PRISM: Simple And Compact Identification and Signatures From Large Prime Degree Isogenies

Andrea Basso[1], Giacomo Borin[1], Wouter Castryck[2], Maria Corte-Real Santos[3], Riccardo Invernizzi[2], Antonin Leroux[4,5], Luciano Maino[6], Frederik Vercauteren[2], and Benjamin Wesolowski[3]

[1] IBM Research Europe, Zürich, Switzerland
[2] COSIC, KU Leuven, Belgium
[3] ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France
[4] DGA-MI, Bruz, France
[5] IRMAR - UMR 6625, Université de Rennes, France
[6] University of Bristol, Bristol, United Kingdom

Abstract. The problem of computing an isogeny of large prime degree from a supersingular elliptic curve of unknown endomorphism ring is assumed to be hard both for classical as well as quantum computers. In this work, we first build a two-round identification protocol whose security reduces to this problem. The challenge consists of a random large prime q and the prover simply replies with an efficient representation of an isogeny of degree q from its public key. Using the hash-and-sign paradigm, we then derive a signature scheme with a very simple and flexible signing procedure and prove its security in the standard model. Our optimized C implementation of the signature scheme shows that signing is roughly 1.8× faster than all SQIsign variants, whereas verification is 1.4× times slower. The sizes of the public key and signature are comparable to existing schemes.

Keywords: Isogenies · Post-Quantum · Signatures · ID Protocols

1 Introduction

Post-quantum cryptography aims to construct cryptographic protocols that are secure against adversaries with access to both classical and quantum computers. This field has recently gained interest from the community due to the increased investment into quantum computing. Most notably, in 2016, NIST began an effort to standardise post-quantum secure key encapsulation mechanisms (KEMs) and digital signature schemes. This culminated in the standardisation of the lattice-based KEM Kyber [14] and signature schemes Dilithium [27], Falcon [32], and the hash-based signature SPHINCS+ [9]. NIST, however, is still seeking alternative signature schemes for standardisation [42] due to the heavy reliance

PRISM stands for PRime degree ISogeny Mechanism

on lattice-based assumptions, and the relatively large signature sizes compared to pre-quantum alternatives.

Isogeny-based cryptography offers a promising answer to these problems. For instance, the isogeny-based signature scheme SQIsign boasts the smallest combined public key and signature size of any post-quantum alternative. It is currently submitted to Round 1 of NIST's alternate call for post-quantum secure signature schemes [16]. The main disadvantage of SQIsign and isogeny-based schemes in general is their inefficiency: signing and verification are orders of magnitude slower than lattice-based alternatives. As such, a large portion of research in isogenies has focused on optimizing the subroutines of SQIsign, see for example [24,17]. Recently, impressive strides have been made in this regard by exploiting the attacks on SIDH/SIKE [15,39,45]. Indeed, these attacks uncovered a powerful tool: using higher-dimensional representations, we can efficiently represent a one-dimensional isogeny of non-smooth degree by embedding it in a higher-dimensional isogeny of smooth degree between products of elliptic curves. The efficiency of the protocol depends on the embedding dimension of the isogenies used in this representation.

The first variant of SQIsign that used the SIDH attacks constructively was SQIsignHD, which relied on four-dimensional representations to obtain an impressive speed-up in the signing time and even smaller signatures than SQIsign. Verification, however, suffered from the heavy cost of computing four-dimensional isogenies. A wave of new protocols — SQIsign2D-West [6], SQIsign2D-East [41] and SQIPrime [28] — presented new variants of SQIsign making use of two-dimensional representations. These two-dimensional variants of SQIsign achieve signature sizes comparable to SQIsignHD, faster signing than SQIsign (though slower than SQIsignHD), and verification faster than both.

Another drawback that affects all variants of SQIsign is their design complexity. The intricate signature process complicates the description of the scheme, and this reflects both in the security analysis and in the flexibility of the design. The former issue leads to security proofs requiring specific ad-hoc oracles and assumptions. The latter makes it hard to use these schemes as a building block for more elaborated protocols. Despite its extremely compact signatures, after several years since its initial publication, only a few advanced functionalities based on SQIsign have been proposed, see e.g. [13,44].

High-dimensional isogenies were used in a different manner by Leroux in [38] to introduce an efficient Verifiable Unpredictable Function (VUF), from which a Verifiable Random Function and the first isogeny-based hash-and-sign signature scheme can be derived. The function underlying Leroux's construction computes the codomain of an isogeny of given kernel, where the kernel has a fixed large prime order. The security thus relies on the hardness of computing such a function without the knowledge of the endomorphism ring of the domain curve.

Contributions. We present a new isogeny-based identification protocol, which we transform into a signature scheme PRISM-sig via the hash-and-sign paradigm. The security of both schemes relies on the fact that computing a large prime degree

isogeny from an elliptic curve E is conjectured to be hard *without knowledge of the endomorphism ring of* E. The identification protocol is very simple:

1. The prover samples a secret key $\phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$ with corresponding public key E_{vk}.
2. A verifier challenges the prover with a large prime q.
3. The prover replies with a two-dimensional representation of a degree-q isogeny from E_{vk}.

The simplicity of the above protocol and the derived signature scheme, particularly when compared to SQIsign and its variants, improves the flexibility of the scheme and makes it easier to assess its security. In particular, we obtain an isogeny-based signature scheme that is secure in the standard model. In view of the simplicity, we also expect our construction to be a useful building block in other, more advanced schemes.

Our contributions are summarized as follows:

- In Section 3, we construct a new efficient identification protocol, which we call PRISM-id. Unlike other isogeny-based ID protocols, PRISM-id is *not* a Σ-protocol: this leads to smaller communication costs and more efficient computations. After constructing an appropriate hash function, we obtain a signature scheme (PRISM-sig) based on our identification protocol via the hash-and-sign paradigm. The resulting signature scheme is simple, flexible, compact, and efficient: at NIST security Level I, public keys and signatures are only 66 and 189 bytes, while signing and verifying take less than 70 and 8 ms, respectively.
- In Section 4, we prove the security of both the identification scheme and the signature scheme in the standard model, showing that their hardness is linked to well-understood problems in isogeny-based cryptography. For the signature scheme, we also prove security under a weaker assumption in the random oracle model.
- In Section 5, we propose concrete parameters for our schemes for NIST Level I, III, and V security, and we provide an optimized implementation in C of PRISM-sig at Level I. We compare its efficiency and key/signature sizes with other variants of SQIsign. We also compare PRISM-id and PRISM-sig, and observe that PRISM-id has a faster verification and much lower communication costs. The reason is that to instantiate a secure signature scheme via the hash-and-sign paradigm we need to avoid hash collisions, forcing us to select larger parameters for PRISM-sig compared to PRISM-id.

Related work. Our construction shares strong conceptual similarities with Leroux's verifiable unique function (VUF) [38] as they both rely on the hardness of computing large degree isogenies without the knowledge of the endomorphism ring. The main difference lies in the degree of the response isogenies: in [38], the isogenies have fixed degree, whereas in our case *the degree is the challenge*, and it thus changes across executions. As a result, our choice of parameters

is less constrained, and our security is arguably better. Indeed, the response isogeny has kernel defined over a field extension of exponential degree with overwhelming probability (whereas in [38] the kernel is defined over a small field extension), which provides us with additional security guarantees even if there were breakthroughs in the computation of large prime degree isogenies.

Compared to SQIsign and its variants, our scheme is significantly simpler. Not only does this simplicity make it easier to analyze and implement the protocol, but it also leads to faster signing times. Our signing procedure requires computing fewer two-dimensional isogenies compared to all the SQIsign variants, and no one-dimensional isogenies at all, leading to a 1.8× speedup when compared to SQIsign2D-West. On the other hand, verification is slightly slower: it takes roughly 1.4× longer than the verification in SQIsign2D-West. The signature size is very compact, which is comparable with most other isogeny-based protocols. For instance, it is the same as SQIsign2D-East.

Outline. We begin by introducing the necessary background in Section 2. In Section 3 we introduce the new identification protocol and digital signature scheme based on large prime degree isogenies, and prove their security in Section 4. Finally, in Section 5 we evaluate the performance of our schemes, both in terms of efficiency and communication cost.

Acknowledgements. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT - No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VR20192203. This work was supported in part by SNSF Consolidator Grant CryptonIs 213766.

2 Preliminaries

In this section we recall some background knowledge about the Deuring correspondence, computation of isogenies in dimension two and different variants of the SQIsign protocol. We assume some familiarity with elliptic curves and their isogenies, and refer the reader to [22,47] for more information. From this point onwards, p is a prime with $p \equiv 3 \pmod 4$.

2.1 Quaternion algebras and the Deuring correspondence

We start by introducing quaternion algebras. A *quaternion algebra* over $\mathbb{Q}$ is a division algebra defined by $\mathbb{Q}+\mathbb{Q}i+\mathbb{Q}j+\mathbb{Q}k$, where $i^2 = a$, $j^2 = b$, $ij = -ji = k$ for some $a, b \in \mathbb{Q}^*$. We denote it by $H(a, b)$. We say that $H(a, b)$ is *ramified* at a place v of $\mathbb{Q}$ if the extension of scalars $H(a, b) \otimes_{\mathbb{Q}} \mathbb{Q}_v$ is not isomorphic to the algebra of 2×2 matrices over $\mathbb{Q}_v$. Up to isomorphism, for a given prime p there exists a unique quaternion algebra ramified exactly at p and ∞, which we denote

by $\mathcal{B}_{p,\infty}$. Since we assume $p \equiv 3 \bmod 4$, we can choose a basis such that $i^2 = -1$ and $j^2 = -p$.

For a given element $\alpha = a + bi + cj + dk \in \mathcal{B}_{p,\infty}$ we define its conjugate $\bar{\alpha} := a - bi - cj - dk$ and its *reduced norm* $\mathrm{nrd}(\alpha) := \alpha\bar{\alpha}$. A *fractional ideal* in $\mathcal{B}_{p,\infty}$ is a $\mathbb{Z}$-submodule of rank 4. An *order* $\mathcal{O}$ is a fractional ideal that is also a subring. An order is *maximal* if it is not properly contained in any other order. Let I be a fractional ideal in $\mathcal{B}_{p,\infty}$. We define the left order of I to be $\mathcal{O}_L(I) := \{\alpha \in \mathcal{B}_{p,\infty} \mid \alpha I \subset I\}$. We can similarly define the right order $\mathcal{O}_R(I)$ of a fractional ideal I, and I is called a connecting ideal for $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$. If I is contained in its left order (or, equivalently, in its right order) then it is an *integral ideal*, or just an *ideal* for short.

For a fractional ideal I, we denote its conjugate by $\bar{I} = \{\bar{\alpha} \mid \alpha \in I\}$. The reduced norm of an ideal I, denoted by $\mathrm{nrd}(I)$, is defined as the gcd of the reduced norms of the elements of I. For a maximal order $\mathcal{O}$, any left $\mathcal{O}$-ideal I can be written as $I = \mathcal{O}\alpha + \mathcal{O}\mathrm{nrd}(I)$ for some $\alpha \in I$. Two ideals I and J are *equivalent* if there exists $\beta \in \mathcal{B}_{p,\infty}^*$ such that $I = J\beta$. We denote equivalence by $I \sim J$. A more detailed discussion of quaternion algebras can be found in [51].

The Deuring Correspondence. Deuring [26] showed a categorical equivalence between maximal orders in $\mathcal{B}_{p,\infty}$ and supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. This equivalence is known as the *Deuring correspondence*. Under this correspondence, to each maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$ we can associate a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ such that $\mathrm{End}(E) \cong \mathcal{O}$. An isogeny $\varphi : E_1 \to E_2$ corresponds to an 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, $\deg(\varphi) = \mathrm{nrd}(I_\varphi)$.

Example 1. Since $p \equiv 3 \bmod 4$, the elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular. We can define endomorphisms $\iota : (x, y) \mapsto (-x, \sqrt{-1}y)$ and $\pi : (x, y) \mapsto (x^p, y^p)$ of E_0, where $\sqrt{-1}$ is a fixed square root of -1 in $\mathbb{F}_{p^2}$. We have the following isomorphism of rings:

$$\begin{aligned} \mathcal{O}_0 := \mathbb{Z}\left\langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \right\rangle &\longrightarrow \mathrm{End}(E), \\ a + bi + cj + dk &\longmapsto a + b\iota + c\pi + d\iota\pi. \end{aligned}$$

Throughout the paper, we will always denote by E_0 the curve $y^2 = x^3 + x$.

Pushforward Isogenies and Ideals. Consider an isogeny $\varphi_1 : E \to E_1$ and a separable isogeny $\varphi_2 : E \to E_2$ of degree coprime to $\deg(\varphi_1)$. We denote by $[\varphi_1]_*\varphi_2 : E_1 \to E'$ the pushforward isogeny of φ_2 under φ_1, i.e., the separable isogeny such that $\ker([\varphi_1]_*\varphi_2) = \varphi_1(\ker(\varphi_2))$; see Figure 1.

Under the Deuring correspondence, we can define the pushforward of I_{φ_2} under I_{φ_1} as the left ideal of $\mathcal{O}_R(I_{\varphi_1})$ corresponding to the isogeny $[\varphi_1]_*\varphi_2$, and we denote it by $[I_{\varphi_1}]_*I_{\varphi_2}$. By [23, Lemma 3] it can be computed as

$$[I_{\varphi_1}]_*I_{\varphi_2} = \frac{1}{\mathrm{nrd}(I_{\varphi_1})}\overline{I_{\varphi_1}}(I_{\varphi_1} \cap I_{\varphi_2}). \tag{1}$$

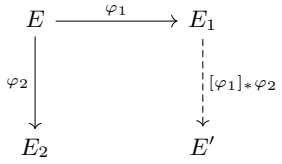


Fig. 1. Pushforward isogeny of φ_2 under φ_1.

We summarize the Deuring correspondence in Table 1.

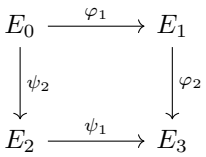
Supersingular elliptic curves	**Quaternions**
Supersingular j-invariants $j(E) \in \mathbb{F}_{p^2}$ (up to Galois conjugacy)	Maximal orders $\mathcal{O} \cong \mathrm{End}(E)$ in $B_{p,\infty}$ (up to isomorphism)
(E_1, φ) with $\varphi : E \to E_1$	I_φ integral left $\mathcal{O}$-ideal and right $\mathcal{O}_1$-ideal
$\theta \in \mathrm{End}(E)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$\mathrm{nrd}(I_\varphi)$
$\widehat{\varphi}$	$\overline{I_\varphi}$
$\varphi : E \to E_1, \psi : E \to E_1$	Equivalent ideals $I_\varphi \sim I_\psi$
$\tau \circ \rho : E \to E_1 \to E_2$	$I_{\tau\circ\rho} = I_\rho \cdot I_\tau$

Table 1. The Deuring correspondence, a summary given in [24].

2.2 Kani's Lemma

Kani's Lemma [35] gives a criterion to compute isogenies of dimension one using isogenies of dimension two. It was at the heart of the recent SIDH attacks [15,39,45], but it quickly turned into a powerful building block for isogeny-based protocols. We will extensively use it in this work. Our formulation follows [39].

Theorem 1 (Kani). *Let d_1, d_2 and N be pairwise coprime integers such that $N = d_1 + d_2$, and let E_0, E_1, E_2, and E_3 be elliptic curves connected by the following diagram of isogenies:*



such that $\deg(\varphi_1) = \deg(\psi_1) = d_1$, $\deg(\varphi_2) = \deg(\psi_2) = d_2$ *and* $\varphi_2 \circ \varphi_1 = \psi_1 \circ \psi_2$. *Then the map*

$$\Phi = \begin{pmatrix} \varphi_1 & \widehat{\varphi_2} \\ -\psi_2 & \widehat{\psi_1} \end{pmatrix} : E_0 \times E_3 \to E_1 \times E_2$$

is an isogeny of (principally polarized) abelian varieties with kernel

$$\ker(\Phi) = \{(\widehat{\varphi_1}(P), \varphi_2(P)) \mid P \in E_1[N]\} \cong \frac{\mathbb{Z}}{N\mathbb{Z}} \times \frac{\mathbb{Z}}{N\mathbb{Z}}.$$

Assuming that N is powersmooth, or N is smooth and all N-torsion points are rational, the isogeny Φ can be efficiently evaluated at any point on $E_0 \times E_3$. E.g., if $N = 2^a$ for some $a \geq 1$ then one can use [21]. In this case, the generators of the kernel defining Φ encode an efficient two-dimensional representation of φ_1.

2.3 Ideal To Isogeny Translation

Translating ideals into the corresponding isogenies under the Deuring correspondence is a fundamental task in isogeny-based cryptography. A first breakthrough was made in 2014 by Kohel, Lauter, Petit and Tignol [36], who introduced the KLPT algorithm that is at the heart of the SQIsign protocol. The KLPT algorithm can be turned into a polynomial-time method for converting ideals to isogenies, but this is usually inefficient in practice, due to the large degree of the auxiliary isogeny appearing in the process, whose kernel elements are in general defined over large field extensions only. Recently, higher dimensional isogenies gave a new algorithmic tool for converting ideals to isogenies, as demonstrated in the recent versions of the SQIsign signature scheme [6,41,28]. In particular, the IdealToIsogeny algorithm from SQIsign2D-West [6] can efficiently translate *any* left $\mathcal{O}_0$-ideal, where $\mathcal{O}_0 = \mathrm{End}(E_0)$, to the corresponding isogeny originating at E_0 when working over $\mathbb{F}_{p^2}$ with $p = f2^e - 1$, for some small odd $f > 0$.

Let I be a left $\mathcal{O}_0$-ideal. Their algorithm consists of four main steps:

1. Find $I_1, I_2 \sim I$ of coprime norms $d_1, d_2 \approx \sqrt{p}$ and u, v such that $d_1 u + d_2 v = 2^e$, with $d_1 u$ coprime to $d_2 v$;
2. Evaluate two isogenies φ_u, φ_v of degrees u, v on $E_0[2^e]$;
3. Letting φ_1 be the isogeny associated to I_1, use Kani's Lemma to compute the isogeny $\varphi_1 \circ \widehat{\varphi_u}$;
4. Use $\varphi_1 \circ \widehat{\varphi_u}$ to recover φ_I.

We now briefly detail each step. A more detailed description of each step can be found in [6, §4.2]. The first step consists in sampling elements β_1 and β_2 in I until the (reduced) norms d_1, d_2 of $I_1 = I\overline{\beta_1}/\mathrm{nrd}(I)$, $I_2 = I\overline{\beta_2}/\mathrm{nrd}(I)$ satisfy the equation $d_1 u + d_2 v = 2^e$ for some $u, v > 0$. The chance of finding a valid pair is higher for smaller d_1, d_2. For this reason, one needs to select β_1 and β_2 among the shortest vectors of I; a heuristic argument detailed in [6, Section 4.2] suggests that a solution can be found efficiently, and this is verified experimentally. We

note that, unlike other similar algorithms, this algorithm does not impose a bound on the norm of the ideal I that is translated.

The second step is a direct application of the QFESTA algorithm [40] for computing isogenies of fixed degree for the coefficients u and v. This method is also discussed in Appendix A.2.

For the third step, let φ_1 be the isogeny associated to I_1, and φ_2 the one associated to I_2. The main idea is inspired by [43]. It consists of embedding the isogenies $\varphi_1 \circ \widehat{\varphi}_u$ and $\varphi_2 \circ \widehat{\varphi}_v$, of degree $d_1 u$ and $d_2 v$ respectively, into a $(2^e, 2^e)$-isogeny using Kani's Lemma. To determine the kernel of this isogeny, we can use a method similar to the one employed for the Step 2 (again see also Appendix A.2). Finally, from this higher dimensional isogeny, it is possible to evaluate $\varphi_1 \circ \widehat{\varphi}_u$, and use it to recover φ_I by applying [6, Lemma 11].

2.4 Identification Protocols and Digital Signatures

Identification protocols and digital signatures are basic cryptographic building blocks that share some similarities, for example a digital signature scheme implies an identification protocol (see [48, Section 10.3]), but have important differences in their definition and security notions.

Definition 1 (Definition 18.1 [12]). *An identification protocol is a triple of probabilistic polynomial-time algorithms* $(\mathsf{KeyGen}, \mathsf{Pvr}, \mathsf{Vrf})$ *such that:*

- $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, is a probabilistic key generation algorithm, that takes as input a security parameter* λ*, and outputs a pair* $(\mathsf{vk}, \mathsf{sk})$*, where* vk *is called the verification key and* sk *is called the secret key;*
- $\mathsf{Pvr}(\mathsf{sk})$*, is an interactive protocol algorithm, called the prover, that takes as input the secret key* sk*;*
- $\mathsf{accept}/\mathsf{reject} \leftarrow \mathsf{Vrf}(\mathsf{vk})$ *is a probabilistic interactive protocol algorithm, called the verifier, that takes as input the verification key* vk *and outputs* accept *or* reject*.*

We model the interaction with the following notation: $\mathsf{output} \leftarrow \mathsf{Vrf}(\mathsf{vk}) \rightleftharpoons \mathsf{Pvr}(\mathsf{sk})$.

We say that the identification scheme is *correct* if Vrf outputs `accept` with probability 1 over the choice of $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and over the randomness in the involved algorithms, i.e., if

$$\Pr\left[\mathsf{accept} \leftarrow \mathsf{Vrf}(\mathsf{vk}) \rightleftharpoons \mathsf{Pvr}(\mathsf{sk})\right] = 1.$$

We consider the security against active attacks, following Boneh and Shoup [12, Definition 18.8].

Definition 2 ([12]). *Consider the following three phase impersonator game for an adversary* $\mathcal{A}$*:*

- ***Setup****: a key pair* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *is generated and the adversary* $\mathcal{A}$ *receives* vk*;*

– ***Probing***: *the adversary* $\mathcal{A}$ *can interact multiple times with an honest prover* $\mathsf{Pvr}(\mathsf{sk})$ *and store the interaction outputs in a state* st*:*

$$\mathsf{st} \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{st}) \leftrightharpoons \mathsf{Pvr}(\mathsf{sk});$$

– ***Impersonation***: *the adversary* $\mathcal{A}(\mathsf{vk}, \mathsf{st})$ *interacts with an honest verifier* $\mathsf{Vrf}(\mathsf{vk})$*:*

$$\mathsf{output}_{\mathcal{A}} \leftarrow \mathsf{Vrf}(\mathsf{vk}) \leftrightharpoons \mathcal{A}(\mathsf{vk}, \mathsf{st}).$$

The adversary wins if the verifier accepts, i.e., $\mathsf{output}_{\mathcal{A}} = \mathsf{accept}$.

An identification protocol is secure against active attacks if for any PPT adversary $\mathcal{A}$ *playing the impersonator game it is not able to authenticate with non-negligible probability, i.e.,* $\Pr\left[\mathsf{output}_{\mathcal{A}} = \mathsf{accept}\right] = \mathsf{negl}(\lambda)$.

A particular class of identification protocols are Σ-protocols. These are three-round interactive protocols usually referred to as the commitment, challenge and response phase, represented by a transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$. An example of a Σ-protocol is the identification protocol underlying SQIsign, that we recall in Section 2.5. Due to the Fiat–Shamir transform [31], secure Σ-protocols can be efficiently transformed into secure signature schemes. Thus, many signature schemes, such as SQIsign, are constructed in this manner.

Definition 3. *A digital signature scheme consists of three probabilistic polynomial-time algorithms* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrf})$ *such that:*

– $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*: On input a security parameter* λ*, the key generation algorithm outputs a pair of verification and signing keys* $(\mathsf{vk}, \mathsf{sk})$*;*
– $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$*: On input a signing key* sk *and a message* msg*, the signing algorithm outputs a signature* σ*;*
– $\mathsf{accept}/\mathsf{reject} \leftarrow \mathsf{Vrf}(\mathsf{vk}, \mathsf{msg}, \sigma)$*: On input a verification key* vk*, a message* msg *and a signature* σ*, the verification algorithm outputs* accept *or* reject.

A signature scheme is correct if, given $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, for any message msg and signature $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$ a run of $\mathsf{Verify}(\mathsf{vk}, \mathsf{msg}, \sigma)$ outputs accept with probability 1.

Definition 4. *A digital signature is secure in the EUF-CMA model if for any PPT adversary* $\mathcal{A}$ *playing the game from Figure 2 it is not able to obtain a valid signature on a non-queried message, i.e.,*

$$\mathsf{Adv}_{\mathcal{A}}^{\text{EUF-CMA}}(\lambda) = \Pr\left[G^{\mathsf{uf}}(\mathcal{A}) = \mathsf{win}\right] = \mathsf{negl}(\lambda). \tag{2}$$

$G^{\mathsf{uf}}(\mathcal{A})$:
1: $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2: $\mathcal{M} \leftarrow \emptyset$
3: $(\mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'}(\mathsf{vk})$
4: **assert** $\mathsf{msg}^* \notin \mathcal{M}$
5: **assert** $\mathsf{Vrf}(\mathsf{vk}, \mathsf{msg}^*, \sigma^*) = \mathsf{accept}$
6: **return** win

$\mathsf{Sign}'(\mathsf{msg})$:
1: $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$
2: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathsf{msg}\}$
3: **return** σ

Fig. 2. Security game for the EUF-CMA property

2.5 SQIsign and its variants

SQIsign [23] is a signature scheme derived from an isogeny-based Σ-protocol. It is represented by the following diagram:

$$\begin{array}{ccc} E_0 & \xrightarrow{\psi_{\mathsf{com}}} & E_{\mathsf{com}} \\ \downarrow{\scriptstyle \phi_{\mathsf{sk}}} & & \searrow{\scriptstyle \phi_{\mathsf{chall}}} \\ E_{\mathsf{vk}} & \xrightarrow{\sigma_{\mathsf{resp}}} & E_{\mathsf{chall}} \end{array} \tag{3}$$

Here, ϕ_{sk} is the secret key with corresponding verification key E_{vk}, E_{com} is the commitment, and ϕ_{chall} is the challenge. As the prover knows the secret isogeny ϕ_{sk}, they have knowledge of the endomorphism ring of E_{vk}. In this way, the prover is the only party capable of producing a valid response σ_{resp} connecting E_{vk} to the challenge curve E_{chall}. To construct the response isogeny, the prover finds the ideal I corresponding to the composition $\phi_{\mathsf{chall}} \circ \psi_{\mathsf{com}} \circ \widehat{\phi}_{\mathsf{sk}}$, then it finds an *equivalent* ideal J not factoring through the secret key ϕ_{sk}, and translates J to its corresponding isogeny σ_{resp}. In the original SQIsign protocol [23] this is done with (a variant of) the KLPT algorithm, that finds J of smooth norm. Instead, in the subsequent *higher-dimensional variants* [6,28,20,41], they can pose milder restrictions on $\mathrm{nrd}(J)$ and use Kani's Lemma (Section 2.2) to represent the isogeny. A more fine-grained discussion can be found in Appendix A.1.

High Degree Oracles and HVZK property. To show that a Σ-protocol satisfies the Honest-Verifier Zero Knowledge property we need to show how to simulate a valid transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$ in polynomial time, without access to the secret key. The natural simulation strategy for the Σ-protocol underlying SQIsign in (3) is to first generate at random $\sigma_{\mathsf{resp}} : E_{\mathsf{vk}} \to E_{\mathsf{chall}}$, then $\widehat{\phi}_{\mathsf{chall}} : E_{\mathsf{chall}} \to E'$,[7] and finally define $E_{\mathsf{com}} := E'$.

In the original SQIsign identification protocol both the challenge isogeny ϕ_{chall} and the response isogeny σ_{resp} have smooth degree, thus they can efficiently

[7] The order of computing ϕ_{chall} and σ_{resp} may be inverted in some variants, but the simulation strategies rely on the same machinery.

be sampled to simulate a transcript, though arguing the indistingishability of the transcript requires an *ad-hoc* assumption. This is not true for the high-dimensional variants: the response isogeny σ_{resp} has a larger non-smooth degree that can be efficiently represented via high-dimensional isogenies to allow the verifier to compute it. However, this representation cannot be provided without the knowledge of the endomorphism ring of the public curve E_{vk}. To overcome this problem, all the security proofs of the higher-dimensional variants rely on auxiliary oracles that provide efficient representations of uniformly random non-smooth degree isogenies. These oracles can be characterized as providing:

- isogenies of a fixed degree given as input, FIDIO [6, Definition 23] and AIO [28, Definition 4];
- isogenies of a random degree satisfying specific conditions, like RUGDIO [20, Definition 20] and RUNDIO [41, Definition 2];
- uniformly random isogenies of uniformly random bounded degree, like RADIO [20, Definition 41] and UTO [6, Definition 21];
- efficient representation of isogenies given their non-smooth kernel, like FIXDIO [38, Definition 4], RUCODIO [28, Definition 3] and RUCGDIO [28, Definition 2].

Though these oracles vary in flavor, they are all needed for the same core reason: to the best of our knowledge, there are no efficient algorithms to compute large prime degree isogenies without leveraging the knowledge of the endomorphism ring of the domain curve.

Although, we do not know how to construct any of these oracles in polynomial time, it is conjectured that a bounded number of queries do not provide any help in compromising the security of the schemes, e.g., by recovering non-trivial endomorphisms. This was first argued in [20, Section 5.3].

3 Identification Protocol and Digital Signature Scheme

In this section, we present a new identification scheme built from the conjectured hardness of constructing large prime degree isogenies from E, without knowledge of the endomorphism ring $\mathrm{End}(E)$. After constructing an appropriate hash function, this identification protocol can easily be transformed into a simple signature scheme, built from the same subroutines. We can frame this last scheme as a *hash-and-sign*-like signature where the trapdoor one-way function consists of the degree evaluation of isogenies, with domain E_{vk}, of large prime degree. Indeed, given an isogeny, its degree can be efficiently recovered by anyone; however, inverting the trapdoor consists of sampling a large prime degree isogeny, a task considered to be hard without access to the endomorphism ring of E_{vk}.

3.1 Identification Protocol

The core idea behind our identification protocol is that computing isogenies of large prime degree from a random elliptic curve with unknown endomorphism

ring is believed to be hard. Indeed, the best known algorithm for computing an isogeny of prime degree q runs in $O(q^{3/2})$ (see Section 4.4). However, this task significantly simplifies with knowledge of the endomorphism ring of the domain elliptic curve. From these two observations, we can now construct an identification scheme: a party can prove knowledge of the endomorphism ring of a given curve E_{vk} by publishing an isogeny of large prime degree $\varphi : E \to E'$. To instantiate the protocol we fix a base prime $p \equiv 3 \bmod 4$ and an integer a such that we have $\mathbb{F}_{p^2}$-rational 2^a-torsion on any supersingular elliptic curve on $\mathbb{F}_{p^2}$ (when working with models having $(p+1)^2$ rational points). For the same number a, we define Primes_a to be the set of primes of exactly a bits, i.e., primes q such that $2^{a-1} < q < 2^a$.

The identification scheme, referred to as PRISM-id and depicted in Figure 3, relies on the following subroutines:

- $\varphi \leftarrow \mathsf{GenIsogeny}(E, \phi, q)$: on input a supersingular elliptic curve E, an isogeny $\phi : E_0 \to E$ (that gives access to the endomorphism ring of E) and a prime q, return an efficient representation of a cyclic isogeny $\varphi : E \to E'$ of degree $q(2^a - q)$; furthermore, φ is uniformly distributed among all cyclic isogenies of degree $q(2^a - q)$ from E.
- $\mathsf{accept/reject} \leftarrow \mathsf{VerIsogeny}(\varphi, E, q)$: on input an efficient representation of an isogeny $\varphi : E \to E'$, verify that it has degree $q(2^a - q)$.

Prover		**Verifier**
$\mathsf{sk} = \phi_{\mathsf{sk}}$		$\mathsf{vk} = E_{\mathsf{vk}}$
	$\xleftarrow{q}$	Sample $q \in \mathsf{Primes}_a$
$\varphi \leftarrow \mathsf{GenIsogeny}(\mathsf{vk}, \mathsf{sk}, q)$		
	$\xrightarrow{\varphi}$	
		if $\mathsf{VerIsogeny}(\varphi, E_{\mathsf{vk}}, q)$ **return** accept

Fig. 3. PRISM-id identification scheme.

Security. The security of these protocols rests on the following assumption: given many isogenies $\varphi_i : E \to E_i$ of degree $q_i(2^a - q_i)$ with q_i an a-bit prime number, the verifier does not learn any information about the endomorphism ring of E. This assumption is plausible as there are already efficient ways to compute large (yet smooth) degree isogenies from a given curve, regardless of any knowledge of the endomorphism ring. Moreover, because Primes_a is defined as the set of primes having a bits exactly, the cofactor $2^a - q$ cannot be another a-bit prime, so seeing isogenies of degrees corresponding to $q_1, \ldots, q_r \in \mathsf{Primes}_a$ does not help to respond to a degree q_{r+1} that has not been queried before. The security of the schemes and the underlying assumptions are discussed in greater detail in Section 4.

Remark 1. The degree of the isogeny ϕ has the form $q(2^a - q)$ so that ϕ can be represented in two dimensions (rather than four) by using Kani's lemma, which results in a more efficient protocol. The verification procedure thus involves computing a two-dimensional isogeny, which is significantly more efficient than the analogous computation in dimension four.

For a slightly different perspective, write the response isogeny ϕ as the composition $\phi = \phi_{2^a-q} \circ \phi_q$, where $\deg \phi_{2^a-q} = 2^a - q$ and $\deg \phi_q = q$. The isogeny ϕ_q can be interpreted as the "real" response isogeny, whereas ϕ_{2^a-q} is just an auxiliary isogeny that is needed to obtain a two-dimensional representation.

It is also possible to instantiate PRISM with higher-dimensional isogenies, which would eliminate the need for an auxiliary isogeny and would lead to an even more compact protocol (four-dimensional representations require smaller-order torsion points), but would also come at the cost of a much slower verification procedure (extrapolating from the software given in [2], four-dimensional isogenies in SQISign-HD verification are roughly 7 times slower than our verification, for similar parameter sets).

3.2 Signature scheme

If we also have access to a collision resistant hash function on the set of large primes $\mathsf{H}_{\mathsf{prime}} : \{0,1\}^* \to \mathsf{Primes}_a$ we can define a digital signature, called PRISM-sig, using the same building blocks of the identification protocol based on the well-known hash-and-sign paradigm in which we hash a given message to a set of suitable prime degrees, and give an isogeny of that degree as its signature. This is explained in Figure 4. We highlight the simplicity of this scheme: it directly reduces to hard problems in isogeny computation, simplifying its analysis in terms of both efficiency and security.

Signer		**Verifier**
$\mathsf{sk} = \phi_{\mathsf{sk}}$		$\mathsf{vk} = E_{\mathsf{vk}}$
$q \leftarrow \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \Vert \mathsf{msg})$		
$\sigma \leftarrow \mathsf{GenIsogeny}(\mathsf{vk}, \mathsf{sk}, q)$		
	$\xrightarrow{\mathsf{msg},\sigma}$	
		$q \leftarrow \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \Vert \mathsf{msg})$
		if $\mathsf{VerIsogeny}(\sigma, E_{\mathsf{vk}}, q)$
		return accept

Fig. 4. PRISM-sig signature scheme.

In Section 3.3 we carefully define the necessary subroutines for the two schemes. In Appendix B, we also give an alternative construction for GenIsogeny. Although this second method is less efficient than the first, it allows for different parameter choices and can hence become useful in specific settings. Furthermore, this exhibits an important property of our schemes: they are flexible and easily

enable many tweaks and adaptations, a property that paves the way to more advanced constructions.

3.3 Subroutines based on IdealToIsogeny

In the previous sections, we introduced our constructions for an identification protocol and signature scheme, however we did not specify how key generation, or the subroutines GenIsogeny and VerIsogeny are constructed. For the signature scheme, we also need to specify the hash function $\mathsf{H}_{\mathsf{prime}}$. We first present our main construction, which relies on the IdealToIsogeny algorithm introduced in Section 2.3. In what follows, we fix $p = f2^e - 1$ for some small integer $f > 0$. This choice is motivated by the need to access rational 2^e-torsion during the IdealToIsogeny algorithm. However, alternative choices are possible. An example is presented in Appendix B. Let $E_0 : y^2 = x^3 + x$ be the supersingular curve with j-invariant 1728, and fix a basis P_0, Q_0 of $E_0(\mathbb{F}_{p^2})[2^a]$, for some $a \leq e$.

Key generation. The key generation procedure is performed by the algorithm KeyGen, which on input the security parameter λ, outputs the public and secret key pair $(\mathsf{vk}, \mathsf{sk})$ as follows. First, sample a random ideal I_{sk} of a random norm $N_{\mathsf{sk}} = \ell_{\mathsf{large}}^n$ with ℓ_{large} being a large prime greater than 2^a and n such that we get a distribution statistically close to uniform. Use IdealToIsogeny to compute the isogeny $\phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$ corresponding to the ideal I_{sk}. By construction, E_{vk} is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Finally, deterministically compute generators $P_{\mathsf{vk}}, Q_{\mathsf{vk}}$ of $E_{\mathsf{vk}}(\mathbb{F}_{p^2})[2^a]$. The secret key sk is set to be $(\phi_{\mathsf{sk}}, I_{\mathsf{sk}})$, and the public key $\mathsf{vk} = (E_{\mathsf{vk}}, P_{\mathsf{vk}}, Q_{\mathsf{vk}})$. Note that $P_{\mathsf{vk}}, Q_{\mathsf{vk}}$ can be built deterministically from E_{vk}.

Isogeny Generation. The algorithm $\mathsf{GenIsogeny}(\mathsf{vk}, \mathsf{sk}, q)$, given in Algorithm 1, is constructed as follows. Parse vk as $E_{\mathsf{vk}}, P_{\mathsf{vk}}, Q_{\mathsf{vk}}$ and sk as $\phi_{\mathsf{sk}}, I_{\mathsf{sk}}$. As the endomorphism ring of E_0 is known (see Example 1), via the secret isogeny $\phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$ we have knowledge of the endomorphism ring of E_{vk}. We can use this to generate an isogeny $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$. More precisely, we compute a two-dimensional representation of the isogeny $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$, consisting of the codomain curve E_{sig} and the image of the basis $(P_{\mathsf{vk}}, Q_{\mathsf{vk}})$ of $E_{\mathsf{vk}}[2^a]$ through σ.

The first step is to generate an $\mathcal{O}_0$-ideal I_{chall} of norm $q(2^a - q)$. This is done using the RandomFixedNormIdeal algorithm from SQIsign2D-West [6]. Due to the definition of N_{sk}, the ideals I_{sk} and I_{chall} have coprime norm, thus we can compute the pushforward of I_{chall} through the secret ideal I_{sk} using Equation (1):

$$I_\sigma = [I_{\mathsf{sk}}]_* I_{\mathrm{chall}} = \frac{1}{N_{\mathsf{sk}}} \overline{I_{\mathsf{sk}}}(I_{\mathsf{sk}} \cap I_{\mathrm{chall}}).$$

From this, we see that the ideal corresponding to the isogeny $\varphi = \sigma \circ \phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{sig}}$ is given by $I = I_{\mathsf{sk}} I_\sigma = I_{\mathsf{sk}} \cap I_{\mathrm{chall}}$. Using IdealToIsogeny from Section 2.3, we

can obtain a representation of φ and compute the images $\varphi(P_0), \varphi(Q_0)$. Finally, we obtain a two-dimensional representation of σ as follows. For $P \in E_{\mathsf{vk}}[2^a]$ we have

$$\sigma(P) = \frac{1}{N_{\mathsf{sk}}}(\varphi \circ \widehat{\phi_{\mathsf{sk}}})(P).$$

If we write $P_{\mathsf{vk}} = [m_{11}]\phi_{\mathsf{sk}}(P_0) + [m_{12}]\phi_{\mathsf{sk}}(Q_0)$ for $m_{11}, m_{12} \in \mathbb{Z}/2^a\mathbb{Z}$, we see that

$$\frac{1}{N_{\mathsf{sk}}}\widehat{\phi_{\mathsf{sk}}}(P_{\mathsf{vk}}) = [m_{11}]P_0 + [m_{12}]Q_0.$$

So, we compute $P_{\mathsf{sig}} := \sigma(P_{\mathsf{vk}})$ as

$$P_{\mathsf{sig}} = \frac{1}{N_{\mathsf{sk}}}(\varphi \circ \widehat{\phi_{\mathsf{sk}}})(P_{\mathsf{vk}}) = [m_{11}]\varphi(P_0) + [m_{12}]\varphi(Q_0),$$

noting that m_{11}, m_{12}, $\varphi(P_0)$ and $\varphi(Q_0)$ are known. We similarly compute $Q_{\mathsf{sig}} := \sigma(Q_{\mathsf{vk}})$ by instead writing $Q_{\mathsf{vk}} = [m_{21}]\phi_{\mathsf{sk}}(P_0) + [m_{22}]\phi_{\mathsf{sk}}(Q_0)$ for $m_{21}, m_{22} \in \mathbb{Z}/2^a\mathbb{Z}$. The output is the two-dimensional representation of σ given by $\mathsf{iso} := (E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}})$.

Remark 2. The RandomFixedNormIdeal algorithm from SQIsign2D-West [6] outputs a uniformly random primitive ideal, therefore the output of GenIsogeny is a uniformly random cyclic isogeny of degree $q(2^a - q)$ from E_{vk}.

Remark 3. To make the GenIsogeny procedure more efficient, we can precompute the integers m_{ij} in KeyGen and store them in the secret key.

Algorithm 1 GenIsogeny$(\mathsf{vk}, \mathsf{sk}, q)$

Input: Secret key $\mathsf{sk} = \phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$ of degree N_{sk} with corresponding ideal I_{sk}, public key $\mathsf{vk} = (E_{\mathsf{vk}}, P_{\mathsf{vk}}, Q_{\mathsf{vk}})$ and prime number $q \in (2^{a-1}, 2^a)$.
Output: A two-dimensional representation of $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$.

1: Find $m_{11}, m_{12} \in \mathbb{Z}/2^a\mathbb{Z}$ such that $P_{\mathsf{vk}} = [m_{11}]\phi_{\mathsf{sk}}(P_0) + [m_{12}]\phi_{\mathsf{sk}}(Q_0)$;
2: Find $m_{21}, m_{22} \in \mathbb{Z}/2^a\mathbb{Z}$ such that $Q_{\mathsf{vk}} = [m_{21}]\phi_{\mathsf{sk}}(P_0) + [m_{22}]\phi_{\mathsf{sk}}(Q_0)$;
▷ *Coefficients from Lines 1, 2 can be precomputed in* KeyGen
3: $N \leftarrow q(2^a - q)$;
4: $I_{\text{chall}} \leftarrow$ RandomFixedNormIdeal $(\mathcal{O}_0, N)$;
5: $I \leftarrow I_{\mathsf{sk}} \cap I_{\text{chall}}$;
6: $\varphi \leftarrow$ IdealToIsogeny(I);
7: Set E_{sig} as the codomain of φ;
8: $P_{\mathsf{sig}} = [m_{11}]\varphi(P_0) + [m_{12}]\varphi(Q_0)$;
9: $Q_{\mathsf{sig}} = [m_{21}]\varphi(P_0) + [m_{22}]\varphi(Q_0)$;
10: **return** $(E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}})$

Verification. We now define the verification algorithm $\mathsf{VerIsogeny}(\mathsf{iso}, E_{\mathsf{vk}}, q)$. Parse the input iso as $(E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}})$. From $\mathsf{GenIsogeny}$ we get $(P_{\mathsf{sig}}, Q_{\mathsf{sig}}) = (\sigma(P_{\mathsf{vk}}), \sigma(Q_{\mathsf{vk}}))$, where we recall that $E_{\mathsf{vk}}[2^a] = \langle P_{\mathsf{vk}}, Q_{\mathsf{vk}} \rangle$. We have to check that these points interpolate an isogeny $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$.

Write $\sigma = \varphi_q \circ \varphi_{q'} = \psi_{q'} \circ \psi_q$, where $\deg(\psi_q) = \deg(\varphi_q) = q$, and $\deg(\psi_{q'}) = \deg(\varphi_{q'}) = 2^a - q$. We can then factor σ using the commuting square

$$\begin{array}{ccc} E_{\mathsf{vk}} & \xrightarrow{\varphi_{q'}} & E_2 \\ \downarrow{\scriptstyle \psi_q} & & \downarrow{\scriptstyle \varphi_q} \\ E_1 & \xrightarrow{\psi_{q'}} & E_{\mathsf{sig}} \end{array}$$

This is a $(q, 2^a - q)$-isogeny diamond, and therefore, by Kani's Lemma, the isogeny

$$\Phi \colon E_{\mathsf{vk}} \times E_{\mathsf{sig}} \to E_1 \times E_2$$

given by the matrix

$$\Phi = \begin{pmatrix} \psi_q & \widehat{\psi}_{q'} \\ -\varphi_{q'} & \widehat{\varphi}_q \end{pmatrix},$$

is a $(2^a, 2^a)$-isogeny between these products of elliptic curves, viewed with their product polarisation. Furthermore, the kernel of Φ is given by

$$\ker(\Phi) = \{([q]P, \sigma(P)) | P \in E_{\mathsf{vk}}[2^a]\} = \langle ([q]P_{\mathsf{vk}}, P_{\mathsf{sig}}), ([q]Q_{\mathsf{vk}}, Q_{\mathsf{sig}}) \rangle .$$

We further optimize this by asking the prover to return $([q^{-1}]P_{\mathsf{sig}}, [q^{-1}]Q_{\mathsf{sig}})$. This comes at virtually no cost in the signing step: by multiplying the coefficients m_{ij} by q^{-1}, the new points can be obtained with the same number of point multiplications. On the other hand, the verifier has

$$\ker(\Phi) = \left\langle (P_{\mathsf{vk}}, [q^{-1}]P_{\mathsf{sig}}), ([Q_{\mathsf{vk}}, [q^{-1}]Q_{\mathsf{sig}}) \right\rangle$$

readily available. They can therefore compute Φ to verify that $(P_{\mathsf{sig}}, Q_{\mathsf{sig}})$ interpolates the isogeny σ. To verify the degrees, we first compute $(P', \_) := \Phi((P_{\mathsf{vk}}, 0))$ and $(Q', \_) := \Phi((Q_{\mathsf{vk}}, 0))$. Observe that if $(E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}})$ is a valid isogeny, we have

$$e_{2^a}(P', Q') = e_{2^a}(P_{\mathsf{vk}}, Q_{\mathsf{vk}})^n,$$

where $n \in \{q, 2^a - q\}$ and e_{2^a} is the 2^a-Weil pairing. Therefore, we can simply compute $e_{\mathsf{vk}} = e_{2^a}(P_{\mathsf{vk}}, Q_{\mathsf{vk}})$ and $e' = e_{2^a}(P', Q')$ and check whether $e_{\mathsf{vk}} = (e')^q$ or $e' = (e_{\mathsf{vk}})^q$.

Hashing in Primes_a. For our signature scheme, we need to define the hash function $\mathsf{H}_{\mathsf{prime}}$ that hashes into the set Primes_a. To construct $\mathsf{H}_{\mathsf{prime}}$ we consider any cryptographic hash function $\mathsf{H}_{a-2} : \{0,1\}^* \to [0, 2^{a-2})$, composed with $h \mapsto 2^{a-1} + 2h + 1$ (prepending and appending a bit 1 to the binary expansion) to end up with an odd integer in the interval $(2^{a-1}, 2^a)$. Given E_{vk} and msg we define

$\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}}\|\mathsf{msg})$ by repeatedly computing $2^{a-1} + 2\mathsf{H}_{a-2}(E_{\mathsf{vk}}\|\mathsf{msg}\|\mathsf{counter}) + 1$ for increasing values of $\mathsf{counter}$ until we hit a prime number. This requires on average $2^{a-2}/\#\mathsf{Primes}_a \approx a\ln(2)/2$ repetitions; notice that the value of $\mathsf{counter}$ can be provided to the verifier in Figure 4 at a minimal cost, its expected size being about $\log a$ bits, to avoid useless hash computations during verification.

Notice that each step in the hash $\mathsf{H}_{\mathsf{prime}}$ evaluation requires performing a primality test [4], which has a quadratic cost in the bit length a.

Remark 4. A slight faster choice for $\mathsf{H}_{\mathsf{prime}}$ would be to first hash to a 2^a bits odd integer and then increase it sequentially until we reach a prime. However, this would introduce a bias towards primes associated to long intervals of non-prime integers, and we therefore avoid this option.

4 Security

In this section, we prove that both the identification protocol and the signature scheme are deeply connected to the hardness of evaluating large prime degree isogenies, a well understood problem in isogeny-based cryptography.

4.1 Key Recovery

First, note that the key recovery problem for both our constructions is simply the standard *Supersingular Endomorphism Ring Problem*, given below.

Problem 1 (Supersingular Endomorphism Ring Problem). Given a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$, find four (efficient representations of) endomorphisms which generate the ring $\mathrm{End}(E)$.

Both the signing and the identification procedure result in revealing isogenies of large prime degree. These are hard to compute without the knowledge of the endomorphism ring. Moreover, it is believed that revealing such isogenies does not help to solve the endomorphism ring problem. This fact was first formulated in [20] where the authors argued that providing an oracle to produce isogenies of arbitrary degree would not impact the security of SQIsignHD. Furthermore, notice that given a curve E anyone can efficiently compute isogenies of large degree without knowledge of the endomorphism ring, as long as this degree is smooth. Since smooth-degree isogenies are sufficient to cover the whole isogeny graph, for each isogeny we reveal there exists an equivalent, smooth-degree isogeny that is computable without any knowledge of $\mathrm{End}(E_{\mathsf{vk}})$. This lends support to the assumption that our protocols do not leak useful information to an attacker.

4.2 Forgery and Impersonation

The security of the protocol relies on the following (new) assumption. Recall that Primes_a is the set of primes of exactly a bits. Consequently, an element $q \in \mathsf{Primes}_a$ is uniquely determined by the value of $q(2^a - q)$.

Definition 5. *A* special degree isogeny oracle *(SPEDIO) is an oracle which takes as input a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ and a prime $q \in \mathsf{Primes}_a$, and returns a uniformly random cyclic isogeny of degree $q(2^a - q)$ from E.*

Problem 2. Given a random supersingular elliptic curve E and a SPEDIO, output an isogeny of degree $q'(2^a - q')$ with $q' \in \mathsf{Primes}_a$ different from all degrees formerly generated by the oracle.

Remark 5. Using the same notation, Problem 2 is at least as hard as the problem of computing a degree q' isogeny. Indeed, given a degree $q'(2^a - q')$ isogeny, the degree q' component can be recovered in polynomial time by factoring it (as we do in the verification procedure). We note that the converse is not as straightforward, as there may be large prime factors in $(2^a - q')$ that are smaller than 2^{a-1}.

Problem 2 can be summarized by the security game G_E^{pdeg} in Figure 5, where E is a supersingular elliptic curve.

$G_E^{\mathsf{pdeg}}(\mathcal{A})$:
1: $\mathcal{Q} \leftarrow \emptyset$
2: $\sigma^* \leftarrow \mathcal{A}^{\mathsf{SPEDIO}}(E)$
3: **assert** $\sigma^* : E \to E^*_{\mathsf{sig}}$ is an isogeny of degree $q(2^a - q)$
4: **assert** $q \in \mathsf{Primes}_a$
5: **assert** $q \notin \mathcal{Q}$
6: **return** win

$\mathsf{SPEDIO}(q)$:
1: **assert** $q \in \mathsf{Primes}_a$;
2: Sample isogeny $\sigma : E \to E'$ of degree $q(2^a - q)$;
3: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$;
4: **return** $\sigma : E \to E_{\mathsf{sig}}$

Fig. 5. Security game for Problem 2

We now show the straightforward relation between the hardness of Problem 2, the security against adaptive attacks of PRISM-id and the unforgeability of PRISM-sig.

Proposition 1. *Under the assumption that Problem 2 is hard, any PPT adversary against adaptive attacks (Definition 2) of* PRISM-id *performing N interactions has a winning probability bounded by $N/\#\mathsf{Primes}_a$.*

Proof. Let $\mathcal{A}$ be an adversary for the impersonator game in Definition 2 with N interactions. Given the supersingular elliptic curve E, we simulate the impersonator game for $\mathcal{A}$ to win the game G_E^{pdeg} in Figure 5, i.e., solve Problem 2. We proceed as follows:

– During **Setup** we set $E_{\mathsf{vk}} = E$ as public key and send it to $\mathcal{A}$;

– During the *probing phase* we use the oracle SPEDIO (defined in Problem 2) to perform N interactions with $\mathcal{A}$, let $\mathcal{Q}$ be the set of queried degrees, which is a set of size at most N. By Remark 2, the output of SPEDIO has the same distribution as the response of an honest prover;
– In the *impersonation phase* the adversary $\mathcal{A}$ wins, i.e., $\mathsf{output}_{\mathcal{A}} = \mathsf{accept}$, if and only if $\mathcal{A}$ can provide an isogeny $\sigma : E \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$ for a uniformly random $q \in \mathsf{Primes}_a$.

If $q \notin \mathcal{Q}$, the isogeny σ is a valid solution for G_E^{pdeg}, against the hardness of Problem 2. Thus, the winning probability of $\mathcal{A}$ is bounded by

$$\Pr\left[q \in \mathcal{Q}\right] = \frac{\#\mathcal{Q}}{\#\mathsf{Primes}_a} \leq \frac{N}{\#\mathsf{Primes}_a}$$

as required. □

We now establish the security of $\mathsf{PRISM\text{-}sig}$ in the standard model, leveraging only the collision resistance property of $\mathsf{H}_{\mathsf{prime}}$ and the hardness of Problem 2. Although this proof is inherently designed to simulate a signing procedure that returns a new random signature per each message msg, it can be straightforwardly adapted to always return the same isogeny when the same message is queried.

Proposition 2. *If* $\mathsf{H}_{\mathsf{prime}}$ *is a collision-resistant cryptographic hash function and Problem 2 is hard, then* $\mathsf{PRISM\text{-}sig}$ *is EUF-CMA secure (Definition 4).*

Proof. We show that given a PPT adversary $\mathcal{A}$ in the EUF-CMA model we can use it to win G_E^{pdeg} (that is equivalent to solving Problem 2) or find a collision for $\mathsf{H}_{\mathsf{prime}}$ in polynomial time. Given the supersingular elliptic curve E, we set it as a public key E_{vk} in our signature scheme. For every message query msg_i, we evaluate the prime $q_i = \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}_i) \in \mathsf{Primes}_a$ and query the oracle $\mathsf{SPEDIO}(q_i)$ to get an isogeny $\sigma_i : E \to E_{\mathsf{sig}}$ of degree $q_i(2^a - q_i)$ and we return it as a signature. By Remark 2, these signatures follow the same distribution as honestly generated signatures for the public key E_{vk}. The strategy is described in Figure 6. The set $\mathcal{M}$ contains the previously queried messages, while $\mathcal{Q}$ collects the values of $\mathsf{H}_{\mathsf{prime}}$ applied to the elements in $\mathcal{M}$.

Since σ^* is a valid signature (due to the assert in Line 4) it satisfies the assertions in Lines 3 and 4 from Figure 5, i.e., it corresponds to a valid prime degree isogeny. We need only to check that it has not already been returned from SPEDIO before (see Line 5). If $q^* = \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}^*) \notin \mathcal{Q}$, then σ^* from Line 7 has a degree different from all isogenies previously returned. Indeed, recall that for all $q \in \mathsf{Primes}_a$, we have $q > 2^a - q$, so for any $q \neq q^* \in \mathsf{Primes}_a$, we have $q(2^a - q) \neq q^*(2^a - q^*)$. So, it is a valid solution to win G_E^{pdeg}.

On the other hand, if $\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}^*) \in \mathcal{Q}$ then in Line 9 we return a collision since we have found a message $\mathsf{msg}_i \in \mathcal{M}$ such that $\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}_i) = \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}^*)$, but $\mathsf{msg}^* \notin \mathcal{M}$. □

$\mathcal{A}'(\mathcal{A})$:

1: $E_{\mathsf{vk}} \leftarrow E$
2: $\mathcal{M}, \mathcal{Q} \leftarrow \emptyset$
3: $\mathsf{msg}^*, \sigma^* \leftarrow \mathcal{A}^{\mathsf{Sign}}(E_{\mathsf{vk}})$
4: **assert** $\mathsf{Verify}(E_{\mathsf{vk}}, \sigma^*, \mathsf{msg}^*)$
5: **assert** $\mathsf{msg}^* \notin \mathcal{M}$
6: **if** $\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg}^*) \notin \mathcal{Q}$ **then**
7: | **return** σ^*;
8: **else**
9: | **return** collision for $\mathsf{H}_{\mathsf{prime}}$;

$\mathsf{Sign}(\mathsf{msg})$:

1: $q \leftarrow \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg})$;
2: $\sigma \leftarrow \mathsf{SPEDIO}(q)$ of degree $q(2^a - q)$;
3: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$;
4: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathsf{msg}\}$;
5: **return** σ

Fig. 6. Reduction from adversary $\mathcal{A}$ for PRISM-sig

4.3 Unforgeability in the ROM

Interestingly, if we model our hash function as a random oracle we can give a security proof under a weaker hardness assumption. Namely, we can consider a variant of Problem 2 in which the adversary has no control over the non-smooth isogenies being provided, nor in the degree of the output isogeny they forge.

Problem 3. Given a random curve E, a set of N isogenies $\{\phi_i : E \to E_i\}_{i=1}^N$ of degree $q_i(2^a - q_i)$ for q_i uniformly random in Primes_a and ϕ_i uniformly random among the isogenies of degree $q_i(2^a - q_i)$, and a prime $\bar{q}$ uniformly random in $\mathsf{Primes}_a \setminus \{q_i\}_{i=1}^N$, give an efficient representation of an isogeny of degree $\bar{q}(2^a - \bar{q})$.

It is immediate to see that if we can solve Problem 3 then we can also solve Problem 2, but potentially not vice-versa. Remark 5 similarly applies to Problem 3.

Proposition 3. *In the random oracle model (ROM), any PPT adversary that wins the EUF-CMA game with advantage ϵ and that performs N_{sign} signing queries and N_{H} hashing queries can be used to solve Problem 3 for $N = N_{\mathsf{sign}} + N_{\mathsf{H}}$ with probability at least $\epsilon N_{\mathsf{H}}^{-2}$.*

Proof. Let $\mathcal{A}$ be a PPT adversary for the EUF-CMA of PRISM-sig. We want to define another PPT algorithm $\mathcal{B}$ that solves Problem 3 for $N = N_{\mathsf{sign}} + N_{\mathsf{H}}$ using $\mathcal{A}$ as a subroutine in the random oracle model. For this $\mathcal{B}$ we need to simulate the answers to the signing and hashing queries. Let $\{\phi_i : E \to E_i\}_{i=1}^N$ be the isogenies of degree $q_i(2^a - q_i)$ for $q_i \in \mathsf{Primes}_a$ received as input from Problem 3, and $\bar{q}$ be the prime degree of the isogeny we have to find to solve Problem 3.

Then, $\mathcal{B}$ fixes $E = E_{\mathsf{vk}}$ and starts to simulate the interactions with $\mathcal{A}$ following the games G_0 and G_1, as explained in Figure 7. Since we are in the ROM, the adversary $\mathcal{A}$ needs to interact with $\mathcal{B}$ to query the hash function $\mathsf{H}_{\mathsf{prime}}$. Let msg^* be the output message of $\mathcal{A}$. We make the following assumptions on the queries:

1. Queries to $\mathsf{H}_{\mathsf{prime}}$ are always of the form $E_{\mathsf{vk}} \| \mathsf{msg}$,

2. $E_{\mathsf{vk}}\|\mathsf{msg}^*$ is part of the N_{H} queries to $\mathsf{H}_{\mathsf{prime}}$, i.e., $\mathsf{msg}^* \in \mathcal{H}$. This is because we can always modify $\mathcal{A}$ to query it before returning msg^* and the signature.
3. $\mathcal{H} \setminus \mathcal{M} \neq \emptyset$, i.e. there is always at least one message queried to $\mathsf{H}_{\mathsf{prime}}$ but not to Sign. In fact by the previous point $\mathsf{msg}^* \in \mathcal{H}$, but by the EUF-CMA definition we need $\mathsf{msg}^* \notin \mathcal{S}$ to have non-zero winning probability, thus at least one element is in the set $\mathcal{H} \setminus \mathcal{M}$.

Setup : G_0 / $\boxed{G_1}$
1: set $j_{\mathsf{H}_{\mathsf{prime}}} \leftarrow 0$, $j_{\mathsf{tot}} \leftarrow 0$;
2: set $\mathcal{M}, \mathcal{H} \leftarrow \emptyset$;
3: initialize empty lists H, S;
4: sample $j_{\bar{q}} \xleftarrow{\$} \{1, \ldots, N_{\mathsf{H}}\}$
5: get $\sigma^*, \mathsf{msg}^* \leftarrow \mathcal{A}^{\mathsf{H}_{\mathsf{prime}}, \mathsf{Sign}}(E_{\mathsf{vk}})$
6: **assert** $\mathsf{msg}^* \notin \mathcal{M}$
7: **assert** $\mathsf{Verify}(E_{\mathsf{vk}}, \mathsf{msg}^*, \sigma^*)$
8: **assert** $\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}}\|\mathsf{msg}^*) = q_{j_{\bar{q}}}$
▷ *main difference with* G^{uf}
9: **return** win

$\mathsf{H}_{\mathsf{prime}}(E\|\mathsf{msg})$:
1: **if** $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] = \perp$ **then**
2: $\quad \mathcal{H} \leftarrow \mathcal{H} \cup \{\mathsf{msg}\}$
3: $\quad j_{\mathsf{H}_{\mathsf{prime}}} \leftarrow j_{\mathsf{H}_{\mathsf{prime}}} + 1$;
4: $\quad j_{\mathsf{tot}} \leftarrow j_{\mathsf{tot}} + 1$;
5: $\quad$ set $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] \leftarrow q_j$;
6: $\quad$ set $\mathsf{S}[\mathsf{msg}] \leftarrow \sigma_j$;
7: **if** $j_{\mathsf{tot}} = j_{\bar{q}}$ **then**
8: $\quad$ set $q_{j_{\bar{q}}} \leftarrow \bar{q}$;
9: $\quad$ set $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] \leftarrow \bar{q}$;
10: **return** $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}]$

$\mathsf{Sign}(\mathsf{msg})$:
1: **if** $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] = q_{j_{\bar{q}}}$ **then**
2: $\quad \mathsf{bad} \leftarrow \mathsf{true}$;
3: **if** $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] = \perp$ **then**
▷ msg *not yet reprogrammed*
4: $\quad j_{\mathsf{tot}} \leftarrow j_{\mathsf{tot}} + 1$;
5: $\quad$ set $\mathsf{H}[E_{\mathsf{vk}}\|\mathsf{msg}] \leftarrow q_i$;
6: $\quad$ set $\mathsf{S}[\mathsf{msg}] \leftarrow \sigma_i$;
7: $\quad \mathcal{M} \leftarrow \mathcal{M} \cup \{\mathsf{msg}\}$;
8: **if** $\mathsf{msg} \in \mathcal{H}$ **then**
9: $\quad \mathcal{M} \leftarrow \mathcal{M} \cup \{\mathsf{msg}\}$; ▷ *already reprogrammed in* $\mathsf{H}_{\mathsf{prime}}$
10: **return** $\mathsf{S}[\mathsf{msg}]$

Fig. 7. Strategy to simulate the signing EUF-CMA model in the ROM

When looking at the game G_0 the only difference from G^{uf}, except for the output type, is the assertion in Line 8. Since $j_{\bar{q}}$ is sampled independently from all the randomness involved in the protocol and used by $\mathcal{A}$, we have that:

$$\Pr[G_0(\mathcal{A}) = \mathsf{win}] = \Pr[G^{\mathsf{uf}}(\mathcal{A}) = \mathsf{win} \wedge \mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}}\|\mathsf{msg}^*) = q_{j_{\bar{q}}}] = \\ = \Pr[G^{\mathsf{uf}}(\mathcal{A}) = \mathsf{win}] \cdot \Pr[\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}}\|\mathsf{msg}^*) = q_{j_{\bar{q}}}] = \epsilon \cdot \frac{1}{N_{\mathsf{H}}} \ . \quad (4)$$

Games G_0 and G_1 are identical until the bad flag is set to true, since $q_{j_{\bar{q}}}$ and $\bar{q}$ have the same distribution. So, the difference is relevant only in the case of a

sign query on the same message. Hence, by standard game-based proof results [3, Lemma 3.7] we have that

$$\Pr[G_0(\mathcal{A}) = \mathsf{win} \wedge \mathsf{Good}_0] = \Pr[G_1(\mathcal{A}) = \mathsf{win} \wedge \mathsf{Good}_1], \tag{5}$$

where Good_i is the event that the flag bad is never changed in the game G_i. Now, observe that in game G_0 there is no reprogramming to $\bar{q}$, thus the two events in the left hand side of Equation (5) are independent. Moreover, as argued before, there is at least one message $\mathcal{H} \setminus \mathcal{M}$, i.e., a message for which G_0 reprograms $\mathsf{H}_{\mathsf{prime}}$ but never queries to Sign. Since $j_{\bar{q}}$ is chosen independently of the randomness used by $\mathcal{A}$ there is at least a probability $1/N_{\mathsf{H}}$ that the reprogrammed message is in $\mathcal{H} \setminus \mathcal{M}$, that implies Good_0. Combining everything we have

$$\Pr[G_1(\mathcal{A}) = \mathsf{win} \wedge \mathsf{Good}_1] \overset{(5)}{=} \Pr[G_0(\mathcal{A}) = \mathsf{win}] \Pr[\mathsf{Good}_0] \overset{(4)}{\geq} \epsilon \frac{1}{N_{\mathsf{H}}^2} .$$

It is clear that if $G_1(\mathcal{A}) = \mathsf{win}$, the isogeny σ^* from Line 5 is a valid solution for Problem 3 that $\mathcal{B}$ can return. □

Remark 6. This reduction strongly relies on the signature being deterministic, as a function of the message msg and the public key E_{vk}. In fact, providing two different signatures for the same message msg requires returning two isogenies of the same degree $\mathsf{H}_{\mathsf{prime}}(E_{\mathsf{vk}} \| \mathsf{msg})$, but the initial data given in Problem 3 only provides us with one isogeny per prime number q_i.

4.4 Best Known Attacks on Hardness Assumptions

In this section we discuss the best known attacks against the hard problems underlying our scheme. This analysis will guide our choice of parameters.

Endomorphism ring problem. As argued above, key recovery against both PRISM-sig and PRISM-id amounts to the computation of the endomorphism ring of the public key E_{vk}. The discussion in [20, Sec. 4] justifies the assumption that E_{vk} is a random curve. Thus, the fastest known algorithms to compute its endomorphism ring have classical complexity in $\widetilde{O}(p^{1/2})$ [25]. The only known quantum speed-up uses Grover's algorithm [34,11], achieving a quantum complexity of $\widetilde{O}(p^{1/4})$. These are the best key recovery attacks against most isogeny-based protocols, including the SQIsign family (see e.g. [6, Sec. 6]). We hence require p to be a prime of at least about 2λ bits for a security level of λ bits.

Computing isogenies of prime degree. The most direct attempt to solve Problems 2 and 3 requires, on input of a prime degree q, to compute a cyclic degree $q(2^a - q)$ isogeny from E_{vk} without the knowledge of $\mathrm{End}(E_{\mathsf{vk}})$. First of all, notice that $q \in \mathsf{Primes}_a$, i.e., it is a prime bigger than 2^{a-1}, and so q is the biggest prime factor of $q(2^a - q)$. As a consequence, the cost of computing a $q(2^a - q)$-isogeny will mostly depend on the cost of computing a degree q isogeny.

In fact, an attacker may even compute isogenies whose degrees are pairwise coprime factors of $q(2^a - q)$ in parallel, all starting from E_{vk}, and then compose them by pushing them forward [46, Prop. 6.15]. The complexity of this approach is dominated by that of computing an isogeny of largest prime power degree. We can thus restrict to studying the cost of computing an isogeny of (large) prime degree q.

There are various methods to compute q-isogenies without relying on the knowledge of the endomorphism ring. An approach is to use Vélu's formulae [50]. These formulae require knowledge of a point of order q. In general, such a point will not be defined over $\mathbb{F}_{p^2}$, but rather over a large field extension. Specifically, we expect this degree to be roughly the size of q [30]. Field operations in an extension of degree q have an overhead that is in $\widetilde{O}(q)$. The amount of field operations needed to compute a degree-q isogeny using Vélu's formulae is again linear in q, and so we obtain a total complexity of $\widetilde{O}(q^2)$. A significant improvement in computing an isogeny from its kernel is achieved through the square-root Vélu algorithm [8]. This algorithm reduces the number of field operations from $\widetilde{O}(q)$ to $\widetilde{O}(q^{1/2})$. As this computation runs over the field extension where a point of order q is defined, the resulting expected complexity of using square-root Vélu algorithm is $\widetilde{O}(q^{3/2})$. However, this complexity assumes that a point of order q is already available, and we must still factor in this cost. One method for obtaining such a point is to sample a random point on the curve $E/\mathbb{F}_{p^q}$ and multiply it by the appropriate cofactor of size around $\frac{p^q}{q}$, by Hasse's Theorem. This requires at least $\log\left(\frac{p^q}{q}\right) \approx q$ point doublings and additions defined on a field extension of degree q, and thus has complexity $\widetilde{O}(q^2)$. Another option is to instead search for its x-coordinate as a root of the q-division polynomial of degree $(q^2-1)/2$. This also requires at least $\widetilde{O}(q^2)$ field operations.

A different approach to computing q-isogenies avoiding large field extensions is to employ kernel polynomials. Once a kernel polynomial has been computed, it is possible to use well-known formulae, such as those in [37], to compute the q-isogeny with $O(q)$ field operations. Moreover, obtaining kernel polynomials without access to an isogeny representation or a point of order q (defined over a large field extension) is also a costly operation. To the best of our knowledge, the fastest method to compute the kernel polynomial is via Elkies algorithm [29] (see, for example, [33, Chapter 25.2.1]). There are two costly steps to this algorihtm. First, we need to find the root of the modular polynomial $\Phi_q(X, Y)$ over $\mathbb{F}_p$, costing $O(q)$ operations, assuming the modular polynomials has already been computed. Secondly, we need to compute q coefficients via a recurrence relation involving all previous coefficients, requiring $\mathcal{O}(q^2)$ operations in $\mathbb{F}_p$. Another method is to factor the q-division polynomial, and extract the kernel polynomial. However, this yields a complexity much larger than the method discussed above.

An alternative approach for computing q-isogenies is to look at the classical modular polynomial $\Phi_q(X, Y)$, which is the polynomial whose roots are pairs of j-invariants of q-isogenous curves. Computing these polynomials for large q is already very challenging. Suppose we want to compute an isogeny originating

from E. For the sake of simplifying the argument, let us assume that it is possible to compute a curve E' which is q-isogenous to E by finding a root of $\Phi_q(j(E), X)$. Even in this case, we still need to find a way to compute an isogeny $\varphi : E \to E'$ of degree q. An algorithm to compute φ is described by Elkies [29]. The complexity of this algorithm is $\widetilde{O}(q^2)$.

The quantum complexity of computing prime degree isogenies has never been thoroughly studied. However, we see no reason to believe that this problem would be amenable to a significant quantum speed-up.

Concluding the discussions above, we estimate the complexity of computing a degree q-isogeny, without knowledge of the endomorphism ring, to be $\widetilde{O}(q^2)$. Hence, when targeting λ-bits security, to guarantee the security of Problems 2 and 3, we only have to impose the mild requirement that $q^2 > 2^\lambda$, which in turn implies $a > \frac{1}{2}\lambda$. Since we already imposed $p \approx 2^{2\lambda}$, this constraint can be easily satisfied. Moreover, notice that this analysis is also quite conservative (e.g., we are not limiting the memory at disposal to the attacker).

Together with the computation of the endomorphism ring, these are the only attacks against PRISM-id we are aware of. Our parameter choice will thus be guided by this analysis. On the other hand, in the construction of PRISM-sig, we have to take into account the security of the hash function, which turns out to be the actual security bottleneck, as argued below. As a consequence, the parameter choice in PRISM-sig has a large margin against attacks based on direct isogeny computations.

Breaking the hash function. First of all, to avoid signature reusing, i.e., to achieve the non-resignability property from [19], we insert the domain isogeny E_{vk} into the hashing input, since this comes at virtually no cost. Then, as our construction PRISM-sig follows a hash-and-sign paradigm, an attacker can obtain forgeries by finding collisions for $\mathsf{H}_{\mathsf{prime}}$.

A priori, it may seem that hashing into Primes_a is not enough to achieve λ-bit security against collision search. Indeed, there are only about $2^{a-1}/(a\ln(2))$ primes in Primes_a. However, we do not hash directly into Primes_a, but rather into the set of odd integers in $(2^{a-1}, 2^a)$ via H_{a-2} and $h \mapsto 2^{a-1} + 2h + 1$, and reject until we find a prime. Recall that the expected number of tries is about $a\ln(2)/2$. Thus, in order to produce a collision for $\mathsf{H}_{\mathsf{prime}}$, the expected number t of calls to H_{a-2} satisfies

$$\binom{t}{2} 2^{-a+2} \approx a\ln(2)/2,$$

(birthday paradox for multiple collisions; see [49, Sec. 4] for algorithmic details). Thus, to get λ-bits collision resistance we want that

$$2^\lambda \leq t \approx 2^{\frac{a-2}{2}}\sqrt{a}, \tag{6}$$

leading to the asymptotic estimate $a \approx 2\lambda - \log\lambda$. Notice that this requirement is much stronger than $a > \frac{1}{2}\lambda$ resulting from the discussion above. On the

other hand, here we have just attributed a cost of 1 to each call to H_{a-2}, which is a very conservative choice. We leave it to the reader to take into account more realistic cost estimates, where one could even use artificially slow hash functions to establish a further reduction of a (but this also affects signing and verification). This trick is reminiscent of other schemes that employ slow hashing or *proof-of-work* [7,10].

It is clear that $a = e \approx 2\lambda$ is good enough to get collision resistance, but we can do slightly better by taking the smallest a satisfying Equation (6). For example, for NIST Level I security (namely, $\lambda = 128$) we can take $a = 251$. Note: in our concrete parameters from Section 5.1 we will choose $a = 248$, but this defect is (amply) compensated by the complexity of evaluating H_{a-2}.

Finally, as was already remarked, given two isogenies of coprime degree from a curve E there exists a polynomial time algorithm (that eventually requires to go to dimension four or eight) to compute the respective pushforwards. This implies that an attacker seeing an isogeny of degree $q(2^a - q)$ can effectively obtain an isogeny for all prime power factors of $2^a - q$ that can then be reused later: the output of the hash function just consists of q. This also explains why we require q to have exactly a bits: in this way we ensure that none of the factors of $2^a - q$ will land in Primes_a, and thus by seeing them an attacker does not learn anything.

5 Implementation and performance

In this section we evaluate the performance of our schemes. We compare PRISM-sig with other recent isogeny-based signatures, and highlight the difference in performance between PRISM-sig and PRISM-id. The repository with the source code can be found at https://github.com/KULeuven-COSIC/PRISM.

5.1 Parameter choices

Following the discussion of Section 4.4, we can now give concrete parameter choices for our schemes. To protect against endomorphism ring computation we require $p \approx 2^{2\lambda}$. To use the IdealToIsogeny algorithm, we also require access to the 2^e-torsion with $2^e \approx p$. This also allows us to represent isogenies of degree up to $2^e \approx 2^{2\lambda}$, satisfying the security requirements against isogeny computation and hash collisions for both PRISM-sig and PRISM-id.

We thus choose primes of the form $p = f2^e - 1$, with f a small cofactor. Such primes are commonly used in isogeny based schemes, for example in SQIsign2D-West. As such, we can follow their parameter choices and exploit the existing optimized implementations for those primes. We report these values for NIST security Levels I, III and V in Table 2, together with the respective public key and signature sizes discussed in the next section.

5.2 Sizes

The public key for both PRISM-sig and PRISM-id is a curve E_{vk}. Since we are working in the Montgomery model, we can represent it with a single scalar in $\mathbb{F}_{p^2}$, with a cost of 4λ bits. A deterministic basis of the 2^a-torsion on E_{vk} can be included in the public key, to optimize verification performance, or computed on demand, to optimize compactness. This situation is completely analogous to SQIsign2D-West. As such, we follow their optimized approach of giving hints for the generation of the deterministic basis, with a cost of 2 bytes (see [6, Sec. 7]).

Signature Sizes for PRISM-sig. An isogeny representation consists of a curve E_{sig} and two points $(P_{\mathsf{sig}}, Q_{\mathsf{sig}})$. We proceed as follows: first, we send both coordinates of P_{sig}. Since $\log p = 2\lambda$ and we are working in $\mathbb{F}_{p^2}$, we can represent these coordinates with 8λ bits. Since we work with Montgomery curves, from P_{sig} we can recover the curve E_{sig} at the cost of a single field inversion. We can then represent Q_{sig} by its x-coordinate plus a bit to indicate the sign of the y-coordinate, which we can recover by computing only a square root. Ignoring the sign bit, this requires a total of 12λ bits to represent the isogeny.

An alternate approach would be to encode both P_{sig} and Q_{sig} by their coefficients with respect to a deterministic basis of the 2^a-torsion on E_{sig}. This is, for instance, the method adopted by SQIsign2D-West [6]. This approach would require, in our case, 4λ bits for the Montgomery coefficient of E_{sig} and $4a$ bits for the points. Since $a \approx 2\lambda$, the total cost would then again be $\approx 12\lambda$ bits. Recall from Equation (6) that this is in fact a slight overestimation and that $\approx 12\lambda - 4\log\lambda$ should do. However, encoding the points in this way would require the prover to compute a deterministic basis for the 2^a-torsion on E_{sig}. This is much more costly than computing a single square root, and we thus conclude that this alternate approach, as stated, is strictly worse than the first one.

If instead of four coefficients we send only three, and recover the last one using pairings with the method discussed in [20, §6.1], the signature would now have size $4\lambda + 3a$. However, this extra compactness comes at the additional cost of two pairing computations on top of the aforementioned deterministic basis. In this article, we prioritize efficiency of our scheme due to the relative compactness of isogeny-based signatures compared to other types of post-quantum signatures. We therefore choose to represent our signatures PRISM-sig as $(P_{\mathsf{sig}}, x(Q_{\mathsf{sig}}), \epsilon)$, where ϵ is a bit used to identify $y(Q_{\mathsf{sig}})$; thus, a signature has size 12λ. Nonetheless, we remark that the other approaches may become useful in contexts in which signature size is crucial.

Communication Costs for PRISM-id. The situation for PRISM-id is different. From Proposition 1 we only need to impose $\#\mathsf{Primes}_a \geq 2^\lambda$ to achieve λ bits security. For this we only need $a \approx \lambda + \log\lambda$, and this also implies hardness for Problem 2. This makes point compression more convenient. On top of the 4λ bits needed for the curve, it is enough to send four (or three, at the cost of pairing computations) coefficients of a bits each, for a total cost of $4\lambda + 4a \approx 8\lambda + 4\log\lambda$

(resp. $\approx 7\lambda + 3\log\lambda$). The compactness that we obtain in this way makes this method more favorable in the context of identification, and potentially for other future protocols relying on this as a building block. The data reported in Table 2 refers to the efficiency-oriented version using four point coefficients.

Notice that the first interaction from Vrf only requires $a - 1$ bits to represent the prime $q \in \mathsf{Primes}_a$, the total communication costs using point compression is then $4\lambda + 4a$.

Table 2. Public-key size and signature size in bytes, respectively communication cost, for the signature scheme PRISM-sig, respectively identification protocol PRISM-id.

	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
PRISM-id a	136	201	265
PRISM-sig a	248	376	500
PRISM-id com. cost	132	197	261
PRISM-sig sig. size	189	288	388

5.3 Comparison with other isogeny-based signatures

We now briefly compare the signature size of PRISM-sig with other isogeny-based signatures in Table 3. For each scheme we report the signature size of the main variant, ignoring logarithmic factors in terms of the security parameter λ. A more compact version of SQIsign2D-East presented in [41], CompactSQIsign2D-East, only requires 10λ bits. However, it comes with a significant computational overhead and an increase in the number of two-dimensional isogeny computations (up to about 770 for Level I). A trade-off similar to the one discussed above could bring SQIsign2D-West signature sizes from 9λ bits to 8λ bits [6, Remark 26]. However, as in our case, the authors prefer the faster version of the scheme. Finally, for SQIPrime we report the size of the signature in the two-dimensional case. In their paper [28], the authors also present a four-dimensional version, which achieves a signature size of 12λ, but seems less practical due to the need for four-dimensional isogenies for verification.

5.4 Performance

We now compare the performance of PRISM-sig with other isogeny based schemes.

In Table 4 are reported the number of isogeny computations in different dimensions performed by our scheme and various other SQIsign variants. With respect to this metric, we see that our scheme has the most efficient signing among

Protocol	**This Work**	SQIsign	SQIsign2D-East	SQIsign2D-West	SQIPrime
Sig. size (bits)	12λ	$\approx 11\lambda$	12λ	9λ	19λ

Table 3. Signature sizes for the signature scheme given in this work, SQIsign, and its most efficient variants.

all the analyzed variants. On the other hand, verification involves twice as many two-dimensional isogenies compared to both SQIsign2D-East and SQIsign2D-West, but no one-dimensional isogenies. Moreover, thanks to the trick mentioned in Section 5.2, we avoid computing a deterministic basis.

Table 4. Number of isogenies computed of each degree for NIST-I parameters. The numbers given in parentheses indicate that they may vary slightly depending on the case.

		Type of isogeny				
Protocol		2	3	5	$(2,2)$	$(2,2,2,2)$
This Work	KeyGen	-	-	-	496	-
	Sign	-	-	-	496	-
	Verify	-	-	-	248	-
SQIsignHD	KeyGen	378	234	-	-	-
	Sign	252	312	-	-	-
	Verify	-	78	-	-	142
SQIsign2D-West	KeyGen	-	-	-	496	-
	Sign	(248)	-	-	992	-
	Verify	(248)	-	-	(126)	-
SQIsign2D-West (Heuristic)	KeyGen	-	-	-	496	-
	Sign	(122)	-	-	624	-
	Verify	(122)	-	-	(126)	-
SQIsign2D-East	KeyGen	-	-	-	253	-
	Sign	127	(2)	(1)	641	-
	Verify	127	(2)	(1)	129	-

To validate this theoretical analysis, we implemented our signature scheme within the code base of SQIsign2D-West [1] and compare performance with SQIsign2D-West by running the two schemes on the same machine. The results are reported in Table 5.

The signature time showcased in Table 5 already proves that our signature time is better than SQIsign2D-West's. It is not as good as the estimates in Table 4 would suggest because of the performance of the LLL implementation (which is a necessary sub-routine of IdealToIsogeny). The computational cost of the current implementation seems to be increasing a lot with the volume of the lattice given as input, and the volume of this lattice is quite larger in our signature scheme than in SQIsign2D-West. As a result, the basis reduction step currently takes around 40% of the total signature time. We expect that this step can be optimized, thus leading to a signing time that would be more consistent with the operation estimates reported in Table 4.

Our verification is slower than SQIsign2D-West by a factor 1.4. This is consistent with our expectations.

Table 5. Run time comparison in millions of clockcycles between our signature scheme and SQIsign2D-West at NIST-I security, with optimized finite field arithmetic. Average run time over 100 iterations on an Intel Core i7 at 2.30 GHz with turbo-boost disabled.

SQIsign2D-West	KeyGen	77.4
	Sign	285.7
	Verify	11.9
This work	KeyGen	78.2
	Sign	157.6
	Verify	16.9

We conclude by briefly comparing the performance of PRISM-sig and PRISM-id. Key generation is exactly the same. In the signing phase of PRISM-sig we have to hash the message, while in PRISM-id we send the points as coefficients and hence we need to compute a deterministic basis. While both these steps have a minor impact on the respective protocols, hashing is faster than finding bases, so we expect PRISM-id to be slightly slower in this phase. On the other hand, the verification for PRISM-id is twice as fast: we only need $\lambda + \log(\lambda)$ two-dimensional $(2,2)$-isogenies (for a total of 135 for NIST Level I) instead of 2λ for PRISM-sig. Notice that 135 two-dimensional isogenies and no one-dimensional isogenies is less than the verification cost for all the signature schemes presented in Table 2. This again highlights an interesting property of our signature scheme: the efficiency bottleneck for PRISM-sig is the need to protect against hash collisions.

6 Conclusion

We present a novel two-round identification protocol and a hash-and-sign digital signature scheme, whose security reduces to the hardness of computing large prime degree isogenies from a curve with unknown endomorphism ring. The response

and the signature consist of an efficient higher-dimensional representation of such isogeny with domain the verification key E_{vk}.

We note that the determining factor for secure parameters is actually the size of the challenge space and not the hardness of the underlying isogeny problem. In particular, for the signature scheme, we need to increase the set Primes_a to protect against collision attacks on the hash function. To mitigate this, we carefully measure the cost of computing the hash function $\mathsf{H}_{\mathsf{prime}}$, which involves repetitions and primality testing, to correctly assess the hardness of collision finding.

We evaluate the performance of the presented schemes, showing that they compare favorably to the original SQIsign and many of its variants. Furthermore, since our construction has the additional advantage of having a much simpler signature procedure, we hope future work will prove its usefulness as a building block for more advanced constructions.

References

1. SQIsign2D-West code. https://github.com/SQISign/sqisign2d-west-ac24.
2. SQIsignHD code. https://github.com/Pierrick-Dartois/SQISignHD-lib.
3. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 418–433, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
4. François Arnault. Rabin-miller primality test: composite numbers which pass it. *mathematics of computation*, 64(209):355–361, 1995.
5. Andrea Basso. POKE: A framework for efficient PKEs, split KEMs, and OPRFs from higher-dimensional isogenies. Cryptology ePrint Archive, Paper 2024/624, 2024.
6. Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D-West: The fast, the small, and the safer. In *ASIACRYPT 2024*. Springer-Verlag, 2024.
7. Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. One tree to rule them all: Optimizing GGM trees and OWFs for post-quantum signatures. *Cryptology ePrint Archive*, 2024.
8. Daniel J Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020.
9. Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT 2015*, pages 368–397. Springer, 2015.
10. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.
11. Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *International Conference on Cryptology in India*, pages 428–442. Springer, 2014.

12. Dan Boneh and Victor Shoup. A graduate course in applied cryptography. *Draft 0.5*, 2020.
13. Giacomo Borin, Yi-Fu Lai, and Antonin Leroux. Erebor and Durian: Full anonymous ring signatures from quaternions and isogenies. Cryptology ePrint Archive, Paper 2024/1185, 2024. https://eprint.iacr.org/2024/1185.
14. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.
15. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *EUROCRYPT 2023*, volume 14008 of *LNCS*, pages 423–447. Springer, 2023.
16. Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQISign specification. https://sqisign.org/spec/sqisign-20230601.pdf, 2023. Accessed: 2023-10-04.
17. Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders. AprèsSQI: Extra fast verification for SQIsign using extension-field signing. Cryptology ePrint Archive, Paper 2023/1559, 2023. https://eprint.iacr.org/2023/1559.
18. Craig Costello. B-SIDH: Supersingular isogeny Diffie-Hellman using twisted torsion. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 440–463. Springer, Heidelberg, December 2020.
19. Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. Buffing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1696–1714. IEEE, 2021.
20. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: new dimensions in cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–32. Springer, 2024.
21. Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. An algorithmic approach to (2, 2)-isogenies in the theta model and applications to isogeny-based cryptography. 2023.
22. Luca De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 2017.
23. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, pages 64–93. Springer, 2020.
24. Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the Deuring correspondence: towards practical and secure SQISign signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 659–690. Springer, 2023.
25. Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78:425–440, 2016.
26. Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 14, pages 197–272. Springer Berlin/Heidelberg, 1941.

27. Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.
28. Max Duparc and Tako Boris Fouotsa. SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In *ASIACRYPT 2024*. Springer-Verlag, 2024.
29. Noam D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational perspectives on number theory, Studies in Advanced Mathematics 7*, pages 21–76. AMS, 1998.
30. Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni. Deuring for the people: Supersingular elliptic curves with prescribed endomorphism ring in general characteristic. *IACR Cryptol. ePrint Arch.*, 2023:106, 2023.
31. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.
32. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5):1–75, 2018.
33. Steven D Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, 2012.
34. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.
35. Ernst Kani. The number of curves of genus two with elliptic differentials. *Journal fur die reine und angewandte Mathematik*, 1997.
36. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.
37. David Russell Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
38. Antonin Leroux. Verifiable random function from the Deuring correspondence and higher dimensional isogenies. *Cryptology ePrint Archive*, 2023.
39. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *EUROCRYPT 2024, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, 2023.
40. Kohei Nakagawa and Hiroshi Onuki. QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In *CRYPTO 2024, Part V*, volume 14924 of *LNCS*, pages 75–106. Springer, 2024.
41. Kohei Nakagawa, Hiroshi Onuki, Wouter Castryck, Mingjie Chen, Riccardo Invernizzi, Gioella Lorenzon, and Frederik Vercauteren. SQIsign2D-East: A new signature scheme using 2-dimensional isogenies. In *ASIACRYPT 2024*. Springer-Verlag, 2024.
42. National Institute of Standards and Technology (NIST). Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process, 2022.

43. Aurel Page and Damien Robert. Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. *Cryptology ePrint Archive*, 2023.
44. Farzin Renan and Péter Kutas. SQIAsignHD: SQIsignHD adaptor signature. Cryptology ePrint Archive, Paper 2024/561, 2024.
45. Damien Robert. Breaking SIDH in polynomial time. In *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, 2023.
46. Damien Robert. On the efficient representation of isogenies. Cryptology ePrint Archive, Paper 2024/1071, 2024. https://eprint.iacr.org/2024/1071.
47. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.
48. Douglas R Stinson. *Cryptography: theory and practice*. Chapman and Hall/CRC, 2005.
49. Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12:1–28, 1999.
50. Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, July 1971.
51. John Voight. *Quaternion algebras*. Springer Nature, 2021.

A Additional Preliminaries

A.1 Different Higher Dimensional Variants of SQIsign

Recently, different variants of SQIsign that use higher-dimensional isogenies have been proposed. They are all inspired by the protocol from Section 2.5, but realize it in different ways. We summarize them here.

SQIsignHD [20]. There are two different versions of this protocol. The first version, called RigorousSQIsignHD, aims for the best possible provable security using eight-dimensional isogenies, which makes it unpractical. The second, FastSQIsignHD, restricts to four-dimensional isogenies to build a scheme with smaller signatures than SQIsign, similarly sized public keys, and significantly faster signing time, but slower verification.

SQIsign2D-West [6]. This variant is based on the IdealToIsogeny algorithm introduced in Section 2.3 to overcome the obstacles encountered in SQIsignHD and go down to dimension 2. It has a security equivalent to RigorousSQIsignHD while having the fastest verification procedure. Signature size is comparable to SQIsignHD, and the signing time is slower than SQIsignHD but faster than SQIsign. The authors of [6] also propose a heuristic variant, with a less rigorous security proof and better signing time (still slower than SQIsignHD).

SQIsign2D-East [41]. Avoids the need for four-dimensional isogenies using endomorphisms in the *Eichler order* $\mathcal{O}_{\mathsf{vk}} \cap \mathcal{O}_0$. Because of this, the security proof requires some ad hoc heuristic. Signing and verification time should be comparable to the signing time of the heuristic version of SQIsign2D-West and the signature size is slightly bigger.

SQIPrime [28]. Uses techniques inspired by [38] to build non-smooth challenge isogenies. While conceptually this seems the closest variant to our proposal, we remark that in SQIPrime the non-smooth isogenies are computed for a fixed prime degree q. Their method is thus not applicable to our setting.

A.2 Isogenies of Fixed Degree

An application of Theorem 1 is the following: given an isogeny $\varphi : E_0 \to E_3$ of degree $u(2^a - u)$, for an odd number u, we can factor it as $\varphi := \varphi_{u'} \circ \varphi_u = \psi_u \circ \psi_{u'}$ with $\deg(\varphi_u) = \deg(\psi_u) = u$ and $\deg(\varphi_{u'}) = \deg(\psi_{u'}) = 2^a - u$. Notice that as long as u is odd $\gcd(u, 2^a - u) = 1$. We have the following square:

$$\begin{array}{ccc} E_0 & \xrightarrow{\varphi_u} & E_1 \\ \downarrow{\scriptstyle \psi_{u'}} & & \downarrow{\scriptstyle \varphi_{u'}} \\ E_2 & \xrightarrow{\psi_u} & E_3 \end{array}$$

Then by Kani's lemma we get

$$\Phi = \begin{pmatrix} \varphi_u & \widehat{\varphi}_{u'} \\ -\psi_{u'} & \widehat{\psi}_u \end{pmatrix} : E_0 \times E_3 \to E_1 \times E_2$$

with kernel

$$\begin{aligned} \ker(\Phi) &= \{(\widehat{\varphi}_u(P), \varphi_{u'}(P)) | P \in E_1[2^a]\} \\ &= \{(\widehat{\varphi}_u(\varphi_u(P)), \varphi_{u'}(\varphi_u(P))) | P \in E_0[2^a]\} \\ &= \{(uP, \varphi(P)) | P \in E_0[2^a]\}. \end{aligned}$$

If we know the images under φ of the 2^a-torsion points of E_0 then we can evaluate Φ. We can thus recover φ_u and $\varphi_{u'}$ as components of Φ; for instance

$$\Phi((R, 0)) = (\varphi_u(R), -\psi_{u'}(R)).$$

Since the kernel defines an isogeny up to isomorphism, the components of Φ could be swapped, e.g. the codomain can be $E_2 \times E_1$ instead of $E_1 \times E_2$. In this case we can recover the correct component using pairings. If we know that the ideal associated to φ is I, the ideal corresponding to φ_u of degree u is $I + u\mathcal{O}$ (see for instance [6, Lemma 6]).

This method has been exploited in QFESTA [40] to compute isogenies of a given degree q starting from E_0. The idea is to find an endomorphism on E_0 of degree $q(2^a - q)$, where a is such that we have access to the 2^a torsion. This is possible thanks to the knowledge of the endomorphism ring of E_0, using for instance [23, Alg. 1]. This endomorphism can be evaluated and then factored in two components: one of degree q and one of degree $2^a - q$. This trick is a key subroutine of the IdealToIsogeny algorithm.

Our techniques for finding an endomorphism in $\mathcal{O}_0$ of given norm require the latter to be greater than p. The case where $q(2^a - q) < p$ can be solved

by using, if available, additional B-torsion for odd B and searching instead an endomorphism of norm $q(2^a - q)B$. The endomorphism can be factored as $\theta = \widehat{\tau} \circ \varphi$ with τ of degree B and φ of degree $q(2^a - q)$, then from the action of θ on the B-torsion we can recover the kernel of τ and recover $\varphi(P)$ as $[1/B]\tau(\theta(P))$ for any point P of order coprime with B.

B Alternative Subroutines Using Odd-degree Secret Isogeny

We now introduce an alternative method to instantiate the key generation and isogeny generation algorithms. For this variant we use a prime such that we have access (over a small extension field of $\mathbb{F}_{p^2}$) to both the 2^a-torsion and the B-torsion, where B is an odd smooth integer. A natural option to achieve this is to choose p to be an SIDH-like prime of the form $p = 2^a Bf - 1$, with $B = 3^b$. In this way, the $2^a B$-torsion is defined over $\mathbb{F}_{p^2}$. We define new KeyGen and GenIsogeny procedures, that are compatible with VerIsogeny from the previous Section 3.3.

Alternative Key Generation. Differently from Section 3.3 sample a uniformly random cyclic order-B subgroup K_{sk} of $E_0[B]$ and compute the associated isogeny $\phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$. Note that, to be secure against meet-in-the-middle key recoveries, we need to have $B \approx 2^{2\lambda}$. Define the secret key as $\mathsf{sk} = (\phi_{\mathsf{sk}}, K_{\mathsf{sk}})$ and the public key as $\mathsf{vk} = (E_{\mathsf{vk}}, P_{\mathsf{vk}}, Q_{\mathsf{vk}})$, where $P_{\mathsf{vk}}, Q_{\mathsf{vk}}$ are deterministically computed generators of $E_{\mathsf{vk}}[2^a]$. As before, points could be optional depending on what one wants to minimize.

Alternative Isogeny Generation. Similarly to the definition of GenIsogeny using IdealToIsogeny in Section 3.3, we want to efficiently represent an isogeny $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$. Differently from Section 3.3, we divide the signature procedure in two phases: in phase one, we generate an isogeny ζ of non-smooth degree $q(2^a - q)$ with domain E_0; in phase two, we push the isogeny representation through ϕ_{sk} to get the signature $\sigma = [\phi_{\mathsf{sk}}]_* \zeta : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$.

Non-smooth degree isogeny generation. We start by computing an isogeny of degree $q(2^a - q)$ with domain E_0 using a small modification of the *QFESTA algorithm* [40] introduced in [5, Algorithm 2]:

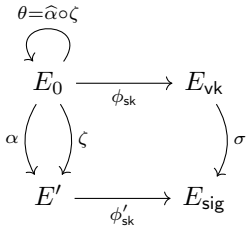
1. Compute a cyclic endomorphism θ of degree $q(2^a - q)B$, which is larger than p, using the RepresentInteger procedure from SQIsign [23,24]. The endomorphism can be factored as a composition of two isogenies ζ and $\widehat{\alpha}$ of degrees $q(2^a - q)$ and B, respectively, such that $\theta = \widehat{\alpha} \circ \zeta$;
2. Find a point R of order B such that $\widehat{\theta}(R) = \mathcal{O}_{E_0}$. Since $\widehat{\theta} = \widehat{\zeta} \circ \alpha$ is cyclic and $\deg(\alpha) = B$, this means that $\ker(\alpha) = \langle R \rangle$, from which α can be computed using Vélu's formulae.

3. For any point $P \in E_0$ of order coprime to B, we can compute $\zeta(P)$ as

$$\zeta(P) = [1/B]\,\alpha \circ \widehat{\alpha} \circ \zeta(P) = [1/B]\,\alpha \circ \theta(P).$$

From this, we can get an efficient higher-dimensional representation by computing the image of a 2^a-torsion basis, as described in Appendix A.2.

Push-forward of ζ. Given an efficient representation of ζ, we can use the diagram in Equation (7) to get the signature isogeny $\sigma = [\phi_{\mathsf{sk}}]_*\zeta$. Note that the right-most square of the diagram is commutative, i.e., $\sigma \circ \phi_{\mathsf{sk}} = \phi'_{\mathsf{sk}} \circ \zeta$. First, note that $\phi'_{\mathsf{sk}} : E' \to E_{\mathsf{sig}}$ can be efficiently computed from the pushed kernel generator $\zeta(K_{\mathsf{sk}})$. Then, given a 2^a-torsion basis (P_2, Q_2) on E_0 such that $\phi_{\mathsf{sk}}(P_2) = P_{\mathsf{vk}}$ and $\phi_{\mathsf{sk}}(Q_2) = Q_{\mathsf{vk}}$,[8] we retrieve the two-dimensional representation from $\sigma(P_{\mathsf{vk}}) = \phi'_{\mathsf{sk}} \circ \zeta(P_2)$ and $\sigma(Q_{\mathsf{vk}}) = \phi'_{\mathsf{sk}} \circ \zeta(Q_2)$.



(7)

Comparison with Section 3.3. This alternative GenIsogeny subroutine only requires the computation of a $(2,2)$-isogenies and two one-dimensional isogenies of degree B, instead of a longer chain of $(2,2)$-isogenies of length $2a$. This could potentially lead to a performance improvement. However, preliminary experiments with our SageMath implementation still suggest that this alternative variant of GenIsogeny involving odd degree isogenies is around $\times 1.25$ slower than the one using IdealToIsogeny. This is primarily because in this variant the underlying prime is larger in order to have sufficient rational 2^aB-torsion (rather than just rational 2^a-torsion). This increased prime size also affects the verification time: it is around $\times 2$ slower. For these reasons we focus on the construction of our schemes given in Section 3.3.

Remark 7. We may decrease the odd torsion B by splitting the secret isogeny in k pieces and performing k push-forwards, in this way we only need $B^k \approx 2^{2\lambda}$. We can also use a smooth torsion over bigger extension fields $E_0/\mathbb{F}_{p^{2k}}$, like in B-SIDH [18]. These options reduce the size of p at the cost of increased computation needed in signing.

[8] This basis can be precomputed during key generation.