\mathsf{com}}$ into an ideal I connecting $\mathrm{End}(E_{\mathsf{com}})$ and $\mathrm{End}(E_{\mathsf{chl}})$, find a random equivalent ideal $I_{\mathsf{rsp}} \sim I$ and translate I_{rsp} into φ_{rsp}.

The ideal $I_{\mathsf{rsp}} \sim I$ is sampled to be relatively easy to translate into an isogeny and with a distribution that ensures one can simulate the response without secret knowledge (zero knowledge property). Those two objectives are in tension and lead to a trade-off between efficiency and rigorous security proofs. In SQIsign, $\mathrm{nrd}(I_{\mathsf{rsp}})$ had to be smooth to make the ideal to isogeny translation possible. The KLPT algorithm [28] was used to find I_{rsp}, resulting in big norms $\mathrm{nrd}(I_{\mathsf{rsp}}) \approx p^{15/4}$, slow ideal to isogeny translation and a very heuristic security proof.

In SQIsignHD [11], the smoothness condition on I_{rsp} is relaxed, allowing for smaller norms, a stronger security proof and a faster response at the expense of the verification time. The idea is to use the higher-dimensional SIDH attack techniques [8,30,39] to represent φ_{rsp}. The prover uses the secret knowledge of $\varphi_{\mathsf{chl}} \circ \varphi_{\mathsf{sk}} \circ \widehat{\varphi}_{\mathsf{com}}$ to evaluate φ_{rsp} on some torsion points. This torsion evaluation (along with $\deg(\varphi_{\mathsf{rsp}})$) is an efficient representation of φ_{rsp} that can be sent to the verifier. To verify the validity of this representation, the verifier computes a four-dimensional isogeny that "embeds" φ_{rsp} by Kani's Lemma. The efficiency of four-dimensional isogeny computation is still an open research question. However, SQIsignHD verification is expected to be slower than SQIsign verification, especially after the latest improvements of AprèsSQI [41]. This was the main motivation for our contribution: accelerate the verification while maintaining a fast signing procedure and strong security proofs (with two-dimensional isogeny computations).

What makes SQIsign2D different from SQIsign and SQIsignHD. As a derivative of SQIsign, SQIsign2D follows the same construction presented above but uses different techniques involving two-dimensional isogeny computations. To perform the verification, we "embed" the response $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ into a two-dimensional 2^r-isogeny. The bottleneck is to find an auxiliary isogeny $\varphi_{\mathsf{aux}} : E_{\mathsf{chl}} \to E_{\mathsf{aux}}$ of degree $2^r - \deg(\varphi_{\mathsf{rsp}})$ to complete the isogeny diamond and apply Kani's Lemma. Additionally, the distribution of φ_{aux} needs to be uniform in order to simplify the proof of the zero knowledge property.

We overcome these issues with an algorithm to sample quaternion ideals of fixed norm with a uniform distribution (called $\mathsf{RandomFixedNormIdeal}$) and another algorithm (called $\mathsf{IdealToIsogeny}$) to translate any left ideal of the order

$\mathcal{O}_0 \cong \text{End}(E_0)$ (with $j(E_0) = 1728$) into an isogeny. IdealToIsogeny uses two-dimensional isogenies and is inspired from the Clapoti algorithm introduced in [34] and RandIsogImages introduced in QFESTA [31, Algorithm 2]. Both RandomFixedNormIdeal and IdealToIsogeny are also used in the key generation and commitment steps to obtain statistically uniform distributions of E_{pk} and E_{com}, with a clear security benefit.

3 Algorithmic building blocks

In this section, we present the main algorithmic building blocks of SQIsign-2D to make the Deuring correspondence effective. We assume we are given a cryptographic size prime $p = c \cdot 2^e - 1$ with $e \in \mathbb{N}$ and $c \in \mathbb{N}$ as small as possible. We can find such p with $c = O(\log(p))$ by Dirichlet's arithmetic progression theorem [18]. We denote by E_0 the supersingular elliptic curve given by $y^2 = x^3 + x$ over $\mathbb{F}_p$ and by $\mathcal{O}_0$ a maximal quaternion order isomorphic to $\text{End}(E_0)$.

First, we briefly introduce FixedDegreeIsogeny, an algorithm to compute the kernel ideal and to evaluate an isogeny of fixed odd degree defined over E_0, which is almost identical to RandIsogImages introduced in QFESTA [31, Algorithm 2]. Then, we present an algorithm IdealToIsogeny to translate any left ideal of $\mathcal{O}_0$ into an efficient representation of isogeny defined over E_0. We finally present an algorithm RandomFixedNormIdeal to sample left ideals of a given maximal order $\mathcal{O} \subseteq \mathcal{B}_{p,\infty}$ of fixed norm with a uniform distribution.

3.1 Generating an arbitrary odd-degree isogeny from E_0

In QFESTA [31], Nakagawa and Onuki introduce an algorithm RandIsogImages to compute non-smooth isogenies originating from E_0. For SQIsign2D, we use their idea and tweak it to construct the FixedDegreeIsogeny algorithm which:

- Takes as input an odd positive integer $u < 2^e$ and a basis (P_0, Q_0) of $E_0[2^e]$.
- Returns the torsion image points $\varphi|_{2^e} = (\varphi(P_0), \varphi(Q_0))$ and the codomain E, where $\varphi \colon E_0 \to E$ is a u-isogeny (as in RandIsogImages), along with its corresponding ideal I (not returned by RandIsogImages).

In the rest of the paper, we will use the notation $\varphi|_N$ to refer to the action of φ on $E_0[N]$. In practice, when we write $\varphi|_N$, we mean $\varphi(P)$ and $\varphi(Q)$, for some basis $\langle P, Q \rangle = E_0[N]$ (as above). A detailed description of FixedDegreeIsogeny is provided in [3, Appendix A.1]. It involves Kani's Lemma and the computation of a 2^e-isogeny.

3.2 Translating a left ideal into an efficient isogeny representation

The state of the art techniques to translate ideals into isogenies impose conditions on the input norm. In SQIsign, the norm had to be smooth and in SQIsignHD, the norm $\text{nrd}(I)$ had to be such that $2^e - \text{nrd}(I)$ can be easily decomposed into a sum of two squares. We now propose an algorithm IdealToIsogeny to translate a left $\mathcal{O}_0$-ideal I of any norm into an isogeny starting from E_0. It is inspired by Page and Robert's work in the context of the Clapoti group action [34]. In Clapoti, the ideal considered is an ideal of a quadratic imaginary order but we can adapt their ideas to quaternion orders.

Let I be a left $\mathcal{O}_0$-ideal. We want to compute the torsion image $\varphi_I|_{2^e}$. The general outline is as follows:

1. Find $I_1, I_2 \sim I$ of coprime norms $d_1, d_2 \approx \sqrt{p}$, and $u, v \in \mathbb{N}^*$ such that $d_1 u + d_2 v = 2^f$ with $f \leq e$ and $d_1 u$ is prime to $d_2 v$.
2. Evaluate isogenies $\varphi_u, \varphi_v : E_0 \to E_u, E_v$ of degrees u and v on $E_0[2^e]$.
3. Use Kani's Lemma on $\varphi_u \circ \widehat{\varphi}_1 : E_I \to E_u$ and $\varphi_v \circ \widehat{\varphi}_2 : E_I \to E_v$, where $\varphi_1, \varphi_2 : E_0 \to E_I$ are the isogenies corresponding to I_1 and I_2 respectively, to compute $\Phi : E_u \times E_v \to E_I \times E'$ that embeds the isogenies $\varphi_1 \circ \widehat{\varphi}_u$ and $\varphi_2 \circ \widehat{\varphi}_v$.
4. Use Φ to compute $\varphi_1 \circ \widehat{\varphi}_u|_{2^e}$ and then $\varphi_u|_{2^e}$ to obtain $\varphi_1|_{2^e}$ and finally obtain $\varphi_I|_{2^e}$.

Step 1. We sample ideals $I_1, I_2 \sim I$ of odd coprime norms d_1 and d_2 until we find positive integers u, v such that $d_1 u + d_2 v = 2^e$. A sufficient (but not necessary) condition for a solution (u, v) to exist is $d_1 d_2 < 2^e$. Hence, the norms d_1 and d_2 should be as small as possible. To find equivalent ideals of such norms, we sample $\beta_i \in I$ with sufficiently small reduced norm and choose $I_i := I\overline{\beta_i}/\operatorname{nrd}(I)$, so that $\operatorname{nrd}(I_i) = q_I(\beta_i) := \operatorname{nrd}(\beta_i)/\operatorname{nrd}(I)$. Minkowski's theorem and [28, Section 3.1] (see also [11, Lemma 12]) ensure that the shortest vector in I has norm $O(\operatorname{nrd}(I)\sqrt{p})$ so we should expect to find $d_1, d_2 \approx \sqrt{p}$ so that $d_1 d_2 \approx p \approx 2^e$ in general. This is not enough to rigorously ensure the existence of u and v.

In [3, Section 3.1], we provide an algorithm ([3, Algorithm 1]) which samples $\beta_1, \beta_2 \in I$ and finds $u, v \in \mathbb{N}^*$ such that $\gcd(uq_I(\beta_1), vq_I(\beta_2)) = 1$ and $uq_I(\beta_1) + vq_I(\beta_2) = 2^e$, where $q_I(\beta) := \operatorname{nrd}(\beta)/\operatorname{nrd}(I)$. This algorithm terminates after $O(\log(p)^2)$ attempts (to sample β_1, β_2) under reasonable heuristics that we motivate therein.

Step 2. We can use FixedDegreeIsogeny [3, Algorithm 7] to evaluate isogenies $\varphi_u, \varphi_v : E_0 \to E_u, E_v$ of degrees u and v on $E_0[2^e]$. Since $u, v \approx \sqrt{p}$, we do not need to compute two-dimensional 2-isogeny chains of full length e in this step, but of half length $e/2$ instead (see [3, Remark 26]).

Remark 8. Alternatively, we may save some time on step 2 at the expense of step 1. Assuming $u = a^2 + b^2$, with $a, b \in \mathbb{Z}$, then we can choose $\varphi_u := [a] + [b]\iota \in \operatorname{End}(E_0)$, with $\iota : (x, y) \in E_0 \mapsto (-x, \sqrt{-1}y) \in E_0$ and similarly for v. Finding u, v in step 1 that can be written easily as a sum of to squares is more costly. There is also a hybrid approach where we only require u (or v) to be a sum of two squares. Experimentally, both of these approaches were on the whole more costly than the proposed method as soon as the ideal given in input is a bit unbalanced (and the smallest possible d_2 is a bit bigger than the expected $\approx \sqrt{p}$). However, we believe that there is room for improvement in our implementation of this search for d_1, d_2, u and v, and this could lead to a different conclusion regarding which variant is the most efficient. Answering this interrogation is left as an interesting open question for future work.

Step 3. We now give more details on steps 3 and 4 inspired by [34]. Consider the following $(d_1 u, d_2 v)$-isogeny diamond:

$$\begin{tikzcd}
E' \arrow[r, "\widehat{\varphi'_v}"] & E_v \\
E_u \arrow[u, "\varphi'_u"] \arrow[r, "\varphi_1 \circ \widehat{\varphi}_u"] \arrow[ur, phantom, "\circlearrowright"] & E_I \arrow[u, "\varphi_v \circ \widehat{\varphi}_2"']
\end{tikzcd}$$

where $\varphi'_u := [\varphi_u \circ \widehat{\varphi}_1]_*(\varphi_v \circ \widehat{\varphi}_2)$ and $\varphi'_v := [\varphi_v \circ \widehat{\varphi}_2]_*(\varphi_u \circ \widehat{\varphi}_1)$ (pushforward isogenies). By Kani's Lemma, we have a 2^f-isogeny:

$$\Phi := \begin{pmatrix} \varphi_1 \circ \widehat{\varphi}_u & \varphi_2 \circ \widehat{\varphi}_v \\ -\varphi'_u & \varphi'_v \end{pmatrix} : E_u \times E_v \to E_I \times E',$$

with kernel:

$$\ker(\Phi) = \{([d_1]\varphi_u(P), \varphi_v \circ \widehat{\varphi}_2 \circ \varphi_1(P)) \mid P \in E_0[2^f]\}.$$

Let $\theta := \widehat{\varphi}_2 \circ \varphi_1 \in \mathrm{End}(E_0)$. By Lemma 9, given I_1 and I_2, if we write $I_1 := I\overline{\beta_1}/\operatorname{nrd}(I)$ and $I_2 := I\overline{\beta_2}/\operatorname{nrd}(I)$ with $\beta_1, \beta_2 \in I$, then we can compute $\theta = \beta_2\overline{\beta_1}/\operatorname{nrd}(I)$ so we can evaluate it easily. By step 2, we also know $\varphi_v|_{2^e}$ and $\varphi_u|_{2^e}$. Hence, we can compute $\ker(\Phi)$ (and evaluate Φ) efficiently. This completes step 3.

Step 4. We first notice that we can evaluate $\varphi_1 \circ \widehat{\varphi}_u$ from the two-dimensional isogeny Φ. This implies we can evaluate φ_1 on $E_0[2^e]$ as follows: $\Phi(\varphi_u(P_0), 0) = ([u]\varphi_1(P_0), *)$ and $\Phi(\varphi_u(Q_0), 0) = ([u]\varphi_1(Q_0), *)$ and we can invert u modulo 2^e since u is odd to get $\varphi_1|_{2^e} = (\varphi_1(P_0), \varphi_1(Q_0))$. To obtain $\varphi_I|_{2^e}$, we rely on the following lemma.

Lemma 9. *For $i \in \{1, 2\}$, if we write $I_i := I\overline{\beta_i}/\operatorname{nrd}(I)$ with $\beta_i \in I$, then $\widehat{\varphi}_i \circ \varphi_I = \beta_i$.*

Proof. Let $i \in \{1, 2\}$. We should expect that $\widehat{\varphi}_i \circ \varphi_I$ corresponds to the ideal $I \cdot \overline{I_i}$, however $\mathcal{O}_R(I) \neq \mathcal{O}_L(\overline{I_i})$ so the product $I \cdot \overline{I_i}$ is not well defined. It is defined up to conjugation of $\overline{I_i}$. We have $\mathcal{O}_L(\overline{I_i}) = \mathcal{O}_R(I_i) = \overline{\beta_i}^{-1}\mathcal{O}_R(I)\overline{\beta_i}$. It follows that $\mathcal{O}_L(\overline{\beta_i} \cdot \overline{I_i} \cdot \overline{\beta_i}^{-1}) = \mathcal{O}_R(I)$ and the ideal corresponding to the isogeny $\widehat{\varphi}_i \circ \varphi_I$ via the Deuring correspondence is:

$$I\overline{\beta_i} \cdot \overline{I_i} \cdot \overline{\beta_i}^{-1} = I\frac{\overline{\beta_i}\beta_i}{\operatorname{nrd}(I)}\overline{I} \cdot \overline{\beta_i}^{-1} = I\overline{I}\frac{\operatorname{nrd}(\beta_i)}{\operatorname{nrd}(I)}\frac{\beta_i}{\operatorname{nrd}(\beta_i)} = \mathcal{O}_0\beta_i.$$

The result follows. □

Following Lemma 9, we have that $[d_1]\varphi_I = \varphi_1 \circ \beta_1$. Since we can evaluate β_1 and φ_1 on $E_0[2^e]$ and d_1 can be inverted modulo 2^e, we can evaluate φ_I on $E_0[2^e]$, completing step 4. Algorithm 2 summarises all these steps.

Algorithm 2 IdealToIsogeny

```
Input: An ideal $I \subseteq \mathcal{O}_0 \cong \mathrm{End}(E_0)$ and a basis $(P_0, Q_0)$ of $E_0[2^e]$.
Output: The image $\varphi_I|_{2^e} = (\varphi_I(P_0), \varphi_I(Q_0))$ of the isogeny $\varphi_I : E_0 \to E_I$ associated to $I$.
Use [3, Algorithm 1] to obtain $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{N}^*$ and $f \leq e$ such that $\gcd(uq_I(\beta_1), vq_I(\beta_2)) = 1$ and $uq_I(\beta_1) + vq_I(\beta_2) = 2^f$
$I_i \leftarrow I\overline{\beta_i}/\mathrm{nrd}(I)$ for $i \in \{1, 2\}$
$\theta \leftarrow \beta_2\overline{\beta_1}/\mathrm{nrd}(I) \in \mathrm{End}(E_0)$    (▷) $\theta := \widehat{\varphi}_2 \circ \varphi_1$
Compute $\varphi_u|_{2^e}$ for a $u$-isogeny $\varphi_u : E_0 \to E_u$    (▷) FixedDegreeIsogeny$(u, P_0, Q_0)$
Compute $\varphi_v|_{2^e}$ for a $v$-isogeny $\varphi_v : E_0 \to E_v$    (▷) FixedDegreeIsogeny$(v, P_0, Q_0)$
Set $K_P \leftarrow [2^{e-f}]([d_1]\varphi_u(P_0), \varphi_v \circ \theta(P_0))$
Set $K_Q \leftarrow [2^{e-f}]([d_1]\varphi_u(Q_0), \varphi_v \circ \theta(Q_0))$
Compute $\Phi : E_u \times E_v \to E_I \times E'$ of kernel $\langle K_P, K_Q \rangle$
Evaluate $\Phi(\varphi_u(P_0), 0) = ([u]\varphi_1(P_0), *)$ and $\Phi(\varphi_u(Q_0), 0) = ([u]\varphi_1(Q_0), *)$ to obtain $\varphi_1|_{2^e}$
Use $\varphi_1|_{2^e}$ to evaluate $\varphi_I = [1/d_1]\varphi_1 \circ \beta_1$ on $(P_0, Q_0)$ and obtain $\varphi_I|_{2^e}$
return $\varphi_I|_{2^e}$
```

3.3 Sampling uniformly at random an ideal of fixed norm

In the protocol, we shall need to uniformly sample at random cyclic isogenies $\varphi : E \to E'$ of fixed degree N several times. When $\mathcal{O} \cong \mathrm{End}(E)$ is known, by the Deuring correspondence this reduces to sampling a left ideal $I \subseteq \mathcal{O}$ of norm N uniformly at random. I is then translated into an isogeny φ (e.g. using Algorithm 2 if $\mathcal{O} = \mathcal{O}_0$). For φ to be cyclic, I has to be *primitive*, that is to say that $I \not\subseteq n\mathcal{O}$ for any integer $n > 1$.

Given a maximal quaternion order $\mathcal{O} \subseteq \mathcal{B}_{p,\infty}$ and an integer N coprime with p, we explain how to sample primitive left ideals $I \subseteq \mathcal{O}$ of norm N. It has been proved that such ideals are in bijection with primitive left ideals of $\mathcal{O}/N\mathcal{O}$ via the reduction modulo N which are themselves in bijection with:

$$\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z}) = \{(x, y) \in (\mathbb{Z}/N\mathbb{Z})^2 \mid \gcd(x, y, N) = 1\}/(\mathbb{Z}/N\mathbb{Z})^*.$$

N being coprime with p, $\mathcal{B}_{p,\infty}$ splits at N and we have an isomorphism $\mathcal{O} \otimes \mathbb{Z}_N \cong M_2(\mathbb{Z}_N)$, where $\mathbb{Z}_N$ is the completion of the localisation of $\mathbb{Z}$ at N. Via the reduction modulo N, we obtain an isomorphism $\varphi_N : \mathcal{O}/N\mathcal{O} \xrightarrow{\sim} M_2(\mathbb{Z}/N\mathbb{Z})$.

Lemma 10 ([27, Lemma 7.2]). *All primitive left ideals of $M_2(\mathbb{Z}/N\mathbb{Z})$ are principal and generated by a matrix*

$$M_{x,y} = \begin{pmatrix} x & y \\ 0 & 0 \end{pmatrix}$$

with $(x:y) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. Hence, we have the following bijection:

$$\begin{aligned} \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z}) &\longrightarrow \{\textit{primitive left ideals } I \subseteq \mathcal{O} \textit{ of norm } N\} \\ (x:y) &\longmapsto \mathcal{O}\varphi_N^{-1}(M_{x,y}) + \mathcal{O}N \end{aligned}$$

As a direct consequence of the above lemma, we obtain:

Lemma 11. *The set of elements $\alpha \in \mathcal{O}$ invertible modulo N acts transitively (by multiplication on the right) on the set of primitive left $\mathcal{O}$-ideals of norm N. Those elements $\alpha \in \mathcal{O}$ invertible modulo N are those of norm coprime with N.*

Proof. Let I be a primitive left $\mathcal{O}$-ideal of norm N. Then, the ideal I corresponds to $(x:y) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ via the bijection of Lemma 10 and is isomorphic to $M_2(\mathbb{Z}/N\mathbb{Z}) \cdot M_{x,y}$ via the composition of the reduction modulo N and φ_N. For any representative $(x,y) \in \mathbb{Z}^2$ of $(x:y) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, we have $\gcd(x,y,N) = 1$ so we may find $u, v \in \mathbb{Z}$ such that $xu + yv \equiv 1 \mod N$, so that:

$$M_{x,y} \begin{pmatrix} u & -y \\ v & x \end{pmatrix} \equiv M_{1,0} \mod N \quad \text{and} \quad \det \begin{pmatrix} u & -y \\ v & x \end{pmatrix} \equiv 1 \mod N$$

Hence, the ideal $M_2(\mathbb{Z}/N\mathbb{Z}) \cdot M_{x,y}$ is in the orbit of $M_2(\mathbb{Z}/N\mathbb{Z}) \cdot M_{1,0}$ under the right action of $GL_2(\mathbb{Z}/N\mathbb{Z})$, and as a consequence, $I/N\mathcal{O}$ is in the orbit of the ideal $I_0/N\mathcal{O} := \mathcal{O}\varphi_N^{-1}(M_{1,0})/N\mathcal{O}$ under the right action of $(\mathcal{O}/N\mathcal{O})^*$.

To conclude, it suffices to prove that the invertible elements of $\mathcal{O}$ modulo N are those of norm coprime with N. If $\alpha \in \mathcal{O}$ is invertible modulo N, there exists $\beta, \gamma \in \mathcal{O}$ such that $\alpha\beta = 1 + N\gamma$, so that

$$\operatorname{nrd}(\alpha)\operatorname{nrd}(\beta) = \operatorname{nrd}(1 + N\gamma) = 1 + N\operatorname{Tr}(\gamma) + N^2\operatorname{nrd}(\gamma) \equiv 1 \mod N,$$

so $\operatorname{nrd}(\alpha)$ is invertible modulo N. Conversely, if $\operatorname{nrd}(\alpha)$ is prime to N, there exists $\lambda \in \mathbb{Z}$ such that $\operatorname{nrd}(\alpha)\lambda \equiv 1 \mod N$. Then, it follows that $\alpha\overline{\alpha}\lambda \equiv 1 \mod N$, so α is invertible modulo N. This completes the proof. □

Lemma 11 ensures that $(\mathcal{O}/N\mathcal{O})^*$ acts transitively on primitive left ideals of norm N by multiplication on the right. Hence, given a primitive left $\mathcal{O}$-ideal I_0 of norm N, if we sample $[\alpha] \in (\mathcal{O}/N\mathcal{O})^*$ uniformly at random, then $I_0\alpha + N\mathcal{O}$ is uniformly random among primitive left $\mathcal{O}$-ideals of norm N.

To obtain such an ideal I_0, we compute $\gamma \in \mathcal{O}$ of norm NM with $\gcd(N, M) = 1$ and without integral factor. This can be done with the algorithms of [29, Section 3.3]. We then consider $I_0 := \mathcal{O}\gamma + \mathcal{O}N$ and sample $[\alpha] \in \mathcal{O}/N\mathcal{O}$ uniformly at random until it is invertible modulo N (which can be checked by computing

$\mathrm{nrd}(\alpha)$). The probability of finding such an α is (by the Chinese remainder theorem):

$$\frac{|GL_2(\mathbb{Z}/N\mathbb{Z})|}{|M_2(\mathbb{Z}/N\mathbb{Z})|} = \prod_{\ell^e || N} \frac{|GL_2(\mathbb{Z}/\ell^e\mathbb{Z})|}{|M_2(\mathbb{Z}/\ell^e\mathbb{Z})|} = \prod_{\ell | N} \left(1 - \frac{1}{\ell}\right)\left(1 - \frac{1}{\ell^2}\right).$$

This quantity is an $\Omega(1/\log\log(N))$ by [23, Theorem 328] so we can find α after $O(\log\log(N))$ tries. These operations are summarised in Algorithm 3.

Algorithm 3 RandomFixedNormIdeal

```
Input: A maximal order $\mathcal{O} \subseteq \mathcal{B}_{p,\infty}$ and an integer $N$ such that $p \nmid N$.
Output: A primitive left $\mathcal{O}$-ideal $I$ of norm $N$ sampled uniformly at random.
Find $\gamma \in \mathcal{O}$ primitive of norm $NM$ with $\gcd(N, M) = 1$   (▷) Using [29, Section 3.3]
repeat
    Sample $u_1, \cdots, u_4 \in [\![0; N-1]\!]$ uniformly at random
    $\alpha \leftarrow \sum_{i=1}^{4} u_i\alpha_i$, where $(\alpha_1, \cdots, \alpha_4)$ is a basis of $\mathcal{O}$
until $\gcd(\mathrm{nrd}(\alpha), N) = 1$
Return $I := \mathcal{O}\gamma\alpha + N\mathcal{O}$
```

4 Detailed description of SQIsign2D

We now present a full description of the SQIsign2D protocol. We start by describing the Σ-protocol underlying SQIsign2D, and then we present the variant of the Fiat-Shamir transform [21] that we rely on to obtain a digital signature protocol.

The protocol uses a field characteristic of the form $p = c \cdot 2^e - 1$, where c is a small cofactor and $\log p \approx 2\lambda$. This is already an improvement over existing SQIsign protocols: since such primes are abundant, it is significantly easier to find parameters, especially at higher security levels, for SQIsign2D than for SQIsign. Compared to SQIsignHD, which uses Montgomery-friendly primes $p = c \cdot 2^e \cdot 3^f - 1$, SQIsign2D primes offer even better opportunities for low-level optimisations, as discussed in Section 6.

4.1 The Σ-protocol

Key generation. During key generation, we sample a random left ideal I_{sk} of $\mathcal{O}_0$ of norm N_{sk} via RandomFixedNormIdeal (Algorithm 3), where N_{sk} is an odd integer of size 4λ. The ideal I_{sk} corresponds to the isogeny $\varphi_{\mathsf{sk}} \colon E_0 \to E_{\mathsf{pk}}$ connecting E_0 to the public key E_{pk}. To be more precise, we compute E_{pk} via IdealToIsogeny.

From a mathematical perspective, the ideal I_{sk} provides enough information to describe the secret isogeny φ_{sk}. However, in order to speed up the response algorithm, we perform additional computations that are stored as internal optimisations – we colour these lines to describe such computations. These internal optimisations are required to obtain a faster translation from the challenge to its corresponding ideal; we will formalise what we mean with “its corresponding ideal” in the paragraph “Response” below.

The gist of these optimisations is to evaluate a basis $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ of the right order $\mathcal{O}_{\mathsf{pk}}$ of I_{sk} at the 2^e-torsion of E_{pk}. This is achieved via [11, Algorithm 9]. The key-generation procedure is formalised in Algorithm 4.

Algorithm 4 Key Generation

```
Output: The public key $\mathsf{pk} = E_{\mathsf{pk}}$ and the secret key $\mathsf{sk} = I_{\mathsf{sk}}$.
$I_{\mathsf{sk}} \leftarrow \mathsf{RandomFixedNormIdeal}(N_{\mathsf{sk}})$
$\varphi_{\mathsf{sk}}|_{2^e}, E_{\mathsf{pk}} \leftarrow \mathsf{IdealToIsogeny}(I_{\mathsf{sk}}, P_0, Q_0)$.
Compute a deterministic basis $(P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ of $E_{\mathsf{pk}}[2^e]$.
Compute a basis $B = (\beta_1, \beta_2, \beta_3, \beta_4)$ of the right order $\mathcal{O}_{\mathsf{pk}}$ of $I_{\mathsf{pk}}$.
Compute the basis $(\tilde{\beta}_1, \tilde{\beta}_2, \tilde{\beta}_3, \tilde{\beta}_4)$ of $\mathrm{End}(E_{\mathsf{pk}})$ corresponding to $B$. (▷) [11, Algorithm 9]
Compute $\mathcal{B} = \left\{\tilde{\beta}_i(P_{\mathsf{pk}}),\ \tilde{\beta}_i(Q_{\mathsf{pk}})\right\}_{i=1,\dots,4}$.
return $\mathsf{pk} := E_{\mathsf{pk}}$ and $\mathsf{sk} := (I_{\mathsf{sk}}, \mathcal{B})$
```

Commitment. The commitment phase is similar to the key-generation computations: as explained above, we first sample a random left ideal I_{com} of $\mathcal{O}_0$ of norm $N_{\mathsf{com}} = \ell_{\mathsf{com}}^n$, for some $n > 0$. In particular, we require $\ell_{\mathsf{com}} > 2^{e_{\mathsf{rsp}}}$, where $2^{e_{\mathsf{rsp}}}$ denotes the largest possible degree of the response isogeny. This condition implies that we can compute the pushforward of any left ideal I of $\mathcal{O}_0$ of norm $< 2^{e_{\mathsf{rsp}}}$ under I_{com}, which is a necessary step in the response computation (see Algorithm 6, Line 9).

One of the outputs of the Commitment algorithm is the curve E_{com} obtained by applying $\mathsf{IdealToIsogeny}$ on I_{com}. Additionally, the algorithm outputs the internal state I_{com}. Similarly to what has been said above, the ideal I_{com} provides enough information to compute the corresponding isogeny $\varphi_{\mathsf{com}} \colon E_0 \to E_{\mathsf{com}}$. However, as an internal optimisation, we also extract and store the isogeny representation $\varphi_{\mathsf{com}}|_{2^e}$. We summarise everything in Algorithm 5.

Algorithm 5 Commitment

```
Output: The commitment curve $E_{\mathsf{com}}$ and the corresponding state $I_{\mathsf{com}}$
$I_{\mathsf{com}} \leftarrow \mathsf{RandomFixedNormIdeal}(N_{\mathsf{com}})$.
$\varphi_{\mathsf{com}}|_{2^e}, E_{\mathsf{com}} \leftarrow \mathsf{IdealToIsogeny}(I_{\mathsf{com}}, P_0, Q_0)$.
return $\mathsf{com} := E_{\mathsf{com}}$ and $\mathsf{st} := (I_{\mathsf{com}}, \varphi_{\mathsf{com}}|_{2^e})$.
```

Challenge. The challenge consists of a positive integer $\mathsf{chl} < 2^{e_{\mathsf{chl}}}$, where e_{chl} is a parameter denoting the size of the challenge space. This integer describes the kernel of the challenge isogeny $\varphi_{\mathsf{chl}} \colon E_{\mathsf{pk}} \to E_{\mathsf{chl}}$:, i.e. $\ker(\varphi_{\mathsf{chl}}) = \langle P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}} \rangle$.

It is worth noting that, although $\deg(\varphi_{\mathsf{chl}}) = 2^e$, the challenge space contains only $2^{e_{\mathsf{chl}}} \ll 2^e$ possible challenges, i.e. we only allow $2^{e_{\mathsf{chl}}}$ possible kernels. Intuitively, the extra length of φ_{chl} is needed to deal with the fact that response isogenies may backtrack with φ_{chl}. This concept is formalised in Theorem 17.

Response. The diagram to keep in mind as we explain the response algorithm is the following one (see Fig. 2), where

- $\varphi_{\mathsf{chl}} \colon E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ is the isogeny described by the challenge chl;
- $\varphi'_{\mathsf{chl}} \colon E_{\mathsf{pk}} \to E^0_{\mathsf{chl}}$ is the portion of φ_{chl} that does not backtrack with the response isogeny;
- $\varphi^{(1)}_{\mathsf{rsp}} \colon E_{\mathsf{com}} \to E'_{\mathsf{chl}}$ is the odd part of the response isogeny;
- $\varphi^{(0)}_{\mathsf{rsp}} \colon E'_{\mathsf{chl}} \to E^0_{\mathsf{chl}}$ is the even, non-backtracking part of the response isogeny;
- $\varphi_{\mathsf{aux}} \colon E_{\mathsf{com}} \to E_{\mathsf{aux}}$ is the auxiliary isogeny needed to embed the isogeny $\varphi^{(1)}_{\mathsf{rsp}}$ into a two-dimensional isogeny;
- $\varphi'_{\mathsf{aux}} \colon E'_{\mathsf{chl}} \to E'_{\mathsf{aux}}$ is the pushforward of φ_{aux} under $\varphi^{(1)}_{\mathsf{rsp}}$.

$$\begin{tikzcd}
E_0 \arrow[rr, "\varphi_{\mathsf{sk}}"] \arrow[d, "\varphi_{\mathsf{com}}"'] & & E_{\mathsf{pk}} \arrow[d, "\varphi'_{\mathsf{chl}}"'] \arrow[ddr, bend left, "\varphi_{\mathsf{chl}}"] & \\
E_{\mathsf{com}} \arrow[r, "\varphi^{(1)}_{\mathsf{rsp}}"] \arrow[d, "\varphi_{\mathsf{aux}}"'] \arrow[dr, phantom, "\circlearrowright"] & E'_{\mathsf{chl}} \arrow[r, "\varphi^{(0)}_{\mathsf{rsp}}"] \arrow[d, "\varphi'_{\mathsf{aux}}"] & E^0_{\mathsf{chl}} \arrow[dr] & \\
E_{\mathsf{aux}} \arrow[r, dotted] & E'_{\mathsf{aux}} & & E_{\mathsf{chl}}
\end{tikzcd}$$

Fig. 2. Response diagram.

The first step is to compute the ideal I_{chl} corresponding to the isogeny $\varphi_{\mathsf{chl}} \colon E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ with kernel $\langle P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}} \rangle$. This is done via [11, Algorithm 9] using the datum $\mathcal{B} = \left\{ \tilde{\beta}_i(P_{\mathsf{pk}}),\, \tilde{\beta}_i(Q_{\mathsf{pk}}) \right\}_{i=1,\ldots,4}$ computed during Key Generation (see Algorithm 4).

The ideal I_{chl} is then employed to compute an isogeny $\varphi_{\mathsf{rsp}} \colon E_{\mathsf{com}} \to E_{\mathsf{chl}}$. To be more precise, the prover first computes an ideal I_{rsp}, equivalent to $\overline{I_{\mathsf{com}}} \cdot I_{\mathsf{sk}} \cdot I_{\mathsf{chl}}$, which is uniformly distributed among the equivalent ideals of norm $< 2^{e_{\mathsf{rsp}}}$. The protocol parameter e_{rsp} is chosen such that the existence of I_{rsp} (or equivalently a connecting isogeny of degree $< 2^{e_{\mathsf{rsp}}}$ between E_{com} and E_{chl}) is guaranteed, which means that $2^{e_{\mathsf{rsp}}}$ must be larger than $2\sqrt{2p}/\pi$. The norm of I_{rsp} must be bounded by $2^{e_{\mathsf{rsp}}}$ so that we can represent $\varphi_{\mathsf{rsp}}\colon E_{\mathsf{com}} \to E_{\mathsf{chl}}$ via a two-dimensional $2^{e_{\mathsf{rsp}}}$-isogeny. In particular, following Kani's Lemma (Theorem 4), the degree of the one-dimensional isogenies represented by such a two-dimensional isogeny must be odd, but this might not be the case for φ_{rsp}. We now explain how to deal with the case of even-degree.

Let us write the norm of I_{rsp} as $\mathrm{nrd}(I_{\mathsf{rsp}}) = q = 2^n q' < 2^{e_{\mathsf{rsp}}}$ for an odd q'. We can think of φ_{rsp} as $\varphi_{\mathsf{rsp}} = \psi \circ \varphi_{\mathsf{rsp}}^{(1)}\colon E_{\mathsf{com}} \to E'_{\mathsf{chl}} \to E_{\mathsf{chl}}$, where $\deg(\varphi_{\mathsf{rsp}}^{(1)}) = q'$ and $\deg(\psi) = 2^n$. It may happen that $\ker(\widehat{\psi}) \cap \ker(\widehat{\varphi_{\mathsf{chl}}})$ is not trivial. Let n_{bt} be the positive integer such that $2^{n_{\mathsf{bt}}} = \#\ker(\widehat{\psi}) \cap \ker(\widehat{\varphi_{\mathsf{chl}}})$. Equivalently, n_{bt} is the largest integer such that $I_{\mathsf{chl}} \cdot \overline{I_{\mathsf{rsp}}} \in 2^{n_{\mathsf{bt}}} \mathcal{O}_{\mathsf{pk}}$.

Let $r' := n - n_{\mathsf{bt}}$ and define $\varphi_{\mathsf{rsp}}^{(0)}\colon E'_{\mathsf{chl}} \to E^0_{\mathsf{chl}}$ to be the isogeny with kernel $\ker(\psi)[2^{r'}]$ – the isogeny $\varphi_{\mathsf{rsp}}^{(0)}$ coincides with the non-backtrack portion of φ_{rsp}. Now, let us factor I_{rsp} as $I^1_{\mathsf{rsp}} \cdot I^0_{\mathsf{rsp}} \cdot I'$, where $\mathrm{nrd}(I^{(1)}_{\mathsf{rsp}}) = q'$ and $\mathrm{nrd}(I^{(0)}_{\mathsf{rsp}}) = 2^{r'}$. The isogenies $\varphi_{\mathsf{rsp}}^{(1)}$ and $\varphi_{\mathsf{rsp}}^{(0)}$ correspond to $I^{(1)}_{\mathsf{rsp}}$ and $I^{(0)}_{\mathsf{rsp}}$, respectively.

Since $\varphi_{\mathsf{rsp}}^{(1)}$ has odd degree, it can be represented via a $2^{e_{\mathsf{rsp}}-n}$-isogeny in dimension 2 by Kani's Lemma. This requires computing an auxiliary isogeny $\varphi'_{\mathsf{aux}}\colon E'_{\mathsf{chl}} \to E'_{\mathsf{aux}}$ of degree $2^{e_{\mathsf{rsp}}-n} - q'$.

As required in Theorem 22, we need the isogeny $\varphi'_{\mathsf{aux}}\colon E'_{\mathsf{chl}} \to E'_{\mathsf{aux}}$ to be uniformly sampled among all the isogenies of degree $2^{e_{\mathsf{rsp}}-n} - q'$. Hence, the prover samples a random left ideal I''_{aux} of $\mathcal{O}_0$ of norm $2^{e_{\mathsf{rsp}}-n} - q'$ and then computes I'_{aux} as the pushforward I'_{aux} of I''_{aux} through $I_{\mathsf{com}} \cdot I^{(1)}_{\mathsf{rsp}}$. The prover can then evaluate $\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}} \circ \varphi_{\mathsf{com}}$ at the 2^e-torsion running $\mathsf{IdealToIsogeny}$ on input $I_{\mathsf{com}} \cdot I^{(1)}_{\mathsf{rsp}} \cdot I'_{\mathsf{aux}}$. Using the datum $\varphi_{\mathsf{com}}|_{2^e}$, the prover has actually access to $\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}}\Big|_{2^e}$.

While a representation of $\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}}$ could act as a valid response, we want the Σ-protocol to be *commitment recoverable*, i.e. it is possible to recompute the commitment curve from a the challenge and corresponding response. This eventually leads to a more compact signature. To achieve such a property, we want the isogeny connecting E_{aux} and E'_{chl}, passing through E_{com}. Thus, the prover has to compute the isogeny $\varphi_{\mathsf{aux}}\colon E_0 \to E_{\mathsf{aux}}$ of degree $2^{e_{\mathsf{rsp}}-n} - q'$ fitting in the following commutative diagram:

$$\begin{tikzcd}
E_{\mathsf{com}} \arrow[r, "\varphi^{(1)}_{\mathsf{rsp}}"] \arrow[d, "\varphi_{\mathsf{aux}}"'] \arrow[dr, phantom, "\circlearrowright"] & E'_{\mathsf{chl}} \arrow[d, "\varphi'_{\mathsf{aux}}"] \\
E_{\mathsf{aux}} \arrow[r, dotted, "\varphi"'] & E'_{\mathsf{aux}}
\end{tikzcd}$$

Such an isogeny can be obtained as one of the components of the two-dimensional $2^{e_{\mathsf{rsp}}-n}$-isogeny Φ with kernel $\{([q]'P, \varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}}(P)) \mid P \in E_{\mathsf{com}}[2^{e_{\mathsf{rsp}}-n}]\}$:

$$\Phi = \begin{pmatrix} \varphi^{(1)}_{\mathsf{rsp}} & -\widehat{\varphi'_{\mathsf{aux}}} \\ \varphi_{\mathsf{aux}} & \widehat{\varphi} \end{pmatrix} : E_{\mathsf{com}} \times E'_{\mathsf{aux}} \to E'_{\mathsf{chl}} \times E_{\mathsf{aux}}.$$

To complete the response algorithm, we still need to compute the non-backtracking part of the response isogeny. Let $\varphi_{\mathsf{rsp}}^{(0)} \colon E'_{\mathsf{chl}} \to E^0_{\mathsf{chl}}$ be such an isogeny, which indeed corresponds to the ideal I^0_{rsp}.

Let $\varphi'_{\mathsf{chl}} \colon E_{\mathsf{pk}} \to E^0_{\mathsf{chl}}$ be the isogeny with kernel $\langle [2^{n_{\mathsf{bt}}}](P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}})\rangle$. In other words, φ'_{chl} is the portion of φ_{chl} that does not backtrack with the response isogeny. Even though φ'_{chl} and $\varphi_{\mathsf{rsp}}^{(0)}$ map onto the same elliptic curve, the curves obtained after an explicit computation of the two isogenies will only be equal up to isomorphism. Thus, the prover additionally has to compute an explicit isomorphism to let the two curves agree.

The explicit computation of the isomorphism between the codomains of φ'_{chl} and $\varphi_{\mathsf{rsp}}^{(0)}$ is required to facilitate the verification. During the verification, the verifier will not compute φ_{chl} but rather compute its non-backtrack portion, i.e. the verifier will only compute the isogeny with kernel $\langle [2^{n_{\mathsf{bt}}}](P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}})\rangle$.

Let $(P_{\mathsf{aux}}, Q_{\mathsf{aux}})$ be a deterministic basis of $E_{\mathsf{aux}}[2^{e_{\mathsf{rsp}}-n_{\mathsf{bt}}}]$ and define

$$P_{\mathsf{chl}} := [2^{e_{\mathsf{rsp}}-n} - q']^{-1}\varphi_{\mathsf{rsp}}^{(0)} \circ \varphi_{\mathsf{rsp}}^{(1)}(P_{\mathsf{aux}}), \quad Q_{\mathsf{chl}} := [2^{e_{\mathsf{rsp}}-n} - q']^{-1}\varphi_{\mathsf{rsp}}^{(0)} \circ \varphi_{\mathsf{rsp}}^{(1)}(Q_{\mathsf{aux}}).$$

The output of the response algorithm consists in $(E_{\mathsf{aux}}, P_{\mathsf{chl}}, Q_{\mathsf{chl}}, r', n_{\mathsf{bt}})$. We collect what has been explained in this paragraph in Algorithm 6.

Verification. On input $(E_{\mathsf{aux}}, P_{\mathsf{chl}}, Q_{\mathsf{chl}}, r', n_{\mathsf{bt}})$, the verifier first computes the isogeny $\varphi_{\mathsf{chl}} : E_0 \to E_{\mathsf{chl}}$ with kernel $\langle [2^{n_{\mathsf{bt}}}](P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}})\rangle$ – this corresponds to the non-backtrack portion of the challenge isogeny as in the previous paragraph. Additionally, they compute $(P_{\mathsf{aux}}, Q_{\mathsf{aux}})$, a deterministic basis of $E_{\mathsf{aux}}[2^{e_{\mathsf{rsp}}-n_{\mathsf{bt}}}]$

If $r' > 0$, it means that the prover has chosen a response isogeny having an even, non-backtrack component. In this case, $[2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]P_{\mathsf{chl}}$ and $[2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]Q_{\mathsf{chl}}$ are linearly dependent, and $\langle [2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]P_{\mathsf{chl}}, [2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]Q_{\mathsf{chl}}\rangle$ is the kernel of the dual of the isogeny $\varphi_{\mathsf{rsp}}^{(0)}$ (Cfr. Fig. 2). The verifier then computes the isogeny $\varphi \colon E_{\mathsf{chl}} \to E'_{\mathsf{chl}}$ with kernel $\langle [2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]P_{\mathsf{chl}}, [2^{e_{\mathsf{rsp}}-r'-n_{\mathsf{bt}}}]Q_{\mathsf{chl}}\rangle$ and updates $E_{\mathsf{chl}} \leftarrow E'_{\mathsf{chl}}$, $P_{\mathsf{chl}} \leftarrow \varphi(P_{\mathsf{chl}})$ and $Q_{\mathsf{chl}} \leftarrow \varphi(Q_{\mathsf{chl}})$.

From Kani's Lemma, it follows that the isogeny Φ with kernel

$$\big\langle \big(P_{\mathsf{chl}}, [2^{r'}]P_{\mathsf{aux}}\big), \big(Q_{\mathsf{chl}}, [2^{r'}]Q_{\mathsf{aux}}\big) \big\rangle$$

maps $E'_{\mathsf{chl}} \times E_{\mathsf{aux}}$ onto $E_{\mathsf{aux}} \times E_{\mathsf{com}}$. This proves the existence of an isogeny connecting E_{com} and E'_{chl}. We summarise these steps in Algorithm 7.

Remark 12 (Technical Remark). In the concrete instantiation, when computing the isogeny Φ with kernel $\mathcal{K} = \big\langle \big(P_{\mathsf{chl}}, [2^{r'}]P_{\mathsf{aux}}\big), \big(Q_{\mathsf{chl}}, [2^{r'}]Q_{\mathsf{aux}}\big) \big\rangle$, we use the formulae in [12]. In particular, in order to avoid the computation of extra square roots in the codomain computation, we use the four torsion above $\mathcal{K}$. As explained in [11, Theorem 56], this also fixes a symplectic four-torsion basis on the codomain, which in turns defines a theta structure.

In the implementation, we always pick the four-torsion above $\mathcal{K}$ such that the codomain is of the form $E'_{\mathsf{aux}} \times E_{\mathsf{com}}$. Therefore, in Algorithm 7, Line 15, we can restrict ourselves to checking that F_2 is isomorphic to E_{com}.

Algorithm 6 Response

```
Input: The public key $E_{\mathsf{pk}}$, the secret key $I_{\mathsf{sk}}, \mathcal{B}$, the commitment $(E_{\mathsf{com}}, \mathsf{com})$, the
    commitment state $I_{\mathsf{com}}, \varphi_{\mathsf{com}}(P_0), \varphi_{\mathsf{com}}(Q_0)$, and the challenge $\mathsf{chl} < 2^{e_{\mathsf{chl}}}$.
Output: $E_{\mathsf{aux}}, P_{\mathsf{aux}}, Q_{\mathsf{aux}}, r', n_{\mathsf{bt}}$
Compute a deterministic basis $(P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ of $E_{\mathsf{pk}}[2^e]$.
Compute the ideal $I_{\mathsf{chl}}$ from $\mathsf{chl}$ and using $\mathcal{B}$.                  (▷) [11, Algorithm 9]
    $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ is the isogeny with kernel $\langle P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}} \rangle$.
Set $J = \overline{I_{\mathsf{com}}} \cdot I_{\mathsf{sk}} \cdot I_{\mathsf{chl}}$.
Compute a uniformly distributed ideal $I_{\mathsf{rsp}}$ equivalent to $J$ of norm $q < 2^{e_{\mathsf{rsp}}}$.
Compute $n$ such that $q = q' \cdot 2^n$, where $q'$ is odd and $n_{\mathsf{bt}} < n$ as the largest integer
    such that $I_{\mathsf{chl}} \cdot \overline{I_{\mathsf{rsp}}} \in 2^{n_{\mathsf{bt}}}\mathcal{O}_{\mathsf{pk}}$.
    // $n_{\mathrm{bt}}$ is the length of the part of the response that backtracks along the challenge isogeny
$r' \leftarrow n - n_{\mathsf{bt}}$.
Factor $I_{\mathsf{rsp}}$ as $I^1_{\mathsf{rsp}} \cdot I^0_{\mathsf{rsp}} \cdot I'$ where $\mathrm{nrd}(I^{(1)}_{\mathsf{rsp}}) = q'$ and $\mathrm{nrd}(I^{(0)}_{\mathsf{rsp}}) = 2^{r'}$.
    // $I^{(1)}_{\mathsf{rsp}}$ is the ideal corresponding to the odd part of the response isogeny $\varphi^{(1)}_{\mathsf{rsp}} : E_{\mathsf{com}} \to E'_{\mathsf{chl}}$,
    and $I^{(0)}_{\mathsf{rsp}}$ is the ideal corresponding to the even part of the response isogeny $\varphi^{(0)}_{\mathsf{rsp}} : E'_{\mathsf{chl}} \to E_{\mathsf{chl}}$.
$I''_{\mathsf{aux}} \leftarrow \mathsf{RandomFixedNormIdeal}(2^{e_{\mathsf{rsp}}-n} - q')$.
Compute $I'_{\mathsf{aux}}$ as the pushforward of $I''_{\mathsf{aux}}$ through $I_{\mathsf{com}} \cdot I^{(1)}_{\mathsf{rsp}}$.
    // $I'_{\mathsf{aux}}$ is the ideal corresponding to an auxiliary isogeny $\varphi'_{\mathsf{aux}} : E'_{\mathsf{chl}} \to E_{\mathsf{aux}}$.
$\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}} \circ \varphi_{\mathsf{com}}\Big|_{2^e}, E'_{\mathsf{aux}} \leftarrow \mathsf{IdealToIsogeny}(I_{\mathsf{com}} \cdot I^{(1)}_{\mathsf{rsp}} \cdot I'_{\mathsf{aux}})$.
$P^0_{\mathsf{com}}, Q^0_{\mathsf{com}} \leftarrow [2^{e-(e_{\mathsf{rsp}}-n)}]\varphi_{\mathsf{com}}(P_0), [2^{e-(e_{\mathsf{rsp}}-n)}]\varphi_{\mathsf{com}}(Q_0)$.
$P^0_{\mathsf{aux}}, Q^0_{\mathsf{aux}} \leftarrow [2^{e-(e_{\mathsf{rsp}}-n)}]\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}} \circ \varphi_{\mathsf{com}}(P_0), [2^{e-(e_{\mathsf{rsp}}-n)}]\varphi'_{\mathsf{aux}} \circ \varphi^{(1)}_{\mathsf{rsp}} \circ \varphi_{\mathsf{com}}(Q_0)$.
Compute $\Phi' : E_{\mathsf{com}} \times E'_{\mathsf{aux}} \to E'_{\mathsf{chl}} \times E_{\mathsf{aux}}$ with kernel $\langle \big([q']P^0_{\mathsf{com}}, P^0_{\mathsf{aux}}\big), \big([q']Q^0_{\mathsf{com}}, Q^0_{\mathsf{aux}}\big) \rangle$
$(\tilde{P}_{\mathsf{chl}}, \tilde{P}_{\mathsf{aux}}) \leftarrow \Phi'(\varphi_{\mathsf{com}}(P_0), 0)$.
$(\tilde{Q}_{\mathsf{chl}}, \tilde{Q}_{\mathsf{aux}}) \leftarrow \Phi'(\varphi_{\mathsf{com}}(Q_0), 0)$.
$E^0_{\mathsf{chl}} \leftarrow E'_{\mathsf{chl}}$.
if $r' > 0$ then
    Compute the isogeny $\varphi^0_{\mathsf{rsp}} : E'_{\mathsf{chl}} \to E^0_{\mathsf{chl}}$ corresponding to $I^0_{\mathsf{rsp}}$.
    $\tilde{P}_{\mathsf{chl}}, \tilde{Q}_{\mathsf{chl}} \leftarrow \varphi^0_{\mathsf{rsp}}(\tilde{P}_{\mathsf{chl}}), \varphi^0_{\mathsf{rsp}}(\tilde{Q}_{\mathsf{chl}})$.
Compute $\varphi'_{\mathsf{chl}} : E_{\mathsf{pk}} \to (E^0_{\mathsf{chl}})'$ of kernel $\langle [2^{n_{\mathsf{bt}}}](P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}}) \rangle$.
Compute the isomorphism $\iota_{\mathsf{chl}} : E^0_{\mathsf{chl}} \to (E^0_{\mathsf{chl}})'$.
$\tilde{P}_{\mathsf{chl}}, \tilde{Q}_{\mathsf{chl}} \leftarrow \iota_{\mathsf{chl}}(\tilde{P}_{\mathsf{chl}}), \iota_{\mathsf{chl}}(\tilde{Q}_{\mathsf{chl}})$.
Compute a deterministic basis $(P_{\mathsf{aux}}, Q_{\mathsf{aux}})$ of $E_{\mathsf{aux}}[2^{e_{\mathsf{rsp}}-n_{\mathsf{bt}}}]$.
Compute $a, b, c, d \in \mathbb{Z}/2^{e_{\mathsf{rsp}}-n_{\mathsf{bt}}}\mathbb{Z}$ such that
        $P_{\mathsf{aux}} = [2^{e-e_{\mathsf{rsp}}+n_{\mathsf{bt}}}]([a]\tilde{P}_{\mathsf{aux}} + [b]\tilde{Q}_{\mathsf{aux}})$   and   $Q_{\mathsf{aux}} = [2^{e-e_{\mathsf{rsp}}+n_{\mathsf{bt}}}]([c]\tilde{P}_{\mathsf{aux}} + [d]\tilde{Q}_{\mathsf{aux}})$.
$P_{\mathsf{chl}}, Q_{\mathsf{chl}} \leftarrow [2^{e-e_{\mathsf{rsp}}+n_{\mathsf{bt}}}]([a]\tilde{P}_{\mathsf{chl}} + [b]\tilde{Q}_{\mathsf{chl}}), [2^{e-e_{\mathsf{rsp}}+n_{\mathsf{bt}}}]([c]\tilde{P}_{\mathsf{chl}} + [d]\tilde{Q}_{\mathsf{chl}})$
return $E_{\mathsf{aux}}, P_{\mathsf{chl}}, Q_{\mathsf{chl}}, r', n_{\mathsf{bt}}$.
```

4.2 The signature protocol

To transform the Σ-protocol in a digital signature, we rely on the Fiat-Shamir transform [21], where the interactive challenge generation is replaced by hashing the commitment, together with the message, to obtain a challenge. However, our protocol differs from a straightforward application of the transform: we rely on the commitment-recoverability property of the underlying Σ-protocol to obtain a smaller signature. Namely, a signature of SQIsign2D consists only of a challenge and the corresponding response. To verify a signature, the verifier

Algorithm 7 Verify

```
Input: The public key $E_{\mathsf{pk}}$, the commitment $E_{\mathsf{com}}$, the challenge $\mathsf{chl}$, the response
    $E_{\mathsf{aux}}, P_{\mathsf{chl}}, Q_{\mathsf{chl}}, r', n_{\mathsf{bt}}$.
Output: $\mathsf{true}$ or $\mathsf{false}$.
 1: Compute a deterministic basis $(P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ of $E_{\mathsf{pk}}[2^e]$.
 2: Compute $\varphi_{\mathsf{chl}} : E_0 \to E_{\mathsf{chl}}$ with kernel $\langle [2^{n_{\mathsf{bt}}}](P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}}) \rangle$.
 3: Compute a deterministic basis $(P_{\mathsf{aux}}, Q_{\mathsf{aux}})$ of $E_{\mathsf{aux}}[2^{e_{\mathsf{rsp}} - n_{\mathsf{bt}}}]$.
 4: if $r' > 0$ then
 5:     if $[2^{e_{\mathsf{rsp}} - n_{\mathsf{bt}} - 1}]Q_{\mathsf{chl}} \neq 0$ then
 6:         $R \leftarrow [2^{e_{\mathsf{rsp}} - n_{\mathsf{bt}} - r'}]Q_{\mathsf{chl}}$
 7:     else
 8:         $R \leftarrow [2^{e_{\mathsf{rsp}} - n_{\mathsf{bt}} - r'}]P_{\mathsf{chl}}$
 9:     Compute $\varphi : E_{\mathsf{chl}} \to E'_{\mathsf{chl}}$ of kernel $\langle R \rangle$.
10:     $E_{\mathsf{chl}} \leftarrow E'_{\mathsf{chl}}$.
11:     $P_{\mathsf{chl}}, Q_{\mathsf{chl}} \leftarrow \varphi(P_{\mathsf{chl}}), \varphi(Q_{\mathsf{chl}})$.
12: Compute $\Phi : E_{\mathsf{chl}} \times E_{\mathsf{aux}} \to F_1 \times F_2$ with kernel $\big\langle \big(P_{\mathsf{chl}}, [2^{r'}]P_{\mathsf{aux}}\big), \big(Q_{\mathsf{chl}}, [2^{r'}]Q_{\mathsf{aux}}\big) \big\rangle$.
13: if the computation of $\Phi$ fails then
14:     return $\mathsf{false}$
15: return $F_2 \cong E_{\mathsf{com}}$
```

recovers the challenge from the signature, checks that the commitment, challenge, and response form a valid transcript for the Σ-protocol, and ensures that the challenge was honestly generated.

For this approach to work, it is necessary that the verifier can extract the commitment from the response. During verification, the verifier first computes the challenge isogeny codomain, and then they obtain the two-dimensional isogeny Φ (see Line 12 of Algorithm 7). The codomain of Φ is either the product $E'_{\mathsf{aux}} \times E_{\mathsf{com}}$ or $E_{\mathsf{com}} \times E'_{\mathsf{aux}}$. While a priori it is not possible to distinguish between the two cases, we rely on a specific method to compute Φ, as explained in Remark 12, that guarantees that the codomain is $E'_{\mathsf{aux}} \times E_{\mathsf{com}}$. Hence, the verifier can extract the commitment curve E_{com} from the codomain of Φ and check the challenge has been honestly generated, i.e. as the output of the hashing of E_{com} and the message to be signed.

5 Security analysis

In this section, we prove that the identification protocol (and thereby the signature scheme obtained by the Fiat–Shamir transform) is secure: it is knowledge-sound and honest-verifier zero-knowledge.

First, note that the key recovery problem for our construction is simply the standard *Supersingular Endomorphism Ring* problem, a foundational problem of isogeny-based cryptography.

Problem 13 (Supersingular Endomorphism Ring problem). Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find four endomorphisms (in efficient representation) which generate the ring $\mathrm{End}(E)$.

The fastest known algorithms for this problem have classical complexity in $\tilde{O}(p^{1/2})$ [16] (see also [35, Theorem 8.8]). The only known quantum speed-up is using Grover's algorithm [22,6], for a quantum complexity in $\tilde{O}(p^{1/4})$.

We prove in Theorem 17 that if $e_{\mathsf{chl}} + e_{\mathsf{rsp}} \leq e$, the protocol has the 2-special soundness property for the language

$$\{(E_{\mathsf{pk}}, \alpha) \mid \alpha \in \mathrm{End}(E_{\mathsf{pk}}) \setminus \mathbb{Z} \text{ in efficient representation}\}.$$

This language corresponds to the *Supersingular One Endomorphism* problem.

Problem 14 (Supersingular One Endomorphism problem). Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find a non-scalar endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$ (in efficient representation).

This One Endomorphism problem is equivalent to the Endomorphism Ring problem [35], i.e., to the key recovery problem for our construction.

Then, we prove in Theorem 22 that if $N_{\mathsf{com}} \geq 2^{4\lambda}$ and $2^{e_{\mathsf{rsp}}} \geq 2\sqrt{2p}/\pi$, then the protocol is statistically honest-verifier zero-knowledge, in a model where the simulator can sample random large-degree isogenies from a given curve (in the classical model, this can only be done efficiently for smooth degree). This model, discussed in Section 5.2, is similar to the security model of SQIsignHD [11].

Impact on parameter selection. In summary, for a security level ensuring λ bits of classical security, one needs to choose a prime $p = \Theta(2^{2\lambda})$. To ensure soundness, one needs $e_{\mathsf{chl}} + e_{\mathsf{rsp}} \leq e$ (recall that $p \approx 2^e$, so $e \approx 2\lambda$). To ensure the statistical honest-verifier zero-knowledge property, one needs $N_{\mathsf{com}} \geq 2^{4\lambda}$ and $2^{e_{\mathsf{rsp}}} \geq 2\sqrt{2p}/\pi$.

5.1 Knowledge soundness

Lemma 15. *Given a commitment E_{com}, a challenge $\mathsf{chl} < 2^{e_{\mathsf{chl}}}$ (generating the challenge isogeny $\varphi_{\mathsf{chl}}\colon E_{\mathsf{pk}} \to E_{\mathsf{chl}}$), and a response $(E_{\mathsf{aux}}, P_{\mathsf{chl}}, Q_{\mathsf{chl}}, r', n_{\mathsf{bt}})$ passing verification, one can extract in polynomial time an efficient representation of an isogeny $\tilde{\sigma} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ of degree at most $2^{e_{\mathsf{rsp}}}$.*

Proof. Write $\psi : E^0_{\mathsf{chl}} \to E_{\mathsf{chl}}$ for the last n_{bt} steps of the challenge isogeny. Let $n = r' + n_{\mathsf{bt}}$. A successful verification ensures that one can extract a $2^{r'}$-isogeny

$$\tilde{\varphi}^{(0)} : \tilde{E}'_{\mathsf{chl}} \to E^0_{\mathsf{chl}},$$

(for some curve $\tilde{E}'_{\mathsf{chl}}$) and an $2^{e_{\mathsf{rsp}}-n}$-isogeny

$$\Phi : \tilde{E}'_{\mathsf{chl}} \times E_{\mathsf{aux}} \to E_{\mathsf{com}} \times \tilde{E}'_{\mathsf{aux}},$$

(for some curve $\tilde{E}'_{\mathsf{aux}}$), in efficient representation. Composing Φ with the inclusion $E'_{\mathsf{chl}} \to E'_{\mathsf{chl}} \times E_{\mathsf{aux}}$ and the projection $E_{\mathsf{com}} \times E'_{\mathsf{aux}} \to E_{\mathsf{com}}$, and taking the dual, we obtain an isogeny $\tilde{\varphi}^{(1)} : E_{\mathsf{com}} \to E'_{\mathsf{chl}}$ of degree at most $2^{e_{\mathsf{rsp}}-r'}$. Let $\tilde{\sigma} = \psi \circ \tilde{\varphi}^{(0)} \circ \tilde{\varphi}^{(1)} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$. It has degree at most

$$\deg(\psi)\deg(\tilde{\varphi}^{(0)})\deg(\tilde{\varphi}^{(1)}) \leq 2^{n_{\mathsf{bt}}} \cdot 2^{e_{\mathsf{rsp}}-n} \cdot 2^{r'} = 2^{e_{\mathsf{rsp}}},$$

which proves the lemma. □

Lemma 16. *Let $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ and $\varphi'_{\mathsf{chl}} : E_{\mathsf{pk}} \to E'_{\mathsf{chl}}$ be two distinct challenges from the same public curve E_{pk}. Then, the largest integer dividing $\varphi'_{\mathsf{chl}} \circ \hat{\varphi}_{\mathsf{chl}} \in \mathrm{Hom}(E_{\mathsf{chl}}, E'_{\mathsf{chl}})$ is smaller than $2^{e_{\mathsf{chl}}}$.*

Proof. Recall that the challenge isogeny φ_{chl} is defined by the kernel $\langle K(\mathsf{chl})\rangle$ with

$$K(\mathsf{chl}) = P_{\mathsf{pk}} + [\mathsf{chl}]Q_{\mathsf{pk}}$$

where $0 \leq \mathsf{chl} < 2^{e_{\mathsf{chl}}}$, and $\langle P_{\mathsf{pk}}, Q_{\mathsf{pk}}\rangle = E_{\mathsf{pk}}[2^e]$. The second challenge isogeny φ'_{chl} is defined similarly by its kernel generator $K(\mathsf{chl}') = P_{\mathsf{pk}} + [\mathsf{chl}']Q_{\mathsf{pk}}$, for some $\mathsf{chl} \neq \mathsf{chl}'$. Since φ_{chl} and φ'_{chl} are cyclic, by [11, Lemma 37] there exists three cyclic isogenies $\varphi_0 : E_{\mathsf{pk}} \to E$, $\varphi_1 : E \to E_{\mathsf{chl}}$ and $\varphi'_1 : E \to E'_{\mathsf{chl}}$ such that $\varphi_{\mathsf{chl}} = \varphi_1 \circ \varphi_0$, $\varphi'_{\mathsf{chl}} = \varphi'_1 \circ \varphi_0$ and $\varphi'_1 \circ \widehat{\varphi}_1$ is cyclic. We call φ_0 the *greatest cyclic factor* of φ_{chl} and φ'_{chl}. It has kernel $\ker(\varphi_0) = \ker(\varphi_{\mathsf{chl}}) \cap \ker(\varphi'_{\mathsf{chl}})$. Since $\varphi'_{\mathsf{chl}} \circ \widehat{\varphi}_{\mathsf{chl}} = [\deg(\varphi_0)]\varphi'_1 \circ \widehat{\varphi}_1$, we see that $\deg(\varphi_0)$ is the largest integer dividing $\varphi'_{\mathsf{chl}} \circ \widehat{\varphi}_{\mathsf{chl}}$ in $\mathrm{Hom}(E_{\mathsf{chl}}, E'_{\mathsf{chl}})$, so we only have to prove that $\deg(\varphi_0) < 2^{e_{\mathsf{chl}}}$.

Let $R \in E_{\mathsf{pk}}$ be a generator of $\ker(\varphi_0)$. Then, $R = [a]K(\mathsf{chl}) = [b]K(\mathsf{chl}')$ for some $a, b \in [\![0; 2^e - 1]\!]$, i.e.,

$$[a-b]P_{\mathsf{pk}} + [a \cdot \mathsf{chl} - b \cdot \mathsf{chl}']Q_{\mathsf{pk}} = 0.$$

Since $(P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ is a basis of $E_{\mathsf{pk}}[2^e]$, it follows that $a - b \equiv 0 \mod 2^e$ so $a = b$ and $a(\mathsf{chl} - \mathsf{chl}') \equiv 0 \mod 2^e$. Since $0 \leq \mathsf{chl} \neq \mathsf{chl}' < 2^{e_{\mathsf{chl}}}$, it follows that $2^{e - e_{\mathsf{chl}}+1} | a$, so that $R \in E_{\mathsf{pk}}[2^{e_{\mathsf{chl}}-1}]$ and $\deg(\varphi_0) \leq 2^{e_{\mathsf{chl}}-1}$. This completes the proof. □

Theorem 17. *If $e_{\mathsf{chl}} + e_{\mathsf{rsp}} \leq e$, then the identification protocol has 2-special soundness for the language*

$$\{(E_{\mathsf{pk}}, \alpha) \mid \alpha \in \mathrm{End}(E_{\mathsf{pk}}) \setminus \mathbb{Z} \textit{ in efficient representation}\}.$$

Proof. Consider two accepting transcripts with the same commitment curve E_{com} but challenge isogenies $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ and $\varphi'_{\mathsf{chl}} : E_{\mathsf{pk}} \to E'_{\mathsf{chl}}$ with distinct kernels. From Lemma 15, we can extract an efficient representation of isogenies $\sigma : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ and $\sigma' : E_{\mathsf{com}} \to E'_{\mathsf{chl}}$, each of degree at most $2^{e_{\mathsf{rsp}}}$. Let $\alpha = \hat{\varphi}'_{\mathsf{chl}} \circ \sigma' \circ \hat{\sigma} \circ \varphi_{\mathsf{chl}} \in \mathrm{End}(E_{\mathsf{pk}})$.

Suppose by contradiction that $\alpha = [m]$ for some $m \in \mathbb{Z}$. We deduce

$$[m] \circ \varphi'_{\mathsf{chl}} \circ \hat{\varphi}_{\mathsf{chl}} = [\deg(\varphi_{\mathsf{chl}})\deg(\varphi'_{\mathsf{chl}})] \circ \sigma' \circ \hat{\sigma}. \tag{2}$$

Write $\varphi'_{\mathsf{chl}} \circ \hat{\varphi}_{\mathsf{chl}} = [2^a] \circ \psi$ and $\sigma' \circ \hat{\sigma} = [d] \circ \nu$ where ψ and ν have cyclic kernel. We deduce from Eq. (2) that $2^a m = d \deg(\varphi_{\mathsf{chl}}) \deg(\varphi'_{\mathsf{chl}})$ is the largest integer dividing either side of the equality, and $\psi = \nu$ is the cyclic part of either side.

On one hand, we have $\deg(\nu) \leq \deg(\sigma) \deg(\sigma') \leq 2^{2e_{\mathsf{rsp}}}$. On the other hand, Lemma 16 implies

$$\deg(\psi) = \frac{\varphi'_{\mathsf{chl}} \circ \hat{\varphi}_{\mathsf{chl}}}{2^{2a}} > 2^{2(e - e_{\mathsf{chl}})} \geq 2^{2e_{\mathsf{rsp}}}.$$

This contradicts the equality $\psi = \nu$. □

5.2 Zero-knowledge property

In this section, we prove that the identification protocol is honest-verifier zero-knowledge. Let us first prove that the commitment curve is indistinguishable from a uniformly random curve.

Lemma 18. *If $N_{\mathsf{com}} \geq 2^{4\lambda}$, then an honestly generated commitment curve E_{com} is at statistical distance $\tilde{O}(2^{-\lambda})$ from a uniformly random supersingular elliptic curve.*

Proof. It follows from [11, Proposition 29] with $\varepsilon = 1$ and $p = \Theta(2^{2\lambda})$. □

To prove that the protocol has the zero-knowledge property, we prove that there exists a simulator producing transcripts indistinguishable from an honest run of the protocol. Like in SQIsignHD [11], the simulator runs in polynomial time if it has access to an oracle producing random isogenies. This "random isogeny" oracle comes in two variants: the UTO and the FIDIO.

Definition 19. *A* uniform target oracle *(UTO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and an integer $N \geq 2\sqrt{2p}/\pi$, and outputs a random isogeny $\varphi : E \to E'$ (in efficient representation) such that:*

1. *The distribution of E' is uniform among all the supersingular elliptic curves.*
2. *The conditional distribution of φ given E' is uniform among isogenies $E \to E'$ of degree smaller or equal to N.*

Remark 20. The condition $N \geq 2\sqrt{2p}/\pi$ ensures such an oracle exists: for any pair (E_1, E_2), the collection of isogenies $E_1 \to E_2$ of degree smaller than N is non-empty (Minkowski's bound for the lattice $\mathrm{Hom}(E_1, E_2)$).

Definition 21. *A* fixed degree isogeny oracle *(FIDIO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and an integer N, and outputs a uniformly random isogeny $\varphi : E \to E'$ (in efficient representation) with domain E and degree N.*

Theorem 22. *If $2^{e_{\mathsf{rsp}}} \geq 2\sqrt{2p}/\pi$ and $N_{\mathsf{com}} \geq 2^{4\lambda}$, then the identification protocol is statistically honest-verifier zero-knowledge in the UTO and FIDIO model. In other words, there exists a polynomial time simulator $\mathcal{S}$ with access to a UTO and a FIDIO that produces random transcripts which are statistically indistinguishable from honest transcripts.*

Proof. The simulator proceeds as follows:

1. Generate an isogeny $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ according to the honest challenge distribution.
2. Call the UTO on input $(E_{\mathsf{chl}}, 2^{e_{\mathsf{rsp}}})$, resulting in the isogeny $\hat{\varphi}_{\mathsf{rsp}} : E_{\mathsf{chl}} \to E_{\mathsf{com}}$.
3. Decompose $\varphi_{\mathsf{rsp}} = \psi \circ \varphi_{\mathsf{rsp}}^{(1)}$ with $q' = \deg(\varphi_{\mathsf{rsp}}^{(1)})$ odd and $\deg(\psi) = 2^n$ a power of two. Let $2^{n_{\mathsf{bt}}} = \#(\ker(\hat{\psi}) \cap \ker(\hat{\varphi}_{\mathsf{chl}}))$. Let $r' = n - n_{\mathsf{bt}}$.
4. Call the FIDIO on input $(E_{\mathsf{com}}, 2^{e_{\mathsf{rsp}}-r'} - q')$, resulting in the isogeny $\varphi_{\mathsf{aux}} : E_{\mathsf{com}} \to E_{\mathsf{aux}}$.

From the properties of the UTO and FIDIO, the above procedure is equivalent to the following one:

1. Generate a uniformly random supersingular curve E_{com}
2. Generate an isogeny $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$ according to the honest challenge distribution.
3. Generate a uniformly random isogeny φ_{rsp} from E_{com} to E_{chl}, of degree at most $2^{e_{\mathsf{rsp}}}$.
4. Decompose $\varphi_{\mathsf{rsp}} = \psi \circ \varphi_{\mathsf{rsp}}^{(1)}$ with $q' = \deg(\varphi_{\mathsf{rsp}}^{(1)})$ odd and $\deg(\psi) = 2^n$ a power of two. Let $2^{n_{\mathsf{bt}}} = \#(\ker(\hat{\psi}) \cap \ker(\hat{\varphi}_{\mathsf{chl}}))$. Let $r' = n - n_{\mathsf{bt}}$.
5. Generate a uniformly random isogeny β from E_{com} and of degree $2^{e_{\mathsf{rsp}}-r'} - q'$.

This is precisely the order in which an honest run of the protocol proceeds. The distribution for the first step matches the honest run by Lemma 18. The distributions of following steps match the honest ones by construction. □

On the UTO and FIDIO oracles. Let us first argue that the UTO is essentially redundant: given a FIDIO, one can implement an oracle that is computationally indistinguishable from a UTO, at least when the bound N is sufficiently large. We proceed in two steps:

1. First, we use the FIDIO to build an oracle which outputs a uniform isogeny σ from E with $\deg(\sigma) \leq N$. In other words, one can turn a FIDIO into a RADIO, following the terminology of [11].
2. Second, we argue that this distribution (the output of a RADIO) is indistinguishable from the output of a UTO.

Recall the definition of a RADIO.

Definition 23 ([11, Definition 41]). *A* random any-degree isogeny oracle *(RADIO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and an integer N, and outputs a uniformly random isogeny $\varphi : E \to E'$ (in efficient representation) with domain E and degree at most N.*

Let us first explain how one can turn a FIDIO into a RADIO. Let f_N be the probability distribution of the degree of the output of a RADIO: for any integer q, let $f_N(q)$ be the probability that the degree of the output of a RADIO on input (E, N) is equal to q. Note that conditional on the degree of the output begin q, the FIDIO and the RADIO follow the same distribution: uniform among isogenies with domain E and degree q. Therefore, to simulate a RADIO, we can proceed as follows: on input (E, N),

1. sample an integer q following the distribution f_N;
2. call the FIDIO on input (E, q), and return the output.

To sample from the distribution f_N, observe that the value $f_N(q) = \tilde{\Theta}(q/N^2)$ can be computed efficiently if the factorisation of q is known. Therefore, we can do rejection sampling by sampling uniformly random integers in $[1, N]$ *together with their factorisation* (see [1]).

We proceed as follows: sample a random degree $q \leq N$, then call the FIDIO to sample a uniform isogeny of degree q from E. The only difficulty is to sample $q \leq N$ with the same distribution as the degree of a UTO-output (it is not the uniform distribution). Given the prime factorisation $q = \prod_i \ell_i^{e_i}$, there are $\prod_i \ell_i^{e_i}$.

Now that we can turn a FIDIO into a RADIO, it remains to argue that a RADIO is indistinguishable from a UTO. For N large enough, it is indeed statistically indistinguishable: conditionally on the target curve, the two distributions are identical, and it is proven in [11, Theorem 42] that when $N = \Theta(p^{1+\varepsilon})$ for $\varepsilon \in (0, 2]$, the distribution on the target curves are at statistical distance $O(p^{-\varepsilon/2})$. Therefore, when $N = \Theta(p^{1+\varepsilon})$, the RADIO and the UTO are at statistical distance $O(p^{-\varepsilon/2})$. The bound $N = O(p^{1/2})$ used in the protocol is not large enough for this theorem to apply, but we expect the distributions to remain computationally indistinguishable.

The conclusion of the above discussion is that in Theorem 22, the UTO is heuristically redundant. In other words, there is a (heuristic) simulator in the FIDIO model. It remains to argue that this FIDIO does not hurt the security assumption: access to a FIDIO does not help with solving the endomorphism ring problem. We refer to the analogous discussion about the security of SQIsignHD in [11, Section 5.3]. In essence, all a FIDIO does is compute a random walk from a source curve. We already know how to compute random walks of smooth degree (by taking a sequence of random isogeny steps of small prime degree), and a FIDIO extends this capability to random walks with potentially large prime steps.

5.3 Security of the signature protocol

In the previous sections, we have shown that the SQIsign2D Σ-protocol is 2-special sound, under the assumed hardness of Problem 13, and zero-knowledge in the UTO and FIDIO model. Hence, a direct application of the Fiat–Shamir transform [21] yields a digital signature that is EUF-CMA secure in the random oracle model (ROM) [36], under the hardness of Problem 13 when the attacker has access to the UTO and FIDIO.

However, the signature protocol whose security is proved in [36] includes commitments in the signature. As explained in Section 4.2, we replace the commitment in the signature with the challenge (by relying on the commitment-recoverability property of the Σ-protocol) to reduce the signature size. To show the security equivalence of the two approaches, we rely on [2, Theorem 2], which requires the commitment-recovering algorithm to be correct and sound.

Given a transcript $(\mathsf{com}, \mathsf{chl}, \mathsf{rsp})$, correctness requires the commitment-recovering algorithm to always produce com given chl and rsp, and it follows from Remark 12. Soundness, in this context, means that it is computationally hard to find a pair of challenge and response $(\mathsf{chl}, \mathsf{rsp})$ for which the commitment-recovering algorithm produces a commitment com such that $(\mathsf{com}, \mathsf{chl}, \mathsf{rsp})$ is *not* a valid transcript. In our case, the commitment-recovering algorithm is perfectly sound (i.e. soundness holds even against unbounded adversaries): the curve produced by the commitment-recovering algorithm introduced in Section 4.2 is always the codomain of an isogeny, efficiently represented in the response, starting from E_{chl}, and the curve E_{com} does not need to satisfy any additional requirement to be a valid commitment; thus, the resulting transcript is always valid.

This shows that the SQIsign2D signature protocol is EUF-CMA secure in the ROM, assuming the hardness of Problem 13 when the attacker has also access to the UTO and FIDIO.

6 Instantiation and experimental results

We selected parameters for the scheme described in Section 4 matching NIST post-quantum security levels I, III and V, and implemented them in C building upon the SQIsign reference implementation. We now give details on our implementation and compare its performance to the other variants of SQIsign.

6.1 Parameter choices and and signature size

Choice of the primes. As mentioned in Section 5, the best attacks against the Supersingular Endomorphism Ring problem have classical complexity $\tilde{O}(p^{1/2})$ and quantum complexity $\tilde{O}(p^{1/4})$, where p is the characteristic of the base field. These are also the best known attacks against SQIsign (see [9, Chapter 9]) and SQIsign2D. Our security reduction, although not tight and formulated in the UTO/FIDIO model, further justifies using these complexities to set parameters.

To reach NIST's security levels I, III and V, we thus look for primes of roughly 256, 384 and 512 bits respectively. For maximum efficiency, we selected primes such that $2p$ fits in 4, 6 and 8 64-bits words. The final requirement is that $p + 1 = c \cdot 2^e$ with c as small as possible; it is also desirable that c has small Hamming weight. Our final choices are listed in Table 2.

Table 2. Chosen parameters for SQIsign2D. Sizes in bytes.

	NIST I	NIST III	NIST V
Prime	$5 \cdot 2^{248} - 1$	$65 \cdot 2^{376} - 1$	$27 \cdot 2^{500} - 1$
Public-key size	66	98	130
Signature size	148	222	294

Signature encoding and sizes. The resulting public key and signature sizes are reported in Table 2. We detail below how these numbers are computed.

As for other SQIsign variants, there are various possibilities to decrease the signature size at the expense of slower verification and signing. For our implementation, we prioritised verification speed over signature size, and thus chose to not use the most advanced compression tricks. As we mentioned already (see Section 4.2), our scheme is commitment recoverable which means that we do not need to include the commitment curve in the signature. This requires a little more work for the signer, but it makes close to no difference for the verification.

Outside of this, the only other real compression we use is to represent the basis $P_{\mathsf{chl}}, Q_{\mathsf{chl}}$ as four elements in $[0, 2^{e_{\mathsf{rsp}}}]$ (that are the coefficients of $P_{\mathsf{chl}}, Q_{\mathsf{chl}}$ in a canonical basis of E_{chl}). For a given level security of λ, we have $\log p \approx 2\lambda$ and $e_{\mathsf{rsp}} \approx \lambda$, so this compression allows us to decrease the size of the basis representation from 8λ (since each point is represented as one element in $\mathbb{F}_{p^2}$) to 4λ. This requires the additional computation a canonical basis of E_{chl}. In general, this is not cheap to compute, but we can abuse tricks specialised for the generation of bases of $E[2^k]$ such as the entangled torsion basis from [43, Algorithm 3.1] or the modification described in [41, Section 5.1].

We can further reduce the cost of the basis generation for the verifier by including hints at the very reasonable cost of increasing the signature size by two bytes. The idea of hints to speed-up basis generation was first introduced as part of the compression procedure in the original SQIsign paper. Using the specialised algorithms [43,41] boils down to selecting x-coordinates with chosen Legendre symbols and checking whether the chosen x is a valid x-coordinate for a point on the curve.

In this context, the hints can be either indices of tables of "good" x-coordinates, or some integer h such that $x = i + h \in \mathbb{F}_{p^2}$ are values with the correct Legendre symbol properties and points on the curve.[9] Moreover, it does not cost anything to the signer to include these hints. In our experiments, the value of the hints never went over 50, thus we conjecture that for the sizes considered for our scheme, the hints for a basis can fit in two bytes with overwhelming probability.

In our scheme, we use hints for the deterministic basis generation required by the verification: one for E_{pk} and one for E_{chl}. Thus, this increases the size of the public key by two bytes and the size of the signature by two bytes.

In the end, the size of the public key is $4\lambda + 16$ bits, and the size of the signature is $9\lambda + 16 + 2\log_2(2\lambda)$ bits (λ for the scalar chl ,4λ for E_{aux}, $4\lambda + 16$ for $P_{\mathsf{chl}}, Q_{\mathsf{chl}}$ and $2\log(2\lambda)$ for r' and n_{bt}).

Remark 24. The representation of $P_{\mathsf{chl}}, Q_{\mathsf{chl}}$ could be further reduced to 3λ, but would require the verifier to compute a pairing to recover the last coefficient. Since pairing are quite costly, we decided not too include this optimisation, but

[9] In our implementation, we begin sampling coordinates from two tables with twenty values. This gives a 2^{-20} chance of failure, which we recover from by then sampling coordinates of the form $x = i + h$ as above. Regardless of whether the basis is generated from a look-up or sampling, the cost for verification is the same thanks to the supplied hint.

it could be part of the signature in cases where size of the signature is critical. Experimentally, for NIST level 1, this would gain 16B on the signature size, at the cost of an increase on the verification time by 5 to 10 percent.

6.2 Implementation choices and optimisations

We implemented SQIsign2D in C by modifying SQIsign's reference code.[10]

Multi-precision integers and quaternion algebras are built on top of the GMP library.[11] The only significant difference with SQIsign is the use of floating point numbers in the LLL algorithm instead of exact rationals.

Arithmetic modulo p has two implementations: one based on the Fiat-Crypto code generator [20] and one optimised implementation using the special form of the primes used, allowing for efficient Montgomery reduction. We give a detail of the design choices of this implementation and future work in [3, Appendix C].

Elliptic curves, pairings, and isogenies. Following standard practice, we represent elliptic curves in Montgomery form and use the formulas in [10,37] to evaluate 2-isogenies and 4-isogenies. Compared to SQIsign, we do not use formulas for isogenies of odd degrees, and in particular we do not need the costly $\sqrt{\text{élu}}$ algorithm [5].

For pairings, we use the biextension/cubical formulas from [40], because these are currently, to the best of our knowledge, the fastest available to compute pairings on Montgomery curves. We note that since we only need to compute pairings between points of order 2^e, we only need to use biextension doublings.

Two-dimensional abelian varieties are represented in theta coordinates and their two-dimensional 2-isogenies are evaluated using the formulas in [12]. We use the projective version of their formulas to remove almost all inversions along the isogeny chain.

All other algorithms are either taken from the implementation of SQIsignHD or have been written from scratch according to the description in Section 3, with minor deviations to allow for several small optimizations, such as commitment recoverability, bases compression, and hints.

6.3 Performance

We ran benchmarks to compare our implementations to the state of the art. All code was compiled on Ubuntu using clang 14, with flags `-march=native -O3`, dynamically linking to the system GMP library (version 6.2.1). Benchmarks were run on an Intel Xeon Gold 6338 (Ice Lake) CPU clocked at 2 GHz with turbo boost disabled. In Table 3 we compare our pure-C implementation to:

- The reference implementation of SQIsign at https://github.com/SQISign/the-sqisign. Because this uses the same modular arithmetic based on Fiat-Crypto, it is a fair comparison for showcasing the higher-level algorithmic improvements of SQIsign2D.

[10] Our code will be available at https://github.com/SQISign/sqisign2d-west-ac24.

[11] https://gmplib.org/.

Table 3. Performance of SQIsign2D on Intel Xeon Gold 6338 (Ice Lake, 2GHz), using generic finite field arithmetic (Fiat-Crypto), GMP 6.2.1. Turbo-boost disabled. Timings in 10^6 cycles.

	Level	SQIsign	SQIsignHD	SQIsign2D
Keygen	I	2,800	190	120
	III	21,300	—	440
	V	91,600	—	1,070
Sign	I	4,600	115	290
	III	39,300	—	1,040
	V	165,000	—	2,490
Verify	I	93	—	25
	III	641	—	98
	V	2,080	—	247

- The implementation of SQIsignHD at https://github.com/Pierrick-Dartois/SQISignHD-lib. This codebase is momentarily lacking a C implementation of the verification, thus we only benchmark key generation and signatures.

For the optimised pure-C implementation we additionally compare to the implementation of SQIsign [15] at https://github.com/SQISign/sqisign-ec23. This has much better assembly optimisations for finite fields and is generally faster than the reference implementation. However, our implementation is the only one to implement all three NIST levels. We additionally implemented the heuristic variant of SQIsign2D described in [3, Appendix B] and included these results under the label SQIsign2D-H. The results are reported in Table 4.

Table 4. Performance of SQIsign2D on Intel Xeon Gold 6338 (Ice Lake, 2GHz), with finite field arithmetic optimised using intrinsics for the Ice Lake architecture, GMP 6.2.1. Turbo-boost disabled. Timings in 10^6 cycles.

	Level	SQIsign ([9])	SQIsign ([15])	SQIsign2D	SQIsign2D-H
Keygen	I	1,700	400	60	58
	III	—	—	170	170
	V	—	—	360	350
Sign	I	2,400	1880	160	100
	III	—	—	460	280
	V	—	—	940	570
Verify	I	39	29	9	9
	III	—	—	29	29
	V	—	—	62	60

References

1. Bach, E.: How to generate factored random numbers. SIAM Journal on Computing **17**(2), 179–193 (1988). https://doi.org/10.1137/0217012
2. Backendal, M., Bellare, M., Sorrell, J., Sun, J.: The Fiat-Shamir zoo: Relating the security of different signature variants. In: Gruschka, N. (ed.) Secure IT Systems - 23rd Nordic Conference, NordSec 2018, Oslo, Norway, November 28-30, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11252, pp. 154–170. Springer (2018). https://doi.org/10.1007/978-3-030-03638-6_10
3. 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. Cryptology ePrint Archive, Report 2024/760 (2024), https://eprint.iacr.org/2024/760
4. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 98–126. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8739-9_4
5. Bernstein, D.J., De Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. Open Book Series **4**(1), 39–55 (2020). https://doi.org/10.2140/obs.2020.4.39
6. Biasse, J.F., Jao, D., Sankar, A.: A quantum algorithm for computing isogenies between supersingular elliptic curves. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014. LNCS, vol. 8885, pp. 428–442. Springer, Cham (Dec 2014). https://doi.org/10.1007/978-3-319-13039-2_25
7. Castryck, W., Chen, M., Invernizzi, R., Lorenzon, G., Vercauteren, F.: Breaking and repairing SQIsign2D-East. Cryptology ePrint Archive, Paper 2024/1453 (2024), https://eprint.iacr.org/2024/1453
8. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_15
9. Chavez-Saab, J., Santos, M.C., De Feo, L., Eriksen, J.K., Hess, B., Kohel, D., Leroux, A., Longa, P., Meyer, M., Panny, L., Patranabis, S., Petit, C., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: SQIsign. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
10. Costello, C., Hisil, H.: A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 303–329. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_11
11. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: New dimensions in cryptography. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 3–32. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_1
12. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to $(2,2)$-isogenies in the theta model and applications to isogeny-based cryptography. Cryptology ePrint Archive, Report 2023/1747 (2023), https://eprint.iacr.org/2023/1747
13. De Feo, L.: Mathematics of isogeny based cryptography (2017), https://arxiv.org/abs/1711.04062
14. 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.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_3

15. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 659–690. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_23
16. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. DCC **78**(2), 425–440 (2016). https://doi.org/10.1007/s10623-014-0010-1
17. Deuring, M.: Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **14**, 197–272 (1941), https://doi.org/10.1007/BF02940746
18. Dirichlet, P.G.L.: Beweis des Satzes, dass jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält. Abhandlungen der Königlichen Preußischen Akademie der Wissenschaften zu Berlin **48**, 45–71 (1837). https://doi.org/10.1017/CBO9781139237321.012
19. Duparc, M., Fouotsa, T.B., Vaudenay, S.: SILBE: an updatable public key encryption scheme from lollipop attacks. Cryptology ePrint Archive, Report 2024/400 (2024), https://eprint.iacr.org/2024/400
20. Erbsen, A., Philipoom, J., Gross, J., Sloan, R., Chlipala, A.: Simple high-level code for cryptographic arithmetic - with proofs, without compromises. In: 2019 IEEE Symposium on Security and Privacy. pp. 1202–1219. IEEE Computer Society Press (May 2019). https://doi.org/10.1109/SP.2019.00005
21. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Berlin, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
22. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
23. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers. Oxford, sixth edn. (1975). https://doi.org/10.1093/oso/9780199219858.001.0001
24. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Renes, J., Soukharev, V., Urbanik, D., Pereira, G., Karabina, K., Hutchinson, A.: SIKE. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
25. 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 (Nov / Dec 2011). https://doi.org/10.1007/978-3-642-25405-5_2
26. Kani, E.: The number of curves of genus two with elliptic differentials. Journal für die reine und angewandte Mathematik **485**, 93–122 (1997). https://doi.org/10.1515/crll.1997.485.93
27. Kirschmer, M., Voight, J.: Algorithmic enumeration of ideal classes for quaternion orders. SIAM Journal on Computing **39**(5), 1714–1747 (2010). https://doi.org/10.1137/080734467
28. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. LMS Journal of Computation and Mathematics **17**(A), 418–432 (2014). https://doi.org/10.1112/S1461157014000151

29. Leroux, A.: Quaternion algebras and isogeny-based cryptography. Ph.D. thesis, École Polytechnique, France (2022), http://www.lix.polytechnique.fr/Labo/Antonin.LEROUX/manuscrit_these.pdf
30. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 448–471. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_16
31. Nakagawa, K., Onuki, H.: QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 75–106. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68388-6_4
32. Nakagawa, K., Onuki, H.: SQIsign2D-east: A new signature scheme using 2-dimensional isogenies. Cryptology ePrint Archive, Report 2024/771 (2024), https://eprint.iacr.org/2024/771
33. Onuki, H., Nakagawa, K.: Ideal-to-isogeny algorithm using 2-dimensional isogenies and its application to SQIsign. Cryptology ePrint Archive, Report 2024/778 (2024), https://eprint.iacr.org/2024/778
34. Page, A., Robert, D.: Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766 (2023), https://eprint.iacr.org/2023/1766
35. Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VI. LNCS, vol. 14656, pp. 388–417. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58751-1_14
36. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 387–398. Springer, Berlin, Heidelberg (May 1996). https://doi.org/10.1007/3-540-68339-9_33
37. Renes, J.: Computing isogenies between Montgomery curves using the action of (0, 0). In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018. pp. 229–247. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_11
38. Robert, D.: Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive, Report 2022/1068 (2022), https://eprint.iacr.org/2022/1068
39. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_17
40. Robert, D.: Fast pairings via biextensions and cubical arithmetic. Cryptology ePrint Archive, Report 2024/517 (2024), https://eprint.iacr.org/2024/517
41. Santos, M.C.R., Eriksen, J.K., Meyer, M., Reijnders, K.: AprèsSQI: Extra fast verification for SQIsign using extension-field signing. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 63–93. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_3
42. Silverman, J.H.: The arithmetic of elliptic curves, Graduate texts in mathematics, vol. 106. Springer (1986). https://doi.org/10.1007/978-0-387-09494-6
43. Zanon, G.H.M., Simplicio, M.A., Pereira, G.C.C.F., Doliskani, J., Barreto, P.S.L.M.: Faster key compression for isogeny-based cryptosystems. IEEE Transactions on Computers **68**(5), 688–701 (2019). https://doi.org/10.1109/TC.2018.2878829