Isogeny-based Signatures with Randomizable Keys

Andrea Basso[1], Giacomo Borin[1,2], Maria Corte-Real Santos[3], Pierrick Dartois[4], Riccardo Invernizzi[5], Luciano Maino[6], Robi Pedersen[7], and Michel Seck[8]

[1] IBM Research Europe, Switzerland
[2] University of Zurich, Switzerland
swrk@gbor.in
[3] ENS Lyon and CNRS
maria.corte_real_santos@ens-lyon.fr
[4] Univ. Rennes, Inria, CNRS, IRISA, UMR 6074, F-35000, Rennes, France
pierrick dot dartois at inria dot fr
[5] COSIC - KU Leuven, Belgium
riccardo.invernizzi@kuleuven.be
[6] University of Birmingham, United Kingdom
l.maino@bham.ac.uk
[7] Technical University of Denmark
robi.pedersen@protonmail.com
[8] Ecole Polytechnique de Thies, LTISI, Senegal
mseck@ept.edu.sn

Abstract. Digital signature schemes based on isogenies are among the most compact signatures achieving post-quantum security. Recent advances, especially those leveraging higher-dimensional isogenies, have also made such schemes practically efficient. However, comparatively little attention has been devoted to endowing these signatures with additional privacy-enhancing properties, such as the re-randomization of keys and the adaptation of signatures to new public keys. Although some results exist in the isogeny group action setting, these signatures suffer from a subexponential quantum attack which renders them rather inefficient.

In this work, we initiate the first systematic study of privacy-enhancing isogeny-based signatures outside the group-action framework. We base our exploration on the notion of *signatures with randomizable keys* (SWRKs) developed by Celi et al. (FC'24), which aims to unify privacy notions related to key updatability and signature adaptation. In particular, we analyze which of their privacy notions can be achieved from the state-of-the-art signatures SQIsign, PRISM and the hash-and-sign signature scheme derived from the Deuring verifiable unpredictable function (DeuringVUF). To this end, we naturally extend SQIsign to an SWRK scheme that allows key randomization, and enhance both PRISM and the DeuringVUF signature to additionally allow for message adaptation. We show that, due to the deterministic nature of the signatures, the DeuringVUF signature achieves perfect adaptability. We formally prove all three of our modifications achieve unlinkability against unbounded adversaries, and remain unforgeable under the same assumptions as the original schemes.

1 Introduction

Digital signatures are fundamental cryptographic primitives that enable authentication and ensure message integrity. Beyond their basic functionality, a large range of advanced features have been developed in the last decades, especially with uses in distributed trust and privacy-preserving applications. In this paper, we focus on the generalized framework of signatures with randomizable keys (SWRK) developed in [17].

The framework of signatures with randomizable keys [17] captures several constructions in the literature, such as malleable signatures [18] (with fixed messages), randomizable signatures [15,14,50], and signatures with re-randomizable keys [39]. While such constructions differ in their specific functionalities, they all consist of signature protocols with additional tools to randomize the public keys, secret keys, and signatures, or a subset of them. Despite the applications of these primitives in various privacy-preserving constructions, for example Signal's quantum-safe private group system [22], anonymity networks (Tor) [58,36], anonymous credentials [23,24], the Privacy Pass protocol [28], and deterministic wallets [27,54], the study of post-quantum signatures with randomizable keys still lags behind the study of other advanced functionalities.

Isogeny-based cryptography is a promising candidate to construct quantum-safe versions of signatures with randomizable keys. This is demonstrated by the several isogeny-based SWRK schemes that have already been proposed in the literature [55,26,56,41], which are all based on the framework of group actions. This framework restricts itself to a subset of possible isogenies to obtain a building block that closely resembles exponentiation in a group setting. Due to this similarity, the group-action framework is well-suited to translate many Diffie-Hellman-based constructions to the post-quantum setting. This flexibility, however, comes at a price: isogeny-based group actions are susceptible to a sub-exponential quantum attack, originally due to Kuperberg [11,49]. While the concrete complexity of this attack is still under debate, for NIST Level I security, the most conservative choice is to use 4096-bit primes [19]. This severely impacts the performance of group-action-based schemes.

To seek better performance, isogeny-based cryptography is moving towards settings in which this quantum subexponential attack does not seem to apply. By exploiting the Deuring correspondence, together with new higher-dimensional tools, a new wave of compact post-quantum signatures have been developed, such as SQIsign [1], PRISM [5] and the hash-and-sign signature based on DeuringVUF [44]. For NIST Level I security, these signatures use 256-bit primes, almost 20 times smaller than the group-action setting, making these the most compact and also efficient isogeny-based signatures in the literature.[9]

[9] Another possible candidate could be the SPRINT signature scheme [33], which relies on SNARK proof systems for isogeny statements, resulting in efficient, but rather large signatures. Although we believe that SPRINT can potentially be turned into a SWRK, the underlying machinery is very different from the signatures based on the Deuring correspondence explored in this work. We therefore leave this exploration as an interesting future research direction.

Compared to the group-action-based schemes, however, these signatures provide less flexibility, making advanced functionalities in general harder to build. Exacerbating this, the lack of overarching frameworks often results in advanced constructions that tend to be somewhat ad-hoc and largely scheme-dependent.

Contributions. In this work, we address these issues and present three novel isogeny-based signatures with randomizable keys. Our constructions build upon SQIsign [1], PRISM [5] and the signature obtained from the DeuringVUF [44, §5.2]. We present our constructions in a unified framework, by abstracting away the heavy machinery of the Deuring correspondence, and relying on a set of high-level algorithms that are shared among the three signatures. Concretely, our contributions are the following.

1. We naturally extend SQIsign to allow for key randomization by endowing it with algorithms to both randomize the public and secret keys. We show that the randomizable version, SQIsign-RK, achieves unforgeability under the same assumptions and parameters as the original scheme.
2. We also extend PRISM and the DeuringVUF signature to allow for key randomization using techniques similar to those for SQIsign-RK, while also enabling message adaptation to the randomized keys. As a stepping stone towards the latter, we propose a new algorithm that allows us to push forward higher-dimensional isogeny representations, encoded via torsion points of order 2^a, through smooth isogenies of degree a power of 2. Both of our schemes achieve unforgeability under the same assumptions as the original schemes. Under appropriate parameter choices, both signatures achieve the *signature adaptation* property, implying that fresh signatures on a randomized public key are indistinguishable from signatures that are pushed forward from a previous public key. Due to the deterministic nature of the DeuringVUF signatures, the latter even achieves the notion of *perfect* adaptation.
3. We further explore the *unlinkability* property of our three constructions, which relates to the hardness of linking randomized public keys to long-term public keys of a signer. We show that under appropriate parameters, all three of our schemes achieve unlinkability against unbounded adversaries. To do this in the DeuringVUF case, we give a formal proof for the mixing-time of random non-backtracking walks on the ℓ-isogeny graph with full level structure, which may be of independent interest. As a consequence, all three of our constructions find applications in (delegatable) anonymous credentials [23,24], the Privacy Pass protocol [28] and deterministic wallets with stealth addresses [51].
4. We provide a SageMath proof-of-concept implementations of the SQIsign, PRISM and DeuringVUF key randomization algorithms along with the PRISM and DeuringVUF signature adaptation algorithm and analyze their relative costs with respect to the original signature schemes. To allow for the implementation of the adaptation procedure for DeuringVUF, we select slightly different parameters than those proposed in [44].

1.1 Technical overview

In isogeny-based cryptography, secret information is often encoded as a "long" isogeny $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}$, where E_0 is a specific curve. The codomain curve, the curve E_{pk}, is then used as public information. Given the curve E_{pk}, recovering the isogeny φ_{sk} is believed to be a hard problem.

This basic concept allows us to randomize public information, a key building block that can be leveraged to design signatures with randomizable keys. Given E_{pk}, we compute an isogeny $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$ to obtain a new public key, the curve E'_{pk}. The secret information corresponding to E'_{pk} is another isogeny path from E_0, which can be derived, for instance, by considering $\phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$.

A desirable property is to ensure that the new public curve E'_{pk} "looks like something truly random". This intuitive concept can be mathematically formalized by looking at the *ℓ-isogeny graph*: a graph whose vertices represent (supersingular) elliptic curves and whose edges describe isogenies of small prime degree ℓ. This graph possesses the *Ramanujan property*, which means, informally, that random walks from any vertex (curve) on the graph rapidly converge to the stationary distribution over the graph. More precisely, such a property is captured by the following theorem.

Theorem 1 (Special case of [7, Thm. 11]). *Let ℓ, p be two distinct primes, k a positive integer and π be any probability distribution on the ℓ-isogeny graph. Then:*

- *π_k, the probability distribution obtained after applying a random non-backtracking isogeny of degree ℓ^k to π;*
- *s, the stationary distribution on the graph,*

have total variation distance at most

$$d_{TV}(\pi_k, s) \leq \frac{1}{4}\sqrt{\frac{p-1}{\ell^k}} \cdot \frac{(\ell+1)(k+1)-2}{(\ell+1)} \tag{1}$$

Randomizing elliptic curves, however, may not be sufficient. We need to ensure that the new public curve admits a secret key that is compatible with the rest of the construction. Consider the original version of SQIsign [20,32]. In such a protocol, public keys are not uniformly distributed among all supersingular curves. Hence, randomizing E_{pk} would not lead to an admissible secret key. Fortunately, this limitation has been overcome by the updated version of SQIsign [1]. In this case, we show that public keys can be fully anonymized, even against unbounded adversaries.

Similar public-key randomization techniques can also be applied to two other isogeny-based digital signature schemes, namely PRISM and the DeuringVUF-based signature. Differently from SQIsign, using PRISM and the DeuringVUF-based signature, we also design an adaptation mechanism that allows us to adjust signatures to new public keys. The main idea behind this signature adaptation is to use pushforwards of isogenies.

Designing adaptation mechanisms requires some additional analysis, especially in the case of DeuringVUF. Namely, we have to ensure that the information coming from pushforwards of isogenies does not accidentally leak data about the randomization isogeny. This problem can be handled by looking at isogenies between elliptic curves with level structure.

Equipped with these tools to randomize public keys and adapt signatures, we prove that our schemes enjoy several advanced functionalities under the hardness of some well-established problems in isogeny-based cryptography. The main insight in proving such properties is that the randomization operation is compatible with pushforwards and pullbacks of isogenies. This allows us to utilize a proof strategy similar to the random self-reducibility of the discrete logarithm problem, a key property for the security and constructions of many classical schemes. This approach was already known to work effectively for group actions, and we extend it here to non-oriented curves as well.

We summarize our contributions and the security properties of our constructions in Tab. 1. We stress that all our constructions remain unforgeable under the same assumptions as the underlying signature scheme, and provide unconditional unlinkability of long-term public keys, as long as the degree of the isogeny used for randomization is large enough.

Table 1: Summary of security properties of our constructions. Here ϕ_{rr} denotes the secret isogeny used for key randomization.

	$\log_2(\deg(\phi_{\mathsf{rr}}))$	Unlinkability	Adaptation	Perfect Adapt.
SQIsign-RK	$\log_2(p) + 2\lambda$	✓	-	-
PRISM-RK	$\log_2(p) + 2\lambda$	✓	✓	-
DeuringVUF-RK	$\log_2(p) + 8\lambda$	✓	✓	✓

Notation. We use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We also use the notation $(\_, \_)$ and $[\_, \_]$ for open and closed intervals, respectively. The security parameter is λ, and $\mathsf{negl}(\lambda)$ denotes any negligible function in λ. We label the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$ as Ell_p. Given $E \in \mathsf{Ell}_p$, we denote the set of separable isogenies with domain E and degree d as $\mathsf{Isog}(E, d)$. We use the notation $a \xleftarrow{\$} A$ to imply that a is sampled uniformly at random from the set A. If instead we use $\leftarrow$, we mean the output of a (probabilistic) algorithm. We say that an algorithm $\mathcal{A}$ runs in polynomial time if it runs in polynomial time in the size of its inputs. We write $t_{\mathcal{A}}$ to denote the running time of $\mathcal{A}$. We also say that its output is of polynomial size if the output size is polynomial in the input size. If $|\varepsilon - \varepsilon'| \leq \mathsf{negl}(\lambda)$, we write $\varepsilon \stackrel{\cdot}{\approx} \varepsilon'$. Given a prime N, we write $\mathbb{Z}/N\mathbb{Z}$ to denote the ring of integers modulo N, $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ to denote the group of invertible 2×2 matrices with entries in $\mathbb{Z}/N\mathbb{Z}$, and $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ to denote the subgroup of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ of matrices with determinant 1.

2 Preliminaries

In this section, we recall some background knowledge about isogeny-based cryptography. We then state the core definitions and security properties for signatures with randomizable keys, which we will use in the rest of the paper.

2.1 Isogenies

We assume some familiarity with elliptic curves and their isogenies, and refer the reader to [29,57] for more details. From this point onwards p is a prime of the form $p = c \cdot 2^f - 1$, where $c > 0$ is a small odd integer. In particular, we have $p \equiv 3 \bmod 4$. We denote by E_0 the supersingular elliptic curve $y^2 = x^3 + x$ of j-invariant 1728 defined over $\mathbb{F}_p$. All elliptic curves we consider will be supersingular. Furthermore, we recall that a separable isogeny $\varphi : E \to E'$ is uniquely defined by its kernel $\ker(\varphi)$ and can be efficiently computed (from its kernel) as long as $\ker(\varphi)$ has smooth order and is contained in a small extension field of the field of definition of E; the degree of φ, denoted by $\deg(\varphi)$, coincides with the cardinality of $\ker(\varphi)$.

Isogenies can be represented in several different ways. Of particular interest are the so-called *efficient representations*.

Definition 1. *Let $\mathscr{A}$ be an algorithm. An* efficient representation *of an isogeny $\varphi : E \longrightarrow E'$ between two elliptic curves defined over $\mathbb{F}_q$ (with respect to $\mathscr{A}$) is some data $D_\varphi \in \{0,1\}^*$ such that:*

(i) $\mathscr{A}(\mathsf{domain}, D_\varphi)$, $\mathscr{A}(\mathsf{codomain}, D_\varphi)$ and $\mathscr{A}(\mathsf{degree}, D_\varphi)$ respectively return E, E' and $\deg(\varphi)$ in polynomial time.
(ii) For all $P \in E$, $\mathscr{A}(P, D_\varphi)$ returns $\varphi(P)$ in polynomial time.

In practice, isogenies of smooth degrees (*e.g.*, a power of 2) can be efficiently represented as a chain of smaller degree isogenies, or equivalently by their kernel (assuming it is defined over a small field extension over the base field). By Kani's lemma [42], isogenies of non-smooth degrees can be efficiently represented as components of higher-dimensional isogenies of smooth degrees. Equivalently, these isogenies can be efficiently represented by torsion point images which yield their higher-dimensional representation. More specifically, in our context, we will use 2D representations.

Definition 2. *Let $\varphi : E_1 \to E_2$ be an isogeny of degree $q(2^n - q)$, for some odd $q > 0$. A* two-dimensional (2D) representation *of $\varphi : E_1 \to E_2$ is given by the tuple $(E_1, P_1, Q_1, E_2, \varphi(P_1), \varphi(Q_1), q, n)$, where (P_1, Q_1) is a basis of $E_1[2^n]$. This can also be seen as a 2D representation of a factor of φ of degree q.*

Given two isogenies $\varphi_1 : E \to E_1$ and $\varphi_2 : E \to E_2$ of coprime degree, 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 $\ker([\varphi_1]_*\varphi_2) = \varphi_1(\ker(\varphi_2))$.

Another important tool is the *Deuring correspondence*. Define $\mathcal{B}_{p,\infty}$ to be the quaternion algebra ramified at p and ∞, *i.e.*, $\mathcal{B}_{p,\infty} = \mathbb{Q}\langle i, j\rangle$, where $i^2 = -1$,

$j^2 = -p$ and $ij = -ji$. The Deuring correspondence is a categorical equivalence between maximal orders in $\mathcal{B}_{p,\infty}$ and supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ [34]. Under this correspondence, to each supersingular elliptic curve E over $\mathbb{F}_{p^2}$ we can associate a maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$ such that $\mathrm{End}(E) \cong \mathcal{O}$. An isogeny $\varphi : E_1 \to E_2$ corresponds to an ideal I_φ, and vice versa when the ideal is compatible with $\mathrm{End}(E_1)$.

In what follows, when writing $\mathrm{End}(E)$, we will refer to a description of the endomorphism ring of E: following [48], a description of $\mathrm{End}(E)$ consists of efficient representations of four endomorphisms generating it as a lattice, or equivalently, an embedding of $\mathrm{End}(E)$ into $\mathcal{B}_{p,\infty}$ [38, Lems. 4, 5]. Concretely, in our applications, we will use the following way to represent the endomorphism ring. When $p \equiv 3 \mod 4$, the elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_p$ has a known endomorphism ring $\mathcal{O}_0 \simeq \mathrm{End}(E_0)$ which has a very special form and convenient arithmetic properties.[10] If E is a supersingular elliptic curve, its endomorphism ring can be represented as an isogeny $\varphi : E_0 \to E$ along with its kernel ideal I_φ. Indeed $\mathrm{End}(E)$ can be inferred in polynomial-time from this data [38, Alg. 4].

If the endomorphism ring of the elliptic curves involved is known, the Deuring correspondence acts as a trapdoor: many hard isogeny computations become efficient. Hence, a representation of the endomorphism ring is often given as input to the main algorithmic building blocks of SQIsign, PRISM and DeuringVUF that we will describe in Sec. 3.1 and then use in the rest of the paper.

2.2 Signatures with randomizable keys

We revisit the definition of signature schemes with randomizable keys (SWRK) from [17]. An algorithmic definition is given in Def. 3. We refer to [17, §4] for a detailed discussion and systematization for the security properties of SWRKs.

Definition 3 (Signatures with Randomizable Keys (SWRK)). *A signature scheme with randomizable keys (SWRK) is a tuple of polynomial-time algorithms* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{RandPK}, \mathsf{RandSK}, \mathsf{VerKey})$*, defined as follows:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$*: a probabilistic algorithm that takes as input the security parameter and outputs the secret key* sk *and the public key* pk*.*
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{msg}) \to \sigma$*: a probabilistic algorithm that takes as input the secret key* sk *and a message* msg *and outputs a signature* σ*. The secret key* sk *can be obtained either from* KeyGen *or from* RandSK*.*
- $\mathsf{Verify}(\mathsf{pk}, \mathsf{msg}, \sigma) \to 0/1$*: a deterministic algorithm that takes as input the public key* pk*, a message* msg *and a signature* σ *and outputs* 1 *if* σ *is valid, else it outputs* 0*.*
- $\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}) \to \mathsf{pk}'$*: a deterministic algorithm that takes as input a public key* pk*, the randomness* rr*, and outputs a new public key* pk'*.*

[10] For instance, norm equations are easier to solve in $\mathcal{O}_0$ than in any maximal order of $\mathcal{B}_{p,\infty}$.

– $\mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr}) \to \mathsf{sk}'$*: a deterministic algorithm that takes as input a secret key* sk*, the corresponding public key* pk*, the randomness* rr*, and outputs a new secret key* sk'*.*[11]
– $\mathsf{VerKey}(\mathsf{pk}', \mathsf{sk}') \to 0/1$*: a deterministic algorithm that takes as input a public key* pk' *and a secret key* sk' *and outputs* 1 *if* pk' *is a valid public key for* sk'*, else it outputs* 0*.*

*An adaptable signature scheme with randomizable keys (*aSWRK*) is an* SWRK *with an additional algorithm* Adapt*, defined as follows:*

– $\mathsf{Adapt}(\mathsf{msg}, \sigma, \mathsf{rr}, \mathsf{pk}) \to \sigma'$*: a deterministic algorithm that takes as input a message* msg*, a signature* σ*, the randomness* rr*, and a public key* pk*, and outputs a new signature* σ' *for the public key* $\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr})$*.*

In the literature, SWRK schemes are also sometimes referred to as signature schemes with key-blinding [37,35]. Following [17], the main security properties for SWRK schemes are correctness and unforgeability. Furthermore, an aSWRK scheme needs to satisfy the signature adaptation property.

Definition 4 (Correctness). *An* SWRK *scheme is* correct *if the tuple* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is a correct signature scheme,* VerKey *succeeds only for valid key pairs, and if the randomization process generates valid key pairs, i.e., for all* rr *it holds that*

$$\mathsf{VerKey}(\mathsf{pk}, \mathsf{sk}) = \mathsf{VerKey}(\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}), \mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr}))\,.$$

Further, for an aSWRK*, the adaptation process generates valid signatures, i.e., for all* $\mathsf{msg}, \sigma, \mathsf{rr}, \mathsf{pk}$ *it holds that if* $\sigma' \leftarrow \mathsf{Adapt}(\mathsf{msg}, \sigma, \mathsf{rr}, \mathsf{pk})$*, then*

$$\mathsf{Verify}(\mathsf{pk}, \mathsf{msg}, \sigma) = \mathsf{Verify}(\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}), \mathsf{msg}, \sigma')\,. \tag{2}$$

The unforgeability property is similar to the classical one for digital signatures, but with the additional possibility that the adversary can query signatures on randomizable keys and return a signature over a randomized key of its choice. There are two possible cases depending on whether the randomness rr is honestly generated or not. We focus on the latter case, called 1-*unforgeability* in [17, Def. 15], in which there are no restrictions on the randomness.

Definition 5 (Signature Unforgeability). *An* SWRK *scheme satisfies the* signature unforgeability *property if for any PPT adversary* $\mathcal{A}$ *playing the game* G^{rkuf} *(Fig. 1), the winning probability* $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{rkuf}}(\lambda) = \Pr\left[G^{\mathsf{rkuf}}(\mathcal{A}) = 1\right]$ *is negligible in* λ*.*

[11] In the definition given in [17], RandSK does not take pk as input. However, given sk, we can always reconstruct pk. We include this to simplify the presentation of our constructions.

Game: $G^{\mathsf{rkuf}}(\mathcal{A})$:

1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$; $\mathcal{M} \leftarrow \emptyset$; $(\mathsf{msg}^*, \sigma^*, \mathsf{rr}^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'(\bullet,\bullet)}(\mathsf{pk})$
2: **if** $\mathsf{rr}^* = \perp$ **then return** $\mathsf{Verify}(\mathsf{pk}, \mathsf{msg}^*, \sigma^*) = 1$ and $\mathsf{msg}^* \notin \mathcal{M}$
3: **else return** $\mathsf{Verify}(\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}^*), \mathsf{msg}^*, \sigma^*) = 1$ and $\mathsf{msg}^* \notin \mathcal{M}$

$\mathsf{Sign}'(\mathsf{msg}, \mathsf{rr})$:

1: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathsf{msg}\}$
2: **if** $\mathsf{rr} = \perp$ **then return** $\mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$
3: **else return** $\mathsf{Sign}(\mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr}), \mathsf{msg})$

Games: G^{adpt}, $\boxed{G^{\mathsf{padpt}}}$

1: $b \xleftarrow{\$} \{0,1\}$; $\mathsf{rr} \xleftarrow{\$} \{0,1\}^*$; $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$;
2: $b' \leftarrow \mathcal{A}^{\mathcal{O}^b_{\mathsf{adapt}}}(\lambda, \mathsf{pk})$; **return** $b = b'$.

$\mathcal{O}^b_{\mathsf{adapt}}(\mathsf{msg})$:

1: $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$;
2: **if** $\mathrm{b} = 0$ **then** $\mathsf{pk}' \leftarrow \mathsf{RandPK}(\mathsf{pk}, \mathsf{rr})$; $\sigma' \leftarrow \mathsf{Adapt}(\mathsf{msg}, \sigma, \mathsf{rr}, \mathsf{pk})$;
3: **else** $\mathsf{pk}' \leftarrow \mathsf{RandPK}(\mathsf{pk}, \mathsf{rr})$; $\mathsf{sk}' \leftarrow \mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr})$; $\sigma' \leftarrow \mathsf{Sign}(\mathsf{sk}', \mathsf{msg})$;
4: **return** $(\boxed{\sigma}, \mathsf{pk}', \sigma')$;

Game: G^{unl};

1: $b \xleftarrow{\$} \{0,1\}$; $\mathsf{rr} \xleftarrow{\$} \{0,1\}^*$; $((\mathsf{sk}_0, \mathsf{pk}_0), (\mathsf{sk}_1, \mathsf{pk}_1), \mathsf{st}) \leftarrow \mathcal{A}(\lambda)$;
2: $\mathsf{pk}' \leftarrow \mathsf{RandPK}(\mathsf{pk}_b, \mathsf{rr})$; $\mathsf{sk}' \leftarrow \mathsf{RandSK}(\mathsf{sk}_b, \mathsf{pk}_b, \mathsf{rr})$;
3: **assert** $\mathsf{VerKey}(\mathsf{sk}_i, \mathsf{pk}_i) = 1$; for $i = 0, 1$; $b' \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}', \bullet)}(\mathsf{st}, \mathsf{pk}')$;
4: **return** $b = b'$;

Fig. 1: Security games for unforgeability (G^{rkuf}), unlinkability (G^{unl}) and for signature adaptation security (G^{adpt}); the boxed information is only given in the case of perfect adaptation (G^{padpt}).

Signature adaptation formalizes the idea that for any message, an adversary cannot distinguish whether a signature σ' is a newly constructed signature for a given public key pk', or whether it was adapted from a signature σ on a previous public key pk to the new one. Observe that we consider a slightly stronger notion of adaptation than the one from [17] in which the adversary is given multiple signatures on adaptively chosen messages. *Perfect*[12] adaptation implies that this statement still holds, even if the adversary is also given the original signature σ on pk.

Definition 6 (Signature Adaptation). *A signature scheme with randomizable keys provides* (perfect) signature adaptation *if for any PPT adversary $\mathcal{A}$ playing the game G^{adpt} (G^{padpt}) in Fig. 1, the winning probability* $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{adpt}}(\lambda) = \left|\Pr\left[G^{\mathsf{adpt}}(\mathcal{A}) = 1\right] - \frac{1}{2}\right|$ ($\mathsf{Adv}^{\mathcal{A}}_{\mathsf{padpt}}(\lambda) = \left|\Pr\left[G^{\mathsf{padpt}}(\mathcal{A}) = 1\right] - \frac{1}{2}\right|$) *is negligible in λ.*

[12] The "perfect" wording might be misleading, as it does not refer to the computational power of the adversary.

Another security notion of SWRKs is that of *unlinkability*, which ensures that randomized public keys are not linkable to the original ones (also referred to as *long-term* keys), even given any polynomial amount of signatures on them. In [17], the authors provide fine-grained distinctions to the various capabilities given to the adversary, for example it may be able to choose the public parameters or the long-term keys itself, or be limited in the signing queries it can make. Here we consider the stronger security notion in which the adversary generates the long-term keys and can query signatures on the randomized key. We do not consider discussions related to the setup since for isogenies all the public parameters are fixed by the scheme specifications for specific security parameters. In [17, Def. 16], this is denoted as $(0, 1, 3)$-unlinkability.

Definition 7 (Unlinkability). *An* SWRK *scheme is unlinkable if for any adversary* $\mathcal{A}$ *playing the game* G^{unl} *from Fig. 1, the winning probability* $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{unl}}(\lambda) = \left|\Pr\left[G^{\mathsf{unl}}(\mathcal{A}) = 1\right] - \frac{1}{2}\right|$ *is negligible in* λ.

3 Building Blocks and Existing Digital Signature Schemes

This section recalls the three digital signature schemes, SQIsign, PRISM and DeuringVUF, from which we build our SWRK schemes. We first introduce the common building blocks used throughout the paper.

3.1 Framework for building signatures

We first introduce the main building blocks that are required to construct all three signature schemes. As is natural in cryptographic schemes, our building blocks can be divided into two main types: algorithms to verify public data; and those to manipulate secret information. In the context of the signature schemes we consider, our public key will be an elliptic curve, whilst the secret key will be the endomorphism ring of this public curve. The hardness of key recovery is then ensured by the conjectured hardness of the endomorphism ring problem, which underlies the security of isogeny-based cryptography.

Beyond these fundamental objects, we will naturally need to generate points on our elliptic curves. For example, to define the two-dimensional representation of an isogeny $\varphi : E \to E'$, we need to generate points $P, Q \in E$ and compute $\varphi(P), \varphi(Q)$.

Generating and verifying public data. We begin with the building blocks that are needed for public data. The first will be the main subroutine of signature verification. Unifying all the signature schemes we consider is the way the signature isogeny is represented. More precisely, it is an isogeny of (large) prime degree given in two-dimensional representation. Thus, to verify our signatures we require a method to verify such representations.

> **Building Block 1:** VerIsogeny(σ, E, E', q) verifies that σ is a valid 2D representation of an isogeny $E \to E'$ of degree q. This algorithm, based on Kani's lemma, was introduced in [8, Alg. 7] (see also [5, § 3.3]).

Note that the input to this algorithm is all public data: σ will be (part of) the signature, and E, E' are public elliptic curves. As discussed above, another useful set of objects are points on our public curves. For the correctness of our protocols, different parties must be able to independently generate points on a given elliptic curve consistently, leading us to the following algorithm.

> **Building Block 2:** $(P, Q) \leftarrow \mathsf{Gen2Tor}(E, n)$. Given a supersingular curve E and an integer n such that $2^n \mid (p+1)$, this algorithm deterministically outputs a basis (P, Q) of $E[2^n]$. In the literature, there are several ways to implement such an algorithm; see, for instance, [1, §2.2.3].

Manipulating secret data. As detailed above, our secret data will mainly be endomorphism rings of public elliptic curves. In practice, we generate our endomorphism rings using isogenies: given an isogeny $\varphi : E \to E'$, where $\mathrm{End}(E)$ is known, we can transfer our knowledge of $\mathrm{End}(E)$ through φ to obtain $\mathrm{End}(E')$. Instantiating this in practice leads us to our first two building blocks:

> **Building Block 3:** $\mathrm{End}(E') \leftarrow \mathsf{IsoToEnd}(\varphi : E \to E', \mathrm{End}(E))$. On input a smooth degree isogeny $\varphi : E \to E'$ and $\mathrm{End}(E)$, this algorithm outputs a representation of $\mathrm{End}(E')$ (*e.g.*, as a $\mathbb{Z}$-basis of $\mathrm{End}(E')$ and/or an isogeny $\psi : E \to E'$ with its kernel ideal I_ψ). This algorithm has been described in [25, Alg. 8] (inspired from [38, Alg. 4]) when the kernel ideal I_φ of φ is also given as input. In App. C, we explain how to proceed when this additional input is not given and has to be computed from φ.
>
> **Building Block 4:** $(\varphi, \mathrm{End}(E')) \leftarrow \mathsf{GenIsogeny}(E, \mathrm{End}(E), d)$. On input an elliptic curve E of known endomorphism ring and an odd integer d, this algorithm outputs a uniformly distributed isogeny $\varphi : E \to E'$ of degree d. This algorithm was implemented in [5, Alg. 1]. We also remark that [5, Alg. 1] can be improved by integrating the techniques from [12].

With the knowledge of $\mathrm{End}(E')$, we are now able to generate new isogenies with specific properties.

> **Building Block 5:** $\sigma \leftarrow \mathsf{ShortIsogeny}(\mathrm{End}(E), \mathrm{End}(E'))$. On input two endomorphism rings $\mathrm{End}(E)$ and $\mathrm{End}(E')$, this algorithm returns a 2D representation of an isogeny $\sigma : E \to E'$ of degree $q \leq 2\sqrt{2p}/\pi$, or more precisely, a 2D-representation of an isogeny $\varphi \circ \sigma : E \to E''$ of degree $q(2^n - q)$ that factors through $\sigma : E \to E'$.
> This algorithm heavily relies on the Deuring correspondence. From the knowledge of quaternion orders $\mathcal{O} \simeq \mathrm{End}(E)$ and $\mathcal{O}' \simeq \mathrm{End}(E')$, we can find by simple quaternion arithmetic, an ideal I *connecting* $\mathcal{O}$ and $\mathcal{O}'$, which is a left $\mathcal{O}$-ideal and a right $\mathcal{O}'$-ideal. We can then sample an equivalent ideal $I_\sigma \sim I$ at random of norm $q \leq 2\sqrt{2p}/\pi$, which corresponds to an isogeny $\sigma : E \to E'$ of degree q via the Deuring correspondence. To generate a 2D-representation of σ, we generate a random auxiliary left $\mathcal{O}'$-ideal J of norm $2^n - q$ with [8, Alg. 3]. As we have seen at the end of Sec. 2.1, knowing $\mathrm{End}(E)$ gives us access to (a 2D-representation of) an isogeny $\varphi_0 : E_0 \to E$

and its associated ideal I_0. Then, applying [8, Alg. 2] to the ideal $I_0 \cdot I_\sigma \cdot J$, we obtain the image by $\varphi_J \circ \sigma \circ \varphi_0$ of a basis of $E_0[2^n]$. With an easy discrete logarithm computation, we infer the image by $\varphi_J \circ \sigma$ of a basis of $E[2^n]$, which yields a 2D representation of σ (and $\varphi_J \circ \sigma$).

Building Block 6: $(\gamma, \mathrm{End}(E'), m) \leftarrow \mathsf{EndomorphismWithFactor}_N(\varphi, \mathrm{End}(E), \mathrm{End}(E'))$. On input an isogeny $\varphi : E \to E'$, two endomorphism rings $\mathrm{End}(E)$ and $\mathrm{End}(E')$, this algorithm returns a non-scalar endomorphism $\gamma \in \mathrm{End}(E)$, such that $\gcd(\deg(\gamma), N) = 1$, another description of the endomorphism ring $\mathrm{End}(E')$ and an integer m. In practice, this algorithm computes another isogeny $\varphi' : E \to E'$, which is then represented by the new description of $\mathrm{End}(E')$. The quaternion γ corresponds to $\widehat{\varphi'} \circ \varphi$, whereas $m = \deg(\varphi')$.
This algorithm can be implemented as follows. As above, from the knowledge of $\mathrm{End}(E), \mathrm{End}(E')$, we can find an ideal I connecting $\mathcal{O}$ and $\mathcal{O}'$. We can then sample an equivalent ideal I' such that $I' = I\gamma/n(I)$ for some quaternion $\gamma \in \mathcal{B}_{p,\infty}$ of norm coprime to N, where $n(I)$ denotes the norm of I; define $\varphi' : E \to E'$ to be the isogeny corresponding to this ideal. The algorithm will output the endomorphism corresponding to γ together with a description of $\mathrm{End}(E')$ with respect to φ' in the sense of the endomorphism ring representation we discussed in Sec. 2.1. Additionally, the algorithm will output $m = n(I')$.

These building blocks suffice to describe SQIsign, PRISM, and the DeuringVUF signature abstractly.

3.2 SQIsign

SQIsign [1] is a signature scheme derived from a three-phase interactive identification protocol (commitment, challenge, response) via the Fiat-Shamir transform. The public key is a random supersingular elliptic curve E_{pk} and the associated secret key is its endomorphism ring $\mathsf{sk} = \mathrm{End}(E_{\mathsf{pk}})$. In practice, both are generated via GenIsogeny applied to E_0 of known endomorphism ring $\mathrm{End}(E_0)$ and a prime $\ell_{\mathsf{large}} > 2^{4\lambda}$. Along with $\mathrm{End}(E_{\mathsf{pk}})$, GenIsogeny returns an isogeny $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}$ of degree ℓ_{large}.

The signature algorithm $\mathsf{Sign}(\mathsf{msg}, \mathsf{sk}, \mathsf{pk})$ starts by a commitment phase where a random supersingular elliptic curve E_{com} is generated along with its endomorphism ring $\mathrm{End}(E_{\mathsf{com}})$, which is kept secret, using $\mathsf{GenIsogeny}(E_0, \mathrm{End}(E_0), \ell_{\mathsf{large}})$ like in the key generation phase. Then, a challenge isogeny $\varphi_{\mathsf{chall}} : E_{\mathsf{pk}} \to E_{\mathsf{chall}}$ of degree $\mathsf{d_{ch}}$ is generated by applying a hash function to the message, commitment and public key $\varphi_{\mathsf{chall}} \leftarrow \mathsf{H_{SQI}}(E_{\mathsf{pk}}, E_{\mathsf{com}}, \mathsf{msg})$. Finally, the signature is generated as an isogeny $\sigma : E_{\mathsf{com}} \to E_{\mathsf{chall}}$ of small degree (given in 2D representation) using the main tools of the Deuring correspondence (IsoToEnd applied to φ_{chall} and $\mathrm{End}(E_{\mathsf{pk}})$ to obtain $\mathrm{End}(E_{\mathsf{chall}})$, and ShortIsogeny applied to $\mathrm{End}(E_{\mathsf{com}})$ and $\mathrm{End}(E_{\mathsf{chall}})$ to obtain σ).

To verify a signature $\sigma := \mathsf{Sign}(\mathsf{msg}, \mathsf{sk}, \mathsf{pk})$, one can recompute $\varphi_{\mathsf{chall}} \leftarrow \mathsf{H_{SQI}}(E_{\mathsf{pk}}, E_{\mathsf{com}}, \mathsf{msg})$ and verify that σ is indeed a valid 2D representation of an isogeny $E_{\mathsf{com}} \to E_{\mathsf{chall}}$. SQIsign is summarized in Fig. 2.

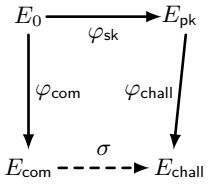
Security. By [8], the unforgeability of SQIsign reduces to the hardness of Prob. 1, assuming the validity of the two oracles below in Defs. 8 and 9.[13]

Problem 1. Given a curve E sampled from the stationary distribution on the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$, find a non-scalar endomorphism of E in efficient representation.

This problem is inherently related to the *supersingular endomorphism ring problem* [48, Thm. 7.2] that is considered the cornerstone of isogeny-based cryptography.

KeyGen(1^λ):

1: $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}, \mathrm{End}(E_{\mathsf{pk}}) \leftarrow$ GenIsogeny$(E_0, \mathrm{End}(E_0), \ell_{\mathsf{large}})$;
2: $\mathsf{sk} \leftarrow \mathrm{End}(E_{\mathsf{pk}})$;
3: $\mathsf{pk} \leftarrow E_{\mathsf{pk}}$;
4: **return** $(\mathsf{sk}, \mathsf{pk})$;



Sign(msg, sk, pk):

1: $(\varphi_{\mathsf{com}}, \mathrm{End}(E_{\mathsf{com}})) \leftarrow$ GenIsogeny$(E_0, \mathrm{End}(E_0), \ell_{\mathsf{large}})$;
2: $\varphi_{\mathsf{chall}} \leftarrow \mathsf{H}_{\mathsf{SQI}}(E_{\mathsf{pk}}, E_{\mathsf{com}}, \mathsf{msg})$;
3: $\mathrm{End}(E_{\mathsf{pk}}) \leftarrow \mathsf{sk}$;
4: $\mathrm{End}(E_{\mathsf{chall}}) \leftarrow$ IsoToEnd$(\varphi_{\mathsf{chall}}, \mathrm{End}(E_{\mathsf{pk}}))$;
5: $\sigma \leftarrow$ ShortIsogeny$(\mathrm{End}(E_{\mathsf{com}}), \mathrm{End}(E_{\mathsf{chall}}))$;
6: **return** $(\phi_{\mathsf{chall}}, \sigma, \mathsf{msg})$;

Verify(msg, σ, pk):

1: Extract E_{com}, the domain of σ;
2: $\varphi_{\mathsf{chall}} \leftarrow \mathsf{H}_{\mathsf{SQI}}(E_{\mathsf{pk}}, E_{\mathsf{com}}, \mathsf{msg})$;
3: Let E_{chall} be the codomain of φ_{chall};
4: $q \leftarrow \deg(\sigma)$;
5: **return** VerIsogeny$(\sigma, E_{\mathsf{com}}, E_{\mathsf{chall}}, q)$;

Fig. 2: SQIsign signature scheme.

Definition 8 (Definition 19 [8]). *A* uniform target oracle *(UTO) is an oracle that takes as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and outputs a random isogeny $\phi : E \to E'$ (in efficient representation) such that:*

1. *E' is sampled uniformly from the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$ with level structure;*
2. *The conditional distribution of φ given E' is uniform over isogenies $\psi : E \to E'$ of degree $\leq 2\sqrt{2p}/\pi$*

Definition 9 (Definition 23 [8]). *A* fixed degree isogeny oracle *(FIDIO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and an integer N, and outputs a uniformly random isogeny $\phi : E \to E'$ (in efficient representation) with domain E and degree N.*

[13] A more formal treatment of the security of SQIsign is given in [3], where the authors introduce a framework that captures the extra information given by these oracles via some *hints*. We prefer to use the argument in [8] for ease of exposition. However, we refer the interested reader to App. B for a proof of the unforgeability of our SWRK scheme based on SQIsign in the hint-based framework.

3.3 PRISM

The PRISM signature scheme [5] is an isogeny-based, hash-and-sign signature scheme that relies on the hardness of computing large prime-degree isogenies originating from elliptic curves whose endomorphism ring is unknown. In what follows, we will use the variant described in [6], as it gives better performance.

One of the most striking features is its conceptual simplicity. Indeed, to describe the signature scheme, we only need to recall the following three additional algorithms.

- $q' \leftarrow \mathsf{H}(\mathsf{msg}, \mathsf{salt})$: this is a hash function that, on input a message msg and a salt $\mathsf{salt} \in \{0,1\}^{2\lambda}$, outputs an integer $q' < 2^a$, where $a = \lambda + 64$.
- $(q, \mathsf{salt}) \leftarrow \mathsf{SaltedDegreeGenerator}(\mathsf{msg})$: this is a function that uses H as a subroutine. $\mathsf{SaltedDegreeGenerator}$ iteratively samples a random $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$, until $q \leftarrow \mathsf{H}(\mathsf{msg}, \mathsf{salt})$ is in Primes_a, the set of all primes q such that $2^{a-1} < q < 2^a$.[14]
- $\{0,1\} \leftarrow \mathsf{VerIsogeny}(\sigma, E, q)$: on input a 2D representation of $\sigma : E \to E_{\mathsf{sig}}$, a supersingular elliptic curve E and a prime $q \in \mathsf{Primes}_a$, the algorithm returns 1 if σ is a valid isogeny of degree $q(2^a - q)$ from E, else it outputs 0.

The security of PRISM relies on the collision resistance of the hash function H and on the hardness of Prob. 2, given below.

Problem 2. Given a random supersingular elliptic 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 $\bar{q}(2^a - \bar{q})$.

We give a description of the signature scheme in Fig. 3, where we fix E_0 to be the supersingular curve of j-invariant 1728 and define ℓ_{large} to be a large enough power of a prime to ensure the uniform distribution of the keys.

3.4 DeuringVUF signature

In [44], Leroux introduces a new verifiable unpredictable function (VUF) from which one can construct a hash-and-sign signature that shares similarities with PRISM; we refer to this construction as DeuringVUF. We present a simplified description of this scheme (Fig. 4), which will be sufficient for our purposes.[15]

In what follows, as in Sec. 3.3, let E_0 be the starting curve with j-invariant 1728, and let ℓ_{large} be a sufficiently large power of a prime to ensure the uniform distribution of the keys. We define N to be a large prime such that $N \mid (p-1)$;

[14] We highlight that we avoid public keys pk as part of the input to facilitate the adaptation process; see Sec. 4.3.

[15] Similarly to Sec. 3.3, we highlight that we avoid public keys pk as part of the input of $\mathsf{H}_{\mathsf{VUF}}$ for compatibility with the adaptation process; see Sec. 4.4.

KeyGen(1^λ):

1: $(\varphi_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}})) \leftarrow$ GenIsogeny$(E_0, \mathrm{End}(E_0), \ell_{\mathsf{large}})$;
2: $\mathsf{sk} \leftarrow \mathrm{End}(E_{\mathsf{pk}})$; $\mathsf{pk} \leftarrow E_{\mathsf{pk}}$;
3: **return** $(\mathsf{sk}, \mathsf{pk})$;

Sign(msg, sk, pk):

1: $(q, \mathsf{salt}) \leftarrow$ SaltedDegreeGenerator(msg);
2: $(\sigma : E_{\mathsf{pk}} \to E_{\mathsf{sig}}, \_) \leftarrow$ GenIsogeny$(E_{\mathsf{pk}}, \mathrm{End}(E_{\mathsf{pk}}), q(2^a - q))$;
3: $(P, Q) \leftarrow$ Gen2Tor(E_{pk}, a);
4: $(P_{\mathsf{sig}}, Q_{\mathsf{sig}}) \leftarrow (\sigma(P), \sigma(Q))$;
5: **return** $(E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}, \mathsf{salt})$;

Verify(msg, $E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}$, salt, pk):

1: $q \leftarrow \mathsf{H}(\mathsf{msg}, \mathsf{salt})$;
2: **assert** $q \in \mathsf{Primes}_a$;
3: $(P, Q) \leftarrow$ Gen2Tor(E_{pk}, a);
4: $\sigma \leftarrow (E_{\mathsf{pk}}, P, Q, E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}, q, a)$;
5: **return** VerIsogeny$(\sigma, E_{\mathsf{pk}}, q)$;

Fig. 3: PRISM signature scheme.

this implies that the N-torsion is defined over $\mathbb{F}_{p^4}$. Additionally, we fix a basis (P_0, Q_0) of $E_0[N]$ such that we know an endomorphism $\kappa \in \mathrm{End}(E_0)$ verifying $\mathrm{tr}(\kappa)^2 - 4\mathrm{nrd}(\kappa)$ is not a square modulo N and $\kappa(P_0) = Q_0$; this is a technical condition which enables fast signing, we refer the reader to [44, §3.1] for more details. Concretely, in [44, §4.2] $\kappa = \iota$ is fixed, since $N \equiv 1 \bmod 4$.

Remark 1. We modified the key generation procedure of [44] – via normalizing the public key – so that $e_N(P_{\mathsf{pk}}, Q_{\mathsf{pk}}) = e_N(P_0, Q_0)$, where e_N is the Weil pairing. This will be crucial in Sec. 5.4 to show that randomized public keys are indistinguishable from freshly generated ones. This comes at no loss of generality, since the Weil pairing is publicly computable.

To describe the hash-and-sign signature, we recall the algorithm

$$\mathsf{KerToIso}_N(E, \mathrm{End}(E), P, Q, \mathsf{M}, (r, s)).$$

On input a supersingular curve E of known $\mathrm{End}(E)$ together with a basis (P, Q) of $E[N]$ and $\mathsf{M} \in M_2(\mathbb{Z}/N\mathbb{Z})$, where $\binom{P}{Q} = \mathsf{M} \cdot \binom{\varphi(P_0)}{\varphi(Q_0)}$ for a known isogeny $\varphi : E_0 \to E$ such that $N \nmid \deg(\varphi)$, and two scalars (r, s), it returns a 2D representation of $\sigma : E \to E'$ such that $\deg(\sigma) = N$ and $\sigma(rP + sQ) = 0$. We refer the reader to [44, Alg. 2, lines 2-7] for a concrete description of this algorithm.

In [44], the author argues that the above signature scheme is existentially unforgeable under chosen message attacks assuming the collision resistance of the hash function $\mathsf{H}_{\mathsf{VUF}}$ and the hardness of a new problem. This problem is defined with respect to a new oracle, returning a fixed degree isogeny.

Definition 10 (Definition 4 [44]). *A* fixed degree N-isogeny oracle *(FIXDIO) is an oracle taking as input a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ and a point $K \in E[N]$, and outputs the N-isogeny $\phi : E \to E/\langle K \rangle$ (in efficient representation).*

Problem 3. Given a uniformly random supersingular elliptic curve $E \xleftarrow{\$} \mathsf{Ell}_p$ and access to a FIXDIO on input E, output an efficient representation of an isogeny of degree N whose kernel is different from all kernels formerly queried to the oracle.

KeyGen(1^λ):

1: $(\varphi_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}})) \leftarrow$ GenIsogeny$(E_0, \mathrm{End}(E_0), \ell_{\mathsf{large}})$;
2: $\mathsf{M}_{\mathsf{sk}} \xleftarrow{\$} \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$;
3: $(P_{\mathsf{pk}}, Q_{\mathsf{pk}}) \leftarrow \ell_{\mathsf{large}}^{-1/2} \cdot \mathsf{M}_{\mathsf{sk}} \cdot \begin{pmatrix} \varphi_{\mathsf{sk}}(P_0) \\ \varphi_{\mathsf{sk}}(Q_0) \end{pmatrix}$
4: $\mathsf{sk} \leftarrow (\ell_{\mathsf{large}}^{-1/2} \cdot \mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}}))$
5: $\mathsf{pk} \leftarrow (E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}})$
6: **return** $(\mathsf{sk}, \mathsf{pk})$

Sign(msg, sk, pk):

1: $(r : s) \leftarrow \mathsf{H}_{\mathsf{VUF}}(\mathsf{msg})$
2: parse sk as $(\mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}}))$
3: parse pk as $(E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}})$
4: $\sigma \leftarrow \mathsf{KerToIso}_N(E_{\mathsf{pk}}, \mathrm{End}(E_{\mathsf{pk}}), P_{\mathsf{pk}}, Q_{\mathsf{pk}}, \mathsf{M}_{\mathsf{sk}}, (r, s))$
5: $(P_{\mathsf{sig}}, Q_{\mathsf{sig}}) \leftarrow (\sigma(P_{\mathsf{pk}}), \sigma(Q_{\mathsf{pk}}))$
6: **return** $(E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}})$

Verify(msg, $E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}$, pk):

1: $\sigma \leftarrow (E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}}, E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}, N, f)$
2: $(r : s) \leftarrow \mathsf{H}_{\mathsf{VUF}}(\mathsf{msg})$
3: **return** $\sigma(rP_{\mathsf{pk}} + sQ_{\mathsf{pk}}) \stackrel{?}{=} 0 \wedge \deg(\sigma) \stackrel{?}{=} N$

Fig. 4: DeuringVUF signature scheme.

Remark 2. Unlike FIDIO, which takes a curve E and degree N and returns a uniform outgoing N-isogeny, FIXDIO is fixed to one curve and degree and deterministically maps an input kernel point to the corresponding isogeny.

4 Isogeny-based signatures with randomizable keys

In this section, we introduce an SWRK scheme based on SQIsign, which we denote by SQIsign-RK, and two aSWRK schemes based on PRISM and DeuringVUF, which we denote by PRISM-RK and DeuringVUF-RK, respectively. The main tool to guarantee adaptability is an algorithm to compute pushforwards, which we describe in Sec. 4.2. We summarize the parameters used in the three constructions in Tab. 2.

In all three constructions, we will update public keys by computing randomization isogenies $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$ of degree $\deg(\phi_{\mathsf{rr}}) = 2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$, where r_{rand}, u_{cmp} will depend on the construction. The exact size of r_{rand} will be discussed in Sec. 5.4.

4.1 SQIsign with randomizable keys

In SQIsign-RK, the algorithms KeyGen, Sign and Verify are the same as in SQIsign (see Sec. 3.2); we only need to describe algorithms to randomize and verify keys.

In our setting, randomizing keys boils down to computing long isogenies originating from the public curve E_{pk}. To be more precise, given a public key E_{pk}, we compute an isogeny $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$, such an isogeny ϕ_{rr} must be long enough to ensure that the new public key $\mathsf{pk}' = E'_{\mathsf{pk}}$ is uniformly distributed among all possible public keys.

To concretely define algorithms to randomize keys, we need to explain how to encode randomness into isogeny paths. To this end, we introduce a hash function $\mathsf{H}_{\mathsf{path}} : \{0,1\}^* \to (\mathbb{Z}/2^{u_{\mathsf{cmp}}}\mathbb{Z})^{r_{\mathsf{rand}}}$, for $u_{\mathsf{cmp}} \leq f$. Then, we define Expand to be an algorithm that, on input a curve E_{pk} and some randomness rr, describes an

Table 2: Summary of parameters used in the constructions.

Parameter	Description
p	prime of the form $p = c \cdot 2^f - 1$.
E_0	Curve $y^2 = x^3 + x$ with known endomorphism ring.
ℓ_{large}	Large degree used to generate uniformly distributed curves.
ϕ_{rr}	Randomization isogeny.
r_{rand}	Number of smooth components in the randomization path.
$\mathsf{H}_{\mathsf{path}}$	Hash function expanding the randomizer into the data defining the randomization path.
a	2-torsion exponent for 2D representations of signatures
u_{cmp}	2-torsion exponent for each smooth component for Expand ($u_{\mathsf{cmp}} \leq f$) and ComputePushforward ($u_{\mathsf{cmp}} \leq f - a$).
q	Prime degree for PRISM signatures.
N	Large prime torsion order used by DeuringVUF.
$P_{\mathsf{pk}}, Q_{\mathsf{pk}}$	Public N-torsion basis included in DeuringVUF public keys.
η	Normalizing scalar $\deg(\phi_{\mathsf{rr}})^{-1/2} \bmod N$ for DeuringVUF-RK.

isogeny $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$ of degree $2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$; we refer to Alg. 1 for more details about the algorithm.

Algorithm 1 Expand$(E_{\mathsf{pk}}, \mathsf{rr})$

$\mathsf{rr}_1, \ldots, \mathsf{rr}_{r_{\mathsf{rand}}} \leftarrow \mathsf{H}_{\mathsf{path}}(E_{\mathsf{pk}}, \mathsf{rr})$.
Deterministically compute P_1, Q_1, a $2^{u_{\mathsf{cmp}}}$-torsion basis in $E_{(1)} := E_{\mathsf{pk}}$.
for $i = 1, \ldots, r_{\mathsf{rand}}$ **do**
    Compute $\phi_i : E_{(i)} \to E_{(i+1)}$ with kernel $\langle P_i + [\mathsf{rr}_i]Q_i \rangle$; $Q_{i+1} \leftarrow \phi_i(Q_i)$.
    Deterministically compute $P_{i+1} \in E_{(i+1)}[2^{u_{\mathsf{cmp}}}]$ linearly independent to Q_{i+1}.
$\phi_{\mathsf{rr}} \leftarrow \phi_{r_{\mathsf{rand}}} \circ \cdots \circ \phi_1$; $E'_{\mathsf{pk}} \leftarrow E_{(r_{\mathsf{rand}}+1)}$
return $\phi_{\mathsf{rr}}, E'_{\mathsf{pk}}$

Equipped with Expand, we can define the algorithms RandPK and RandSK that allow us to randomize public keys and private keys, respectively. In particular, to randomize secret keys, we use IsoToEnd: given a secret key $\mathsf{sk} = \mathrm{End}(E_{\mathsf{pk}})$, we randomize sk by computing $\mathsf{sk}' = \mathrm{End}(E'_{\mathsf{pk}}) \leftarrow \mathsf{IsoToEnd}(\phi_{\mathsf{rr}}, \mathrm{End}(E_{\mathsf{pk}}))$. Lastly, we need to explain how to verify keys. Given a secret key sk, which consists of an endomorphism ring $\mathrm{End}(E')$ for some supersingular elliptic curve E', we run the algorithm ShortIsogeny$(\mathrm{End}(E_0), \mathsf{sk})$ to obtain an isogeny $\varphi : E_0 \to E'$ and check that E' coincides with $\mathsf{pk} = E_{\mathsf{pk}}$. We present these in Fig. 5. For SQIsign-RK we set $u_{\mathsf{cmp}} = f$.

RandPK(pk, rr):

1: $E_{\mathsf{pk}} \leftarrow \mathsf{pk}$;
2: $(\phi_{\mathsf{rr}}, E'_{\mathsf{pk}}) \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
3: **return** E'_{pk}.

RandSK(sk, pk, rr):

1: $\mathrm{End}(E_{\mathsf{pk}}) \leftarrow \mathsf{sk}$; $E_{\mathsf{pk}} \leftarrow \mathsf{pk}$;
2: $\phi_{\mathsf{rr}}, E'_{\mathsf{pk}} \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
3: $\mathrm{End}(E'_{\mathsf{pk}}) \leftarrow \mathsf{IsoToEnd}(\phi_{\mathsf{rr}}, \mathrm{End}(E_{\mathsf{pk}}))$;
4: **return** $\mathrm{End}(E'_{\mathsf{pk}})$

VerKey(pk, sk):

1: $(\varphi : E_0 \to E') \leftarrow \mathsf{ShortIsogeny}(\mathrm{End}(E_0), \mathsf{sk})$; $E_{\mathsf{pk}} \leftarrow \mathsf{pk}$
2: **return** $E' \stackrel{?}{=} E_{\mathsf{pk}}$

Fig. 5: RandSK, RandPK and VerKey algorithms for SQIsign-RK, with $u_{\mathsf{cmp}} = f$.

4.2 The update algorithm

In this section, we introduce the main tool that allows us to adapt signatures: an efficient algorithm for computing the pushforward of a 2D representation of an isogeny σ under a second isogeny ϕ of smooth degree.

The goal is to update a signature $\sigma : E \to E_\sigma$ by computing its pushforward under an isogeny $\phi : E \to E'$. The update algorithm takes:

- A 2D representation of σ, *i.e.*, a tuple $(E, P, Q, E_\sigma, P_\sigma, Q_\sigma, q, a)$, where $P, Q \in E[2^a]$ and $P_\sigma, Q_\sigma \in E_\sigma[2^a]$, for some $a < f$.
- A $2^\bullet$-isogeny ϕ, represented by a rational point $K \in E(\mathbb{F}_{p^2})$ such that $\ker \phi = \langle K \rangle$.

This problem is not new: in [53, Prop. 6.15], Robert gives a polynomial-time algorithm to solve this task as long as the isogenies involved have coprime degrees. However, even though these general techniques are polynomial time, they are not efficient in practice. Indeed, they rely on the computation of four- and eight-dimensional isogenies, which need to be evaluated on a torsion basis defined over an extension of the base field. More efficient techniques were proposed in [47,9,45], but they are incompatible with our setting. Furthermore, the order of the points in the representation of σ is not coprime with the degree of the isogeny ϕ (they are both powers of two), which adds further complications.

At a high level, the update algorithm first computes the pushforward $\phi' = [\sigma]_*\phi$ using the rational kernel of ϕ: if $\ker \phi = \langle K \rangle$, then $\ker \phi' = \langle \sigma(K) \rangle$. From ϕ', one could hope to obtain a representation of the pushforward $\sigma' = [\phi]_*\sigma$ by mapping a torsion basis on E_1 under $\phi' \circ \sigma \circ \hat{\phi}$, obtaining the image of a torsion basis under σ', after the appropriate scaling. This is indeed possible if the degree of ϕ and the order of the points used in the 2D representation is coprime.

However, the torsion basis we use in the 2D representation has order a power of two, both for efficiency reasons (2D representations are much more efficient with such points) and because the only torsion basis of large order in $E(\mathbb{F}_{p^2})$ has order a power of two; the degree of the isogenies ϕ and ϕ' is also a power of two, which prevents a straightforward application of this strategy.

Instead, we rely on the fact that the points in the 2D representation of σ do not have maximal order. Recall that the order of these points is 2^a, while the prime characteristic is $p = c \cdot 2^f - 1$, with $f > a$. We thus pick $u_{\mathsf{cmp}} \leq f - a$, and

we split the isogeny ϕ as the composition of multiple isogenies $\phi = \phi_{r_{\mathsf{rand}}} \circ \cdots \circ \phi_1$ where $\deg \phi_i = 2^{u_{\mathsf{cmp}}}$. We then compute the pushforward isogeny σ' stepwise, as depicted by Fig. 6.

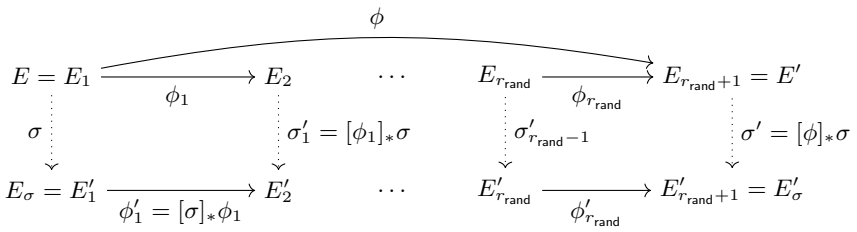


Fig. 6: Computation of σ update. Dotted lines represent isogenies with a 2D representation, while solid lines are isogenies represented by a kernel generator.

The first step is to compute the $\sigma'_1 = [\phi_1]_*\sigma$, where $\phi_1 : E \to E_1$. Let P_1, Q_1 be a random basis of $E_1[2^f]$, and write $\phi'_i = [\sigma]_*\phi_i$. We have

$$\sigma'_1([2^{u_{\mathsf{cmp}}}]P_1) = \phi'_1 \circ \sigma \circ \widehat{\phi}_1(P_1) \quad \text{and} \quad \sigma'_1([2^{u_{\mathsf{cmp}}}]Q_1) = \phi'_1 \circ \sigma \circ \widehat{\phi}_1(Q_1).$$

This means we can evaluate σ'_1 on a 2^a-basis obtained from $[2^{u_{\mathsf{cmp}}}]P_1, [2^{u_{\mathsf{cmp}}}]Q_1$ because we know ϕ_1, ϕ'_1, and σ. From this, we can obtain a 2D representation of σ'_1. We then repeat this process for each component ϕ_i to obtain σ' as required. We summarize this algorithm in Alg. 2, which we denote by ComputePushforward.

Algorithm 2 ComputePushforward(σ, ϕ)

Input: An isogeny $\sigma : E \to E_\sigma$, given in 2D representation, *i.e.*, a tuple $(E, P, Q, E_\sigma, P_\sigma, Q_\sigma, q, a)$, where $P, Q \in E[2^a]$ and $P_\sigma, Q_\sigma \in E_\sigma[2^a]$.
An isogeny $\phi : E \to E'$ of degree 2^m, where $m = u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}$ and $u_{\mathsf{cmp}} \leq f - a$, represented by a kernel generator.

Output: A 2D representation of $[\phi]_*\sigma$

Write $\phi = \phi_{r_{\mathsf{rand}}} \circ \cdots \circ \phi_1$ for $\deg(\phi_i) = 2^{u_{\mathsf{cmp}}}$
Write $E_1 = E$, $E_{r_{\mathsf{rand}}+1} = E'$, $\phi_i : E_i \to E_{i+1}$, $E'_1 = E_\sigma$
for $i = 1, \ldots, r_{\mathsf{rand}}$ **do**
    Compute $\phi'_i : E'_i \to E'_{i+1}$ with kernel $\langle \sigma(K) \rangle$ for $K \leftarrow \ker \phi_i$.
    Sample a deterministic basis P, Q of $E'_{i+1}[2^f]$.
    $P' \leftarrow \phi'_i \circ \sigma \circ \widehat{\phi}_i(P)$; $Q' \leftarrow \phi'_i \circ \sigma \circ \widehat{\phi}_i(Q)$
    $\sigma \leftarrow (E'_{i+1}, [2^{f-a}]P, [2^{f-a}]Q, [2^{f-a-u_{\mathsf{cmp}}}]P', [2^{f-a-u_{\mathsf{cmp}}}]Q', q, a)$
return σ

4.3 Adaptable PRISM with randomizable keys

In this section, we explain how to instantiate an aSWRK scheme based on PRISM. To randomize and verify keys, we employ the same algorithms as in SQIsign-RK

(see Sec. 4.1). Extra care must be devoted to the adaptation algorithm: given a signature $\sigma : E_{\mathsf{pk}} \to E_{\mathsf{sig}}$, we adapt it to a new signature for pk' by computing $(\sigma' : E'_{\mathsf{pk}} \to E'_{\mathsf{sig}}) \leftarrow \mathsf{ComputePushforward}(\sigma, \phi_{\mathsf{rr}})$ with $u_{\mathsf{cmp}} = f - a$.

When adapting signatures, the algorithm will reveal $[\phi_{\mathsf{rr}}]_*\sigma$, for some signature isogeny $\sigma : E_{\mathsf{pk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$. We summarize the Adapt algorithm of our aSWRK scheme in Fig. 7; note that KeyGen, Sign and Verify are as in Fig. 3, and RandSK, RandPK and VerKey are the same as in Fig. 5, but for $u_{\mathsf{cmp}} = f - a$.

$\mathsf{Adapt}(\mathsf{msg}, E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}, \mathsf{salt}, \mathsf{rr}, \mathsf{pk})$:
1: $E_{\mathsf{pk}} \leftarrow \mathsf{pk}$;
2: $q \leftarrow \mathsf{H}(\mathsf{msg}, \mathsf{salt})$;
3: $(P, Q) \leftarrow \mathsf{Gen2Tor}(E_{\mathsf{pk}}, a)$;
4: $\sigma \leftarrow (E_{\mathsf{pk}}, P, Q, E_{\mathsf{sig}}, P_{\mathsf{sig}}, Q_{\mathsf{sig}}, q, a)$;
5: **assert** $\mathsf{VerIsogeny}(\sigma, E_{\mathsf{pk}}, E_{\mathsf{sig}}, q)$;
6: Get $\phi_{\mathsf{rr}}, E'_{\mathsf{pk}} \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
7: $(\sigma' : E'_{\mathsf{pk}} \to E'_{\mathsf{sig}}) \leftarrow \mathsf{ComputePushforward}(\sigma, \phi_{\mathsf{rr}})$;
8: $(P', Q') \leftarrow \mathsf{Gen2Tor}(E'_{\mathsf{pk}}, a)$;
9: $(P'_{\mathsf{sig}}, Q'_{\mathsf{sig}}) \leftarrow (\sigma'(P'), \sigma'(Q'))$;
10: **return** $(E'_{\mathsf{sig}}, P'_{\mathsf{sig}}, Q'_{\mathsf{sig}}, \mathsf{salt})$

Fig. 7: Randomizable PRISM, denoted PRISM-RK. The algorithms Expand, RandPK, RandSK and VerKey are the same as in Sec. 4.1.

4.4 Adaptable DeuringVUF signature with randomizable keys

Using the previous tools, with again $u_{\mathsf{cmp}} = f - a$, we also instantiate an aSWRK scheme based on the DeuringVUF signature (see Sec. 3.4), denoted DeuringVUF-RK; we present the needed algorithms in Fig. 8. Notable differences from the procedure in Fig. 5 are:

1. we need to ensure that the randomized basis $P'_{\mathsf{pk}}, Q'_{\mathsf{pk}}$ has the same Weil pairing value as the new basis $P_{\mathsf{pk}}, Q_{\mathsf{pk}}$ after randomization. For this, in RandPK, we multiply the basis by $\eta = \deg(\phi_{\mathsf{rr}})^{-1/2} \bmod N$.
2. to sign after a key randomization we need to recover the action of the secret key φ_{sk} under P_0, Q_0 and the new randomized basis. For this, we define P_0, Q_0 so that we can compute the action of any endomorphism on them. We follow the procedure described in [13], which is compatible with DeuringVUF public setup in [44, §4.2].

Public setup. We select ℓ and N so that ℓ is a square modulo N, so that $\deg(\phi_{\mathsf{rr}})^{-1/2} \bmod N$ can be efficiently computed. For instance, if $\ell = 2$ it is enough $N \equiv 1 \bmod 8$. We also require that $f > a$, as specified in Sec. 4.2.

To compute the action of the secret key φ_{sk} on the new basis, we follow the blueprint from [13]. To simplify the discussion, we assume that $N \equiv 1 \bmod 4$, as

in [44, §4.2], so that $\kappa = \iota$. First, by [13, Lem. 5] – and subsequent discussion – we just need to be able to compute the action of the endomorphism $\gamma : E_0 \to E_0$ factoring through $\varphi_{\mathsf{sk}}, \phi_{\mathsf{rr}}$ and φ'_{sk}, similarly to what is done in [13, Alg. 1]. Let M_γ be the matrix describing the action of γ on $E_0[N]$ with respect to a basis of $E_0[N]$. As in [13, Rem. 6], we need to select $P_0, Q_0 \in E_0[N]$ and ζ such that:

- 1, ι, ζ, $\iota\zeta$ generate $\mathrm{End}(E_0)/N\,\mathrm{End}(E_0)$ as a $(\mathbb{Z}/N\mathbb{Z})$-module, where ι is the endomorphism on E_0 corresponding to $\sqrt{-1}$;
- $\zeta(P_0) = [\alpha]P_0$, for some $\alpha \neq 0 \bmod N$ and $Q_0 = \iota(P_0)$ (recall that $\iota = \kappa$);
- (P_0, Q_0) form a basis for $E_0[N]$.

Decompose $\zeta \circ \iota$ as $m_1 + m_2\iota + m_3\zeta + m_4\iota\zeta$ in $\mathrm{End}(E_0)/N\,\mathrm{End}(E_0)$. In this way, with respect to the basis (P_0, Q_0), we have $\mathsf{M}_\iota = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ and $\mathsf{M}_\zeta = \left(\begin{smallmatrix} \alpha & 0 \\ m_1+\alpha m_3 & m_2+\alpha m_4 \end{smallmatrix}\right)$. The matrices associated to $1, \iota\zeta$ are clearly the identity and $\mathsf{M}_\iota\mathsf{M}_\zeta$, respectively. We define $\mathsf{ComputeAction}_N$ to be the algorithm that outputs M_γ by writing γ as a combination of 1, ι, ζ and $\iota\zeta$ (that form a basis) and then using the matrices defined above.

To finally compute ζ and the basis (P_0, Q_0) we set $\zeta = \pi + n\iota$, where π is the Frobenius endomorphism and $(-p - n^2)$ is a square modulo N; define α to be one of the square roots. To compute P_0, we sample $P' \in E_0[N]$ until $P := (\zeta + \alpha)(P')$ is of order N. We set $P_0 = P$, so that $\zeta(P_0) = [\alpha]P_0$. We also observe that $\langle P_0 \rangle$ intersects trivially with $\langle \iota(P_0) \rangle$. Suppose, by contradiction, that $\iota(P_0)$ and P_0 are linearly dependent. Since the order N of P_0 is an odd prime, $\langle P_0 \rangle$ is also an eigenspace of ι. This implies that ι and ζ commute on $\langle P_0 \rangle$, so P_0 is in the kernel of $(\iota\zeta - \zeta\iota) = 2\iota\pi$. Since π is inseparable (so has trivial kernel), P_0 must be a point of order 2, which gives a contradiction. Thus (P_0, Q_0) forms a basis of $E_0[N]$. We finally remark that 1, ι, ζ, $\iota\zeta$ generate $\mathrm{End}(E_0)/N\,\mathrm{End}(E_0)$ since ι and ζ do not commute. Also, the kernel ideal of P_0 can be found as $I_{P_0} = \mathcal{O}_0\langle \pi - \alpha, N \rangle$.

Algorithmic details. Equipped with $\mathsf{ComputeAction}_N$, we explain how to instantiate RandSK. Let $\mathsf{sk} = (\mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}}))$. We observe that, given $\mathrm{End}(E_{\mathsf{pk}})$, we have an efficient representation of φ_{sk}: as explained in Sec. 2.1, we use a description of $\mathrm{End}(E_{\mathsf{pk}})$ that is equivalent to knowing φ_{sk}. As done for the previous constructions, we compute an isogeny $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$. Given ϕ_{rr} and φ_{sk}, we run $\mathsf{EndomorphismWithFactor}_N(\phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}}, \mathrm{End}(E_0), \mathrm{End}(E'_{\mathsf{pk}}))$ to get an endomorphism γ, together with a new description of $\mathrm{End}(E'_{\mathsf{pk}})$ and an integer m. Finally, we use $\mathsf{ComputeAction}_N(\gamma)$ to obtain $\mathsf{M}_\gamma \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ giving the action of γ on $E_0[N]$. The updated secret matrix will then be $\mathsf{M}' = (m^{-1} \pmod N)\mathsf{M}_{\mathsf{sk}}\mathsf{M}_\gamma$.

The algorithm VerKey simply checks that, given $\mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}})$, we can recompute the secret isogeny φ_{sk}, which should map onto E_{pk} and $(P_{\mathsf{pk}}, Q_{\mathsf{pk}}) = \mathsf{M}_{\mathsf{sk}} \cdot \varphi_{\mathsf{sk}}(P_0, Q_0)$, assuming that $(P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ describes a basis of $E_{\mathsf{pk}}[N]$.

The Adapt algorithm works in a similar way as in Sec. 4.3. We summarize our construction in Fig. 8.

4.5 Correctness

We finish this section by proving the correctness of all schemes.

RandSK(sk, pk, rr):

1: $\mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}}) \leftarrow \mathsf{sk}$;
2: From $\mathrm{End}(E_{\mathsf{pk}})$, recover a representation of $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}$;
3: $\phi_{\mathsf{rr}}, E'_{\mathsf{pk}} \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
4: $\mathrm{End}(E'_{\mathsf{pk}}) \leftarrow \mathsf{IsoToEnd}(\phi_{\mathsf{rr}}, \mathrm{End}(E_{\mathsf{pk}}))$;
5: $\gamma, \mathrm{End}(E'_{\mathsf{pk}}), m \leftarrow \mathsf{EndomorphismWithFactor}_N(\phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}}, \mathrm{End}(E_0), \mathrm{End}(E'_{\mathsf{pk}}))$
6: $\mathsf{M}_\gamma \leftarrow \mathsf{ComputeAction}_N(\gamma)$;
7: $\mathsf{M}'_{\mathsf{sk}} \leftarrow (m^{-1} \pmod N)\mathsf{M}_{\mathsf{sk}}\mathsf{M}_\gamma$;
8: **return** $\mathsf{M}'_{\mathsf{sk}}, \mathrm{End}(E'_{\mathsf{pk}})$

RandPK(pk, rr):

1: $E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}} \leftarrow \mathsf{pk}$;
2: $(\phi_{\mathsf{rr}}, E'_{\mathsf{pk}}) \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
3: $P'_{\mathsf{pk}}, Q'_{\mathsf{pk}} \leftarrow \eta\phi_{\mathsf{rr}}(P_{\mathsf{pk}}), \eta\phi_{\mathsf{rr}}(Q_{\mathsf{pk}})$;
4: **return** $E'_{\mathsf{pk}}, P'_{\mathsf{pk}}, Q'_{\mathsf{pk}}$

Adapt(msg, σ, rr, pk):

1: $E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}} \leftarrow \mathsf{pk}$;
2: $\phi_{\mathsf{rr}}, E'_{\mathsf{pk}} \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$;
3: $P'_{\mathsf{pk}}, Q'_{\mathsf{pk}} \leftarrow \eta\phi_{\mathsf{rr}}(P_{\mathsf{pk}}), \eta\phi_{\mathsf{rr}}(Q_{\mathsf{pk}})$;
4: $\mathsf{pk}' \leftarrow (E'_{\mathsf{pk}}, P'_{\mathsf{pk}}, Q'_{\mathsf{pk}})$;
5: $(\sigma' : E'_{\mathsf{pk}} \to E'_{\mathsf{sig}}) \leftarrow$ $\mathsf{ComputePushforward}(\sigma, \phi_{\mathsf{rr}})$;
6: $(P'_{\mathsf{sig}}, Q'_{\mathsf{sig}}) \leftarrow (\sigma'(P'_{\mathsf{pk}}), \sigma'(Q'_{\mathsf{pk}}))$;
7: **return** $(E'_{\mathsf{sig}}, P'_{\mathsf{sig}}, Q'_{\mathsf{sig}})$

VerKey(pk, sk):

1: $E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}} \leftarrow \mathsf{pk}$; $\mathsf{M}_{\mathsf{sk}}, \mathrm{End}(E_{\mathsf{pk}}) \leftarrow \mathsf{sk}$;
2: From $\mathrm{End}(E_{\mathsf{pk}})$, recover a representation of $\varphi_{\mathsf{sk}} : E_0 \to E'$;
3: **assert** $P_{\mathsf{pk}}, Q_{\mathsf{pk}}$ have order N;
4: **return** $\left(E' \stackrel{?}{=} E_{\mathsf{pk}}\right) \wedge \left((P_{\mathsf{pk}}, Q_{\mathsf{pk}}) \stackrel{?}{=} \mathsf{M}_{\mathsf{sk}} \cdot \varphi_{\mathsf{sk}}(P_0, Q_0)\right) \wedge$
$\left(e_N(P_0, Q_0) \stackrel{?}{=} e_N(P_{\mathsf{pk}}, Q_{\mathsf{pk}})\right)$.

Fig. 8: Randomizable DeuringVUF DeuringVUF-RK. The algorithm Expand is the same as in SQIsign-RK. The KeyGen, Sign and Verify algorithms are the same as in the original protocol, given in Fig. 4. Note that $\eta = \deg(\phi_{\mathsf{rr}})^{-1/2} \bmod N$.

Proposition 1. *The* SWRK *schemes* SQIsign-RK *(Sec. 4.1),* PRISM-RK *(Sec. 4.3) and* DeuringVUF-RK *(Sec. 4.4) are correct, as per Def. 4.*

Proof. The correctness of SQIsign-RK and PRISM-RK follows by construction. We therefore focus on DeuringVUF-RK. Let DeuringVUF-RK = (KeyGen, Sign, Verify, RandPK, RandSK, VerKey, Adapt) defined in Sec. 4.4. We first show that, for all randomness rr, it holds that

$$\mathsf{VerKey}(\mathsf{pk}, \mathsf{sk}) = \mathsf{VerKey}(\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}), \mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr}))\,. \tag{3}$$

Let $\mathsf{sk} = (\mathsf{M}_{\mathsf{sk}}, \mathcal{O})$ and $\mathsf{pk} = (E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}})$, and suppose $\mathsf{RandSK}(\mathsf{sk}, \mathsf{pk}, \mathsf{rr}) = (\mathsf{M}'_{\mathsf{sk}}, \mathcal{O}')$ and $\mathsf{RandPK}(\mathsf{pk}, \mathsf{rr}) = (E'_{\mathsf{pk}}, P'_{\mathsf{pk}}, Q'_{\mathsf{pk}})$. We have:

– By the correctness of Expand, IsoToEnd and $\mathsf{EndomorphismWithFactor}_N$, $\mathcal{O}'$ is (a representation of) the endomorphism ring of E'_{pk} if and only if $\mathcal{O}$ is (a representation of) the endomorphism ring of E_{pk}. As a result, in Line 2 of VerKey, from $\mathcal{O}'$ we correctly recover a representation of $\varphi'_{\mathsf{sk}} : E_0 \to E'$ if and only if we can correctly extract $\varphi_{\mathsf{sk}} : E_0 \to E$ from $\mathcal{O}$. Namely, $E' = E'_{\mathsf{pk}}$ if and only if $E = E_{\mathsf{pk}}$.

– As ϕ_{rr} has degree coprime to N, $P'_{\mathsf{pk}}, Q'_{\mathsf{pk}}$ have order N if and only if $P_{\mathsf{pk}}, Q_{\mathsf{pk}}$ have order N, so the assertion in Line 3 passes.
– By definition, we have $\mathsf{M}'_{\mathsf{sk}} = (\deg(\varphi_{\mathsf{sk}})^{-1} \bmod N) \cdot \mathsf{M}_{\mathsf{sk}}\mathsf{M}_\gamma$, where M_γ gives the action of $\gamma \in \mathrm{End}(E_0)$ on $P_0, Q_0 \in E_0[N]$. By the correctness of $\mathsf{EndomorphismWithFactor}_N$, $\gamma = \widehat{\varphi'_{\mathsf{sk}}} \circ \phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}}$. Therefore, we have

$$\begin{aligned}\mathsf{M}'_{\mathsf{sk}} \cdot \begin{pmatrix}\varphi'_{\mathsf{sk}}(P_0)\\ \varphi'_{\mathsf{sk}}(Q_0)\end{pmatrix} &= (\deg(\varphi_{\mathsf{sk}})^{-1} \bmod N) \cdot \mathsf{M}_{\mathsf{sk}}\mathsf{M}_\gamma \cdot \begin{pmatrix}\varphi'_{\mathsf{sk}}(P_0)\\ \varphi'_{\mathsf{sk}}(Q_0)\end{pmatrix}\\ &= \mathsf{M}_{\mathsf{sk}} \cdot \begin{pmatrix}\phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}}(P_0)\\ \phi_{\mathsf{rr}} \circ \varphi_{\mathsf{sk}}(Q_0)\end{pmatrix} = \begin{pmatrix}P'_{\mathsf{pk}}\\ Q'_{\mathsf{pk}}\end{pmatrix},\end{aligned}$$

if and only if $\mathsf{M}_{\mathsf{sk}} \cdot \begin{pmatrix}\varphi_{\mathsf{sk}}(P_0)\\ \varphi_{\mathsf{sk}}(Q_0)\end{pmatrix} = \begin{pmatrix}P_{\mathsf{pk}}\\ Q_{\mathsf{pk}}\end{pmatrix}$.

Combining these, we get correctness of the key randomization algorithms. The correctness of the algorithm Adapt follows from the correctness of ComputePushforward.

5 Security Properties

This section analyzes the security of the SWRK schemes from Sec. 4. We show in Sec. 5.1 that the pushforward properties make adapted signatures indistinguishable from fresh signatures for PRISM-RK and DeuringVUF-RK. Their unforgeability reduces to that of the underlying scheme as follows.

Given a forger $\mathcal{A}$ for the randomized scheme, build a forger for the original scheme that:

1. answers $\mathcal{A}$'s signing queries using the original signing oracle and adaptation;
2. "de-randomizes" $\mathcal{A}$'s forgery to a forgery for the original scheme using the pullback operation.

We detail this for PRISM-RK and DeuringVUF-RK in Sec. 5.2. Since this argument does not directly apply to SQIsign-RK, Sec. 5.3 gives an ad-hoc reduction. Finally, Sec. 5.4 shows that the mixing properties of isogeny graphs imply the anonymization of the long-term public keys, and hence the unlinkability of our schemes.

5.1 Signature adaptation

Both PRISM-RK and DeuringVUF-RK provide signature adaptation against unbounded adversaries; DeuringVUF-RK even achieves perfect adaptation.

Proposition 2. *In the random oracle model, PRISM-RK provides signature adaptation (Def. 6) against unbounded adversaries.*

Proof. By the definitions of RandPK and Adapt, the same randomizer rr gives the same update isogeny $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$ and signature domain E'_{pk} for both fresh and pushed-forward signatures. The same input $(\mathsf{msg}, \mathsf{salt})$ to $\mathsf{H}_{\mathsf{prime}}$ also gives the same $q \in \mathsf{Primes}_a$. Since GenIsogeny returns a uniform isogeny of given degree and codomain [5, Rem. 2], Fig. 7 reduces the claim to showing that, for a degree $d = q(2^a - q)$ and fixed $(E_{\mathsf{pk}}, E'_{\mathsf{pk}}, \phi_{\mathsf{rr}}, q)$, the distributions

$$\{[\phi_{\mathsf{rr}}]_*\psi \mid \psi \xleftarrow{\$} \mathsf{Isog}(E_{\mathsf{pk}}, q(2^a - q)\} \quad \text{and} \quad \{\psi' \mid \psi' \xleftarrow{\$} \mathsf{Isog}(E'_{\mathsf{pk}}, q(2^a - q))\}$$

are indistinguishable. For a separable isogeny $\phi_{\mathsf{rr}} : E \to E'$, the pushforward by ϕ_{rr} bijects the degree-d isogenies with domain E and those with domain E' whenever d is coprime to $\deg(\phi_{\mathsf{rr}})$.[16] The two distributions are therefore identical, even for unbounded adversaries. □

Proposition 3. *In the random oracle model,* DeuringVUF-RK *provides perfect signature adaptation (Def. 6) against unbounded adversaries controlling the randomness.*

Proof. For fixed rr and E_{pk}, let $\phi_{\mathsf{rr}} : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$ come from $\mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr})$ so the randomized public key is $(E'_{\mathsf{pk}}, \eta\phi_{\mathsf{rr}}(P_{\mathsf{pk}}), \eta\phi_{\mathsf{rr}}(Q_{\mathsf{pk}}))$. For $(r : s) \leftarrow \mathsf{H}_{\mathsf{VUF}}(\mathsf{msg})$, both fresh signing and adaptation output the unique isogeny with kernel $\langle r\phi_{\mathsf{rr}}(P_{\mathsf{pk}}) + s\phi_{\mathsf{rr}}(Q_{\mathsf{pk}})\rangle$; uniqueness follows because this kernel is cyclic and N is coprime to p [57, Prop. 4.12]. Hence $\mathcal{O}^b_{\mathsf{adapt}}(\mathsf{msg})$ has identical output for $b = 0$ and $b = 1$. □

5.2 Unforgeability for adaptable schemes

Unforgeability of PRISM-RK reduces to EUF-CMA security of PRISM (Prop. 4), while the same holds for DeuringVUF-RK and DeuringVUF (Prop. 5).

Proposition 4. *If* PRISM *satisfies EUF-CMA then* PRISM-RK *satisfies Def. 5,*

Proof. Given an adversary $\mathcal{A}_{\mathsf{rkuf}}$ against G^{rkuf} in Fig. 1, construct a PRISM EUF-CMA adversary $\mathcal{B}$ with access to a signing oracle Sign. On input E_{pk}, $\mathcal{B}$ forwards it to $\mathcal{A}_{\mathsf{rkuf}}$ as the long-term public key. For each signing query $(\mathsf{msg}, \mathsf{rr})$, $\mathcal{B}$:

1. Queries the signing oracle of PRISM to get $\sigma \leftarrow \mathsf{Sign}(\mathsf{msg})$.
2. If $\mathsf{rr} = \perp$, returns σ to $\mathcal{A}_{\mathsf{rkuf}}$.
3. If $\mathsf{rr} \neq \perp$, computes $\sigma' \leftarrow \mathsf{Adapt}(\mathsf{msg}, \sigma, \mathsf{rr}, E_{\mathsf{pk}})$ and returns σ' to $\mathcal{A}_{\mathsf{rkuf}}$.

When $\mathcal{A}_{\mathsf{rkuf}}$ outputs $(\mathsf{msg}^\bullet, \mathsf{rr}^\bullet, \sigma^\bullet)$, $\mathcal{B}$:

1. Uses $\mathsf{rr}^\bullet$ to compute $(\phi_{\mathsf{rr}^\bullet}, E^\bullet_{\mathsf{pk}}) \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr}^\bullet)$.
2. Computes the pullback $\sigma^* = [\phi_{\mathsf{rr}^\bullet}]^*\sigma^\bullet$.
3. Outputs $(\mathsf{msg}^\bullet, \sigma^*)$.

[16] Note that $q(2^a - q)$ is always odd.

The view of $\mathcal{A}_{\mathsf{rkuf}}$ is distributed as in G^{rkuf}: the long-term key is generated as in KeyGen, and by Prop. 2, adapted signatures have the same distribution as fresh PRISM-RK signatures of degree $d = q(2^a - q)$ with uniform salt.

If $\mathsf{Verify}(\mathsf{msg}^\bullet, \sigma^\bullet, E_{\mathsf{pk}}^\bullet)$ accepts, then $\sigma^\bullet$ represents a degree-d signature for $E_{\mathsf{pk}}^\bullet$. Since $\deg(\phi_{\mathsf{rr}\bullet})$ is coprime to d, the pullback $\sigma^* = [\phi_{\mathsf{rr}\bullet}]^*\sigma^\bullet$ is a valid PRISM signature on E_{pk}. All signing queries made by $\mathcal{A}_{\mathsf{rkuf}}$ were forwarded to Sign, so $(\mathsf{msg}^\bullet, \sigma^*)$ is a PRISM forgery exactly when $(\mathsf{msg}^\bullet, \mathsf{rr}^\bullet, \sigma^\bullet)$ is a PRISM-RK forgery, giving $\mathsf{Adv}^{\mathcal{A}_{\mathsf{rkuf}}}_{\mathsf{rkuf}} = \mathsf{Adv}^{\mathcal{B}}_{\mathsf{EUF\text{-}CMA}}$. □

Using similar arguments to Prop. 4, we obtain the same result for DeuringVUF-RK. We postpone the proof to App. A.1.

Proposition 5. DeuringVUF-RK *satisfies the signature unforgeability property (Def. 5), assuming* DeuringVUF *satisfies EUF-CMA.*

5.3 Unforgeability of SQIsign-RK

As in [8], we use the idealized isogeny oracles Defs. 8 and 9. They let us simulate SQIsign-RK signing queries and reduce unforgeability to finding a non-scalar endomorphism of the public curve. For completeness, App. B gives a more formal hint-based proof using only non-interactive assumptions, following [3].

Signing queries simulation. Similar to [8, Thm. 22], we can use the UTO and FIDIO oracles to simulate signing queries.

Corollary 1 (of Theorem 22 in [8]). *In the random oracle model, there exists a polynomial time simulator S with access to a* UTO *and a* FIDIO *that produces signatures which are statistically indistinguishable from honest signatures.*

Proof. Let S' be the simulator from [8, Thm. 22], which uses a UTO and a FIDIO to produce statistically indistinguishable transcripts for the SQIsign identification protocol. On signing query $(\mathsf{msg}, \mathsf{rr})$, S computes the randomized public key $E'_{\mathsf{pk}} \leftarrow \mathsf{RandPK}(E_{\mathsf{pk}}, \mathsf{rr})$, runs S' on E'_{pk} to get $(E_{\mathsf{com}}, \phi_{\mathsf{chall}}, \sigma_{\mathsf{resp}})$, and reprograms the random oracle so that $\mathsf{H}_{\mathsf{SQI}}(E'_{\mathsf{pk}}, E_{\mathsf{com}}, \mathsf{msg}) = \phi_{\mathsf{chall}}$ and returns the signature $\sigma = (\phi_{\mathsf{chall}}, \sigma_{\mathsf{resp}})$. Since the commitment curve has negligible min-entropy by *e.g.* [3, Lem. 4.1], the random oracle reprogramming only incurs a negligible additive loss in the distinguishing advantage. □

Remark 3. SQIsign signing requires the response isogeny to be uniformly random among isogenies with the same codomain and bounded degree, a property not preserved by pushforwards [3, Prob. 3]. The UTO oracle hides this technical issue; while the proof in App. B addresses it explicitly.

From forgery to endomorphism. For SQIsign-RK, de-randomizing a forgery on E'_{pk} does not directly de-randomize the commitment curve, whose endomorphism ring is unknown at that point of the reduction. Thus, we prefer instead to rewind the forger and extract directly a non-scalar endomorphism of the public curve.

For tightness, we use the Multi-Instance Reset Lemma (Lem. 2 in App. B) from [43], a generalization of the Forking Lemma [10]. It runs the adversary on N independent instances, here randomized public keys sampled statistically close to the stationary distribution by Thm. 1. If the forger succeeds with non-negligible probability, rewinding yields two accepting transcripts with non-negligible probability.

Theorem 2. *In the random-oracle,* UTO*, and* FIDIO *model, if Prob. 1 is hard, then* SQIsign-RK *satisfies the signature unforgeability property (Def. 5).*

Proof. We transform an adversary $\mathcal{A}$ against SQIsign-RK unforgeability into an adversary against Prob. 1 with input the curve E_{pk}.

Let $t_{\mathcal{A}}$ be the running time and $\varepsilon_{\mathcal{A}}$ the success probability of $\mathcal{A}$ and ChSet the codomain of the random oracle $\mathsf{H}_{\mathsf{SQI}}$. Let N_{sign} be the upper bound on the number of random-oracle queries performed by $\mathcal{A}$.

Simulation of the signing oracle. By Cor. 1, a simulator S' answers SQIsign-RK signing queries using a UTO and a FIDIO. Game $\mathcal{B}$ forwards the input public key E_{pk} to $\mathcal{A}$, answers signing queries with S', and returns $\mathcal{A}$'s forgery. Since the commitment has negligible min-entropy ([3, Lem. 4.1]) and the signatures returned by S' are statistically indistinguishable from honest signatures, we have $\mathsf{Adv}^{\mathcal{B}} \approx \mathsf{Adv}^{\mathcal{A}}$ and $t_{\mathcal{B}} \approx t_{\mathcal{A}}$.

Random oracle guessing. Game $\mathcal{C}$ takes E_{pk} and a random string $h \in \mathsf{ChSet}$, forwards E_{pk} to $\mathcal{B}$, and controls the random oracle as $\mathcal{B}$ does, except that one random oracle query is guessed and reprogrammed to h. If $\mathcal{B}$'s forgery is invalid or does not use the guessed query, $\mathcal{C}$ outputs $(0, \perp)$; otherwise it returns 1 and the expanded forgery $(E'_{\mathsf{pk}}, E_{\mathsf{com}}), (\phi_{\mathsf{chall}}, \sigma_{\mathsf{resp}}, \mathsf{rr})$. Note that E'_{pk} and h uniquely determine the challenge isogeny ϕ_{chall}.

The running time is $t_{\mathcal{C}} = t_{\mathcal{B}}$. Since $\mathcal{C}$ makes no additional random oracle queries and the guessed index is independent of E_{pk} and $\mathcal{B}$'s randomness, $\mathsf{Adv}^{\mathcal{C}} \geq \mathsf{Adv}^{\mathcal{B}}/q$.

Rewinding and endomorphism extraction. Apply Lem. 2 to $\mathcal{C}$ to obtain $\mathsf{Rwd}_{\mathcal{C}}$, and proceed as in [43, Lem. 3.5]. For $N \geq 1$, define $\mathcal{E}$ to run $\mathcal{C}$ through $\mathsf{Rwd}_{\mathcal{C}}$ on N randomized public keys:

- For each $i \in [1, N]$, sample an isogeny walk $\tau_i : E_{\mathsf{pk}} \to E^{(i)}_{\mathsf{pk}}$ of degree $d = \ell^k$ large enough that $E^{(i)}_{\mathsf{pk}}$ is statistically close to stationary by Thm. 1.
- Run $\mathsf{Rwd}_{\mathcal{C}}$ on $E^{(1)}_{\mathsf{pk}}, \cdots, E^{(N)}_{\mathsf{pk}}$, obtaining either $\perp$ or two valid SQIsign transcripts $(\mathsf{tr}^{(1)}, \mathsf{tr}^{(2)})$ with the same random oracle input.

If the output is not $\perp$, parse each transcript $\mathsf{tr}^{(i)}$ as $(E'^{(i)}_{\mathsf{pk}}, E^{(i)}_{\mathsf{com}}), (\phi^{(i)}_{\mathsf{chall}}, \sigma^{(i)}_{\mathsf{resp}}, \mathsf{rr}^{(i)})$ for $i \in \{1, 2\}$. Since the transcripts have the same random oracle input, they correspond to the same randomized curve $E'^{(1)}_{\mathsf{pk}} = E'^{(2)}_{\mathsf{pk}}$ and to the same commitment curve $E^{(1)}_{\mathsf{com}} = E^{(2)}_{\mathsf{com}}$, but they have different challenge isogenies $\phi^{(1)}_{\mathsf{chall}} \neq \phi^{(2)}_{\mathsf{chall}}$. The 2-special soundness of SQIsign then extracts a non-scalar endomorphism α of $E'^{(1)}_{\mathsf{pk}}$ [8, Thm. 19]. By the correctness conditions on the randomness, we

have an isogeny $\phi : E_{\mathsf{pk}}^{(i^*)} \to E_{\mathsf{pk}}^{\prime(1)}$ of degree d_r. Also, by definition we have an isogeny $\tau_{i^*} : E_{\mathsf{pk}} \to E_{\mathsf{pk}}^{(i^*)}$ of degree d. Let $\eta := \phi \circ \tau_{i^*}$. The lollipop endomorphism $\alpha' = \widehat{\eta} \circ \alpha \circ \eta$ is a non-scalar endomorphism of E_{pk}, thus solving Prob. 1. By contradiction, assume that $\varphi \circ \alpha \circ \widehat{\varphi} = [m]$, i.e. is a scalar endomorphism; this would imply that $m^2 = \deg(\varphi)^2 \cdot \deg(\alpha)$. Hence $\varphi \circ \alpha = [m/\deg(\varphi)]\varphi$, so α is scalar, a contradiction.
By Lem. 2, the probability that $i^* \geq 1$ is at least

$$\mathsf{Adv}^{\mathsf{Rwd}_{\mathcal{C}}} \geq \left(1 - \left(1 - \mathsf{Adv}^{\mathcal{C}} + \frac{1}{\mathsf{ChSet}}\right)^{N}\right)^2 \tag{4}$$

and the running time of $\mathsf{Rwd}_{\mathcal{C}}$ is $t_{\mathsf{Rwd}_{\mathcal{C}}} \approx 2Nt_{\mathcal{C}}$. Choosing $N = (\mathsf{Adv}^{\mathcal{C}} - 1/|\mathsf{ChSet}|)^{-1}$ gives $6\mathsf{Adv}^{\mathsf{Rwd}_{\mathcal{C}}}/t_{\mathsf{Rwd}_{\mathcal{C}}} \approx \mathsf{Adv}^{\mathcal{C}}/t_{\mathcal{C}}$, giving a tight reduction to Prob. 1. □

5.4 Unlinkability

Unlinkability (Def. 7) asks whether an adversary can link a randomized public key pk' to the long-term public key pk, even after querying signatures on pk'.

SQISign-RK and PRISM-RK Recall that key randomization for SQIsign-RK and PRISM-RK in Fig. 5 is done via a randomizing isogeny ϕ_{rr} of degree $2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$. By the mixing properties of the ℓ-isogeny graph, if $d_r = 2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$ is large enough, the distribution of pk' becomes independent of the long-term key pk, even if the latter is adversarially chosen.

Proposition 6. *Let the randomizing isogeny be of degree $d_r = 2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$. If $u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}} \geq \log p + 2\lambda + \log(u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}})$, then SQIsign-RK and PRISM-RK provide statistical unlinkability against unbounded adversaries (Def. 7).*

Proof. After receiving $(\mathsf{sk}_0, \mathsf{pk}_0, \mathsf{sk}_1, \mathsf{pk}_1)$, sample $b \leftarrow \{0,1\}$ and rr, set $\mathsf{sk}' \leftarrow \mathsf{RandSK}(\mathsf{sk}_b, \mathsf{pk}_b, \mathsf{rr})$ and $\mathsf{pk}' \leftarrow \mathsf{RandPK}(\mathsf{pk}_b, \mathsf{rr})$, and send pk' to the adversary. Signing queries are answered using sk'; for an unbounded adversary, this oracle adds no information since sk' can be found by exhaustive search.

Let π_i be the probability distribution of pk_i output by the adversary. Let π_{ik} be the distribution of pk_i after k steps of the random walk given by ϕ_{rr}, a uniformly generated cyclic isogeny of degree $d_r = 2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$. Under the degree condition on k, Thm. 1 implies that $d_{TV}(\pi_{ik}, s) \leq 2^{-\lambda-1}$ for both $i = 0, 1$, so both $(\pi_i)_\ell^{(k)}$ are statistically indistinguishable from the stationary distribution s on Ell_p. This immediately implies that $d_{TV}(\pi_{1k}, \pi_{2k}) \leq 2^{-\lambda}$, thus even for an unbounded adversary $\mathcal{A}$, we find $\mathsf{Adv}_{\mathsf{unl}}^{\mathcal{A}} \leq 2^{-\lambda}$. □

Unlinkability of DeuringVUF. For DeuringVUF-RK, the public key includes a curve and a basis of its N-torsion, obtained by applying ϕ_{rr} to a fixed basis on the original public key, composed with a scalar multiplication. De-anonymization is therefore a variant of the *Decisional SIDH isogeny (DSIDH) problem* [30]: for a supersingular curve E and basis P, Q of $E[N]$, distinguish

1. $(E', \phi(P), \phi(Q))$, where $\phi : E \to E'$ is a uniformly random cyclic d-isogeny;
2. (E', P', Q'), where E' is sampled from the stationary distribution and P', Q' is a random basis of its N-torsion, such that $e_N(P', Q') = e_N(P, Q)^d$.

When N is powersmooth and $d \leq N$, the SIDH attacks solve this problem efficiently [52,46,16]. However, longer isogeny walks make the two distributions indistinguishable by the mixing properties of the ℓ-isogeny graph with full level structure [31], generalizing Thm. 1.

Theorem 3. *Let p, N, ℓ be distinct primes with ℓ a square mod N and $N \geq 3$, $k > 0$ an integer and $\eta \in \mathbb{Z}/N\mathbb{Z}$ a square root of ℓ^{-k} modulo N. Then, given any supersingular elliptic curve E and basis P, Q of $E[N]$, we have that:*

- *the distribution π_k of $(E', \eta\phi(P), \eta\phi(Q))$ for $\phi : E \to E'$ a random cyclic isogeny of degree ℓ^k;*
- *the stationary distribution ν of the set of curves and points of N-torsion (E', P', Q') such that $e_N(P', Q') = e_N(P, Q)$, where e_N is the Weil pairing;*

have total variation distance at most

$$d_{TV}(\nu, \pi_k) \leq \sqrt{\frac{pN^3}{\ell^k}} \cdot (k+1) \ . \tag{5}$$

Proof (sketch). This theorem applies the results of [21] and [48], once translated to isogeny graphs with full level structure. The full proof is in App. A.2.

The main intuition is to use the spectral gap of the ℓ-isogeny graph with full level structure: a graph whose vertices are the triples (E', P', Q') such that E' is a supersingular elliptic curve and P', Q' is a basis of $E'[N]$, while edges are given by ℓ-isogenies $\phi : E' \to E''$ such that $\phi(P') = P''$ and $\phi(Q') = Q''$. They are extensively studied in [21], where the authors show that the adjacency operator of this graph can be diagonalized and its eigenvalues can be bounded using the theory of automorphic forms. They also show that the only obstruction to the connectivity of the graph is given by the Weil pairing, so from the multiplicative order k' of ℓ modulo N we can deduce that the graph has exactly $(N-1)/k'$ connected components, with each having a k'-multipartite structure.

We remove these obstructions by including a multiplication by η in the random walk, which is a bijection between each of the different multipartitions of the graph. To account for this *ad-hoc* modification, we use the more versatile functorial definitions and results from [48]. We also specialise our analysis to the non-backtracking random walk, giving a recursive definition of the corresponding adjacency operator and bounding its eigenvalues using the previous step and classical techniques on Chebyshev polynomials.

We conclude using spectral gap to bound the total variation distance between the distribution of the random walk and the stationary distribution. □

For $\ell = 2$, Thm. 3 gives a concrete bound $k \geq 2\lambda + \log p + 3\log N + \log(k)$ such that randomized public keys are statistically indistinguishable from random ones, independently of the long-term public key. Using this bound we can claim:

Corollary 2 (of Thm. 3). *Let the randomizing isogeny be of degree* $d_r = 2^{u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}}}$*, if* $u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}} \geq 2\lambda + \log p + 3\log N + \log(u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}})$ DeuringVUF-RK *satisfies unlinkability against unbounded adversaries (Def. 7).*

Proof (Sketch). The proof is completely analogous to the proof of Prop. 6. We emphasize that here it is crucial to consider a graph without obstructions. Without the multiplication by η and the check in VerKey that the Weil pairing of the input basis is correct, an adversary could easily link randomized public keys by giving as input two public keys in two different multipartitions of the graph, that can easily be distinguished with the Weil pairing.

6 Instantiation and implementation

We implemented SQIsign-RK, PRISM-RK and DeuringVUF-RK in Python/SageMath on top of the most recent public implementation of SQIsign and PRISM[17] based on the Qlapoti algorithm [12] to translate ideals into isogenies and the parameters from [2, Ch. 5] and [5, Tab. 2] for all NIST security levels. To our knowledge, this is the first implementation of DeuringVUF-RK in Python/SageMath. For the latter, to satisfy the requirements of Alg. 2 for the Adapt algorithm, we selected new parameters in App. A.3. We provide the code at

https://github.com/giacomoborin/isogeny_swrk

In Tab. 4, we compare the key randomization procedures (RandPK, RandSK) to the key generation in terms of execution time and the signature adaptation of PRISM-RK (Adapt) to the signing procedure. We observe that the public key randomization (RandPK) is relatively efficient, while the secret key randomization (RandSK) is significantly slower. Indeed, while the computation of a (long) chain of 2-isogenies (in dimension 1) in RandPK is fast in practice, the call to IsoToEnd in RandSK involves several calls to the Qlapoti algorithm involving $(2,2)$-isogeny computations (in dimension 2), which cost more (see Appendix C). Similarly, Adapt in PRISM-RK is around 15 times slower than the signing procedure, due to the large number of pushforwards needed in Alg. 2 (resp. 9, 7, and 6 at NIST levels I, III and V), that each require a $(2,2)$-isogeny chain computation. To demonstrate this, in Tab. 3 we count the number of 2-isogenies and $(2,2)$-isogenies (which are the dominant operations) involved in each procedure and NIST security level. The implementation of DeuringVUF-RK is still very preliminary and significantly slower than the other two schemes, as shown by the larger relative costs in Tab. 4. This is mainly due to having to use a non-optimized $\mathbb{F}_{p^4}$ arithmetic: several relevant subroutines like optimized basis sampling and x-only isogeny computations are currently performed over $\mathbb{F}_{p^4}$ rather than on the twisted curve. This implementation primarily shows the feasibility of the construction, and could be significantly improved by optimising underlying subroutines over the twisted curve. We leave this for future work. As for PRISM-RK, Adapt is greatly affected by the number of pushforwards, which for this parameter set is 52.

[17] https://github.com/KULeuven-COSIC/sqisign_prism_v2

Note that the number of pushforwards (for PRISM-RK given by $r_{\mathsf{rand}} = \lceil 2\log_2(p)/u_{\mathsf{cmp}} \rceil$ with $u_{\mathsf{cmp}} = f - a$) in Adapt could greatly be reduced by increasing the parameter f (i.e., the 2-valuation of $p+1$) and keeping a fixed, at the expense of a slower $\mathbb{F}_{p^2}$-arithmetic. This would be particularly beneficial for DeuringVUF-RK, where the number of pushforwards greatly affects the running time of adaption. We leave this for future work, since a proper selection of parameters for DeuringVUF-RK would require first more optimised subroutines.

Table 3: Number of isogenies computed of each type for each NIST security level. Parentheses indicate that they may vary slightly.

Security level		NIST-I		NIST-III		NIST-V	
Isogeny type		2	$(2,2)$	2	$(2,2)$	2	$(2,2)$
SQIsign-RK	KeyGen	-	248	-	376	-	500
	Sign	(248)	(622)	(376)	(944)	(500)	(1253)
	Verify	(248)	(126)	(376)	(192)	(500)	(253)
	RandPK	496	-	752	-	1000	-
	RandSK	496	496	752	752	1000	1000
PRISM-RK	KeyGen	-	248	-	376	-	500
	Sign	-	248	-	376	-	500
	Verify	-	192	-	256	-	320
	RandPK	504	-	840	-	1080	-
	RandSK	504	744	840	1128	1080	1500
	Adapt	1008	1728	1680	1792	2160	1920

Table 4: Relative running times of SQIsign-RK, PRISM-RK, DeuringVUF-RK additional features to update keys (RandPK, RandSK) and signatures (Adapt) for each NIST security level. Results were obtained with Sage 10.8 using Python 3.13.7 on an Apple M3 Pro running MacOS Sonoma 14.3.

Scheme	SQIsign-RK			PRISM-RK			DeuringVUF-RK
Security level	NIST-I	NIST-III	NIST-V	NIST-I	NIST-III	NIST-V	NIST-I
RandPK/KeyGen	0.41	0.39	0.40	0.93	0.91	0.91	10.94
RandSK/KeyGen	2.51	2.28	2.27	4.99	6.54	7.78	15.96
Adapt/Sign	-	-	-	17.01	16.585	15.62	88.92

Acknowledgments

Robi Pedersen was supported by the research grant VIL53029 from VILLUM FONDEN. Maria Corte-Real Santos was supported by the European Research Council under grant No. 101116169 (AGATHA CRYPTY). Pierrick Dartois was supported by the France 2030 program under grant ANR-22-PETQ-0008 (PQ-TLS) and BPI France project RESQUE. Riccardo Invernizzi is supported by the European Research Council under grant No. 101020788 (ISOCRYPT), by the Research Council KU Leuven grant C14/ 24/099, by Cybersecurity Research

Flanders with reference number VR20192203, and by Research Foundation – Flanders (FWO) under a PhD Fellowship fundamental research (project number 1138925N). Luciano Maino was partially supported by EPSRC through grant number EP/V011324/1. Michel Seck was partially supported by the French National Research Organization CNRS through the CNRS-Afrique project DIALS. Andrea Basso and Giacomo Borin are also supported by SNSF Consolidator Grant CryptonIs 213766. The authors deeply thank Guido Maria Lido for insightful discussions on the structure of isogeny graphs with level structures.

References

1. Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.
2. Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign: Algorithm specifications and supporting documentation. Technical report, National Institute of Standards and Technology, 2025.
3. Marius A. Aardal, Andrea Basso, Luca De Feo, Sikhar Patranabis, and Benjamin Wesolowski. A complete security proof of SQIsign. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part VI*, volume 16005 of *Lecture Notes in Computer Science*, pages 190–222, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.
4. Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(04):585–603, 2007.
5. Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM: Simple and compact identification and signatures from large prime degree isogenies. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part III*, volume 15676 of *Lecture Notes in Computer Science*, pages 300–332, Røros, Norway, May 12–15, 2025. Springer, Cham, Switzerland.
6. Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM with a pinch of salt: Simple, efficient and strongly unforgeable signatures from isogenies. Cryptology ePrint Archive, Paper 2026/443, 2026.

7. Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part II*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.
8. 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 Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part III*, volume 15486 of *Lecture Notes in Computer Science*, pages 339–370, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore.
9. Andrea Basso and Luciano Maino. POKÉ: A compact and efficient PKE from higher-dimensional isogenies. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part II*, volume 15602 of *Lecture Notes in Computer Science*, pages 94–123, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.
10. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006: 13th Conference on Computer and Communications Security*, pages 390–399, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press.
11. Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland.
12. Giacomo Borin, Maria Corte-Real Santos, Jonathan Komada Eriksen, Riccardo Invernizzi, Marzio Mula, Sina Schaeffler, and Frederik Vercauteren. Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part IV*, volume 16248 of *Lecture Notes in Computer Science*, pages 174–205, Melbourne, VIC, Australia, December 8–12, 2025. Springer, Singapore, Singapore.
13. Giacomo Borin, Luca De Feo, Guido Maria Lido, and Sina Schaeffler. The SQInstructor: a guide to SQIsign and the deuring correspondence with level structures. Cryptology ePrint Archive, Paper 2026/493, 2026.
14. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289, Amalfi, Italy, September 12–13, 2003. Springer Berlin Heidelberg, Germany.
15. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72, Santa Barbara, CA, USA, August 15–19, 2004. Springer Berlin Heidelberg, Germany.
16. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

17. Sofía Celi, Scott Griffy, Lucjan Hanzlik, Octavio Perez-Kempner, and Daniel Slamanig. Sok: Signatures with randomizable keys. In *International Conference on Financial Cryptography and Data Security*, pages 160–187. Springer, 2024.
18. Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials. In Anupam Datta and Cedric Fournet, editors, *CSF 2014: IEEE 27th Computer Security Foundations Symposium*, pages 199–213, Vienna, Austria, July 19–22, 2014. IEEE Computer Society Press.
19. Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, 12(3):349–368, September 2022.
20. 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. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.
21. Giulio Codogni and Guido Lido. Spectral theory of isogeny graphs, 2025.
22. Graeme Connell, Sebastian Faller, Felix Günther, Julia Hesse, Vadim Lyubashevsky, and Rolfe Schmidt. A quantum-safe private group system for signal from key re-randomizable signatures. Cryptology ePrint Archive, Paper 2026/453, 2026.
23. Aisling Connolly, Pascal Lafourcade, and Octavio Perez-Kempner. Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022: 25th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 13177 of *Lecture Notes in Computer Science*, pages 409–438, Virtual Event, March 8–11, 2022. Springer, Cham, Switzerland.
24. Elizabeth C. Crites and Anna Lysyanskaya. Delegatable anonymous credentials from mercurial signatures. In Mitsuru Matsui, editor, *Topics in Cryptology – CT-RSA 2019*, volume 11405 of *Lecture Notes in Computer Science*, pages 535–555, San Francisco, CA, USA, March 4–8, 2019. Springer, Cham, Switzerland.
25. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: New dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436, 2023.
26. Poulami Das, Andreas Erwig, Michael Meyer, and Patrick Struck. Efficient post-quantum secure deterministic threshold wallets from isogenies. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 522–532, 2024.
27. Poulami Das, Sebastian Faust, and Julian Loss. A formal treatment of deterministic wallets. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 651–668, New York, NY, USA, 2019. Association for Computing Machinery.
28. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *Proceedings on Privacy Enhancing Technologies*, 2018(3):164–180, July 2018.
29. Luca De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 2017.
30. Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology*

– *ASIACRYPT 2022, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 310–339, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland.
31. Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny. Isogeny problems with level structure. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part VII*, volume 14657 of *Lecture Notes in Computer Science*, pages 181–204, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.
32. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland.
33. Thomas den Hollander, Shai Levin, Marzio Mula, Robi Pedersen, Daniel Slamanig, and Sebastian A. Spindler. SPRINT: New isogeny proofs of knowledge and isogeny-based signatures. Cryptology ePrint Archive, Paper 2026/364, 2026.
34. 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.
35. Edward Eaton, Tancrède Lepoint, and Christopher A. Wood. Security analysis of signature schemes with key blinding. Cryptology ePrint Archive, Report 2023/380, 2023.
36. Edward Eaton, Sajin Sasy, and Ian Goldberg. Improving the privacy of tor onion services. In *International Conference on Applied Cryptography and Network Security*, volume 13269, pages 273–292. Springer, 2022.
37. Edward Eaton, Douglas Stebila, and Roy Stracovsky. Post-quantum key-blinding for authentication in anonymity networks. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021: 7th International Conference on Cryptology and Information Security in Latin America*, volume 12912 of *Lecture Notes in Computer Science*, pages 67–87, Bogotá, Colombia, October 6–8, 2021. Springer, Cham, Switzerland.
38. Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 329–368, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland.
39. Nils Fleischhacker, Johannes Krupp, Giulio Malavolta, Jonas Schneider, Dominique Schröder, and Mark Simkin. Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 301–330, Taipei, Taiwan, March 6–9, 2016. Springer Berlin Heidelberg, Germany.
40. Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 3–33, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland.
41. Tobias Guggemos and Farzin Renan. Key-updatable identity-based signature schemes. In Ratna Dutta, Luca De Feo, and Sugata Gangopadhyay, editors, *Progress*

in Cryptology - INDOCRYPT 2025: 26th International Conference in Cryptology in India, volume 16372 of *Lecture Notes in Computer Science*, pages 216–238, Bhubaneshwar, India, December 14–17, 2025. Springer, Cham, Switzerland.
42. Ernst Kani. The existence of curves of genus two with elliptic differentials. *Journal of Number Theory*, 64(1):130–161, 1997.
43. Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 33–61, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany.
44. Antonin Leroux. Verifiable random function from the deuring correspondence and higher dimensional isogenies. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VII*, volume 15607 of *Lecture Notes in Computer Science*, pages 167–194, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.
45. Antonin Leroux and Maxime Roméas. Compact, efficient and CCA-secure updatable encryption from isogenies. Cryptology ePrint Archive, Report 2025/1853, 2025.
46. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.
47. Tomoki Moriya. LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram. Cryptology ePrint Archive, Report 2024/521, 2024.
48. Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 388–417, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.
49. Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland.
50. David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *Topics in Cryptology – CT-RSA 2016*, volume 9610 of *Lecture Notes in Computer Science*, pages 111–126, San Francisco, CA, USA, February 29 – March 4, 2016. Springer, Cham, Switzerland.
51. Sihang Pu, Sri Aravinda Krishnan Thyagarajan, Nico Döttling, and Lucjan Hanzlik. Post quantum fuzzy stealth signatures and applications. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023: 30th Conference on Computer and Communications Security*, pages 371–385, Copenhagen, Denmark, November 26–30, 2023. ACM Press.
52. Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.
53. Damien Robert. On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Report 2024/1071, 2024.
54. Michel Seck and Adeline Roux-Langlois. Towards post-quantum bitcoin blockchain using dilithium signature. *IACR Communications in Cryptology*, 2(3), 2025.

55. Surbhi Shaw and Ratna Dutta. Compact stateful deterministic wallet from isogeny-based signature featuring uniquely rerandomizable public keys. In Jing Deng, Vladimir Kolesnikov, and Alexander A. Schwarzmann, editors, *CANS 23: 22nd International Conference on Cryptology and Network Security*, volume 14342 of *Lecture Notes in Computer Science*, pages 392–413, Augusta, GA, USA, October 31 – November 2, 2023. Springer, Singapore, Singapore.
56. Surbhi Shaw and Ratna Dutta. Post-quantum secure compact deterministic wallets from isogeny-based signatures with rerandomized keys. *Theoretical Computer Science*, 1035:115127, 2025.
57. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.
58. Tor. Tor directory protocol, version 3. https://spec.torproject.org/rend-spec/keyblinding-scheme.html.

A Additional material for DeuringVUF-RK

A.1 Unforgeability of DeuringVUF-RK

Proposition 5. DeuringVUF-RK *satisfies the signature unforgeability property (Def. 5), assuming* DeuringVUF *satisfies* EUF-CMA.

Proof (of Prop. 5). We define $\mathcal{B}$ as an adversary against Prob. 3 with oracle access to both a FIXDIO and to an adversary $\mathcal{A}_{\mathsf{rkuf}}$ against G^{rkuf} from Fig. 1. On input E_{pk}, $\mathcal{B}$ samples a random basis $P_{\mathsf{pk}}, Q_{\mathsf{pk}}$ of $E_{\mathsf{pk}}[N]$ and forwards $(E_{\mathsf{pk}}, P_{\mathsf{pk}}, Q_{\mathsf{pk}})$ to $\mathcal{A}_{\mathsf{rkuf}}$ as the long-term public key. Now, whenever $\mathcal{A}_{\mathsf{rkuf}}$ queries the Sign' oracle on $(\mathsf{msg}, \mathsf{rr})$, $\mathcal{B}$ proceeds as follows.

1. Query $(r : s) \leftarrow \mathsf{H}_{\mathsf{VUF}}(\mathsf{msg})$ and compute $K = rP_{\mathsf{pk}} + sQ_{\mathsf{pk}}$.
2. Query the FIXDIO on K to receive an N-isogeny $\sigma : E_{\mathsf{pk}} \to E_\sigma = E_{\mathsf{pk}}/\langle K \rangle$.
3. If $\mathsf{rr} = \perp$, return σ to $\mathcal{A}_{\mathsf{rkuf}}$.
4. If $\mathsf{rr} \neq \perp$, compute and return $\sigma' \leftarrow \mathsf{Adapt}(\mathsf{msg}, \sigma, \mathsf{rr}, E_{\mathsf{pk}})$.

Then on output $(\mathsf{msg}^\bullet, \mathsf{rr}^\bullet, \sigma^\bullet)$ by $\mathcal{A}_{\mathsf{rkuf}}$, $\mathcal{B}$ proceeds as follows:

1. Use $\mathsf{rr}^\bullet$ to compute $(\phi_{\mathsf{rr}^\bullet}, E^\bullet_{\mathsf{pk}}) \leftarrow \mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr}^\bullet)$.
2. Compute the pullback $\sigma^* = [\phi_{\mathsf{rr}^\bullet}]^* \sigma^\bullet$.
3. Output $(\mathsf{msg}^\bullet, \sigma^*)$.

We again show that the inputs that $\mathcal{A}_{\mathsf{rkuf}}$ receives are distributed equally to G^{rkuf}. First note both the public key curve in Prob. 3 as well as in G^{rkuf} are uniformly random over Ell_p. In the latter this is achieved by using a large enough prime power ℓ_{large} during key generation as discussed in Sec. 3.4. Further, $\mathcal{B}$ samples the basis $P_{\mathsf{pk}}, Q_{\mathsf{pk}}$ uniformly at random, while in KeyGen, the secret matrix M_{sk} uniformizes the maps of the special points P_0, Q_0, rendering both descriptions equivalent. Regarding signatures, it is easy to see that due to the the deterministic nature of signing in DeuringVUF-RK, any signature is uniquely determined by the parameters r and s. Thus, any fresh signature on E'_{pk} or pushforward from E_{pk} will result in the exact same signature with kernel $\langle r\phi_{\mathsf{rr}}(P_{\mathsf{pk}}) + s\phi_{\mathsf{rr}}(Q_{\mathsf{pk}}) \rangle$, making both perfectly indistinguishable.

Finally, due to the coprimality of N and d_r, $(\mathsf{msg}^\bullet, \sigma^*)$ is a valid forgery on E_{pk} exactly if $(\mathsf{msg}^\bullet, \mathsf{rr}^\bullet, \sigma^\bullet)$ is a valid signature on $E^\bullet_{\mathsf{pk}}$ and we find $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{rkuf}} = \mathsf{Adv}^{\mathcal{B}}_{\mathsf{EUF\text{-}CMA}}$. □

A.2 Complete proof of Thm. 3

Theorem 3. *Let p, N, ℓ be distinct primes with ℓ a square mod N and $N \geq 3$, $k > 0$ an integer and $\eta \in \mathbb{Z}/N\mathbb{Z}$ a square root of ℓ^{-k} modulo N. Then, given any supersingular elliptic curve E and basis P, Q of $E[N]$, we have that:*

- *the distribution π_k of $(E', \eta\phi(P), \eta\phi(Q))$ for $\phi : E \to E'$ a random cyclic isogeny of degree ℓ^k;*
- *the stationary distribution ν of the set of curves and points of N-torsion (E', P', Q') such that $e_N(P', Q') = e_N(P, Q)$, where e_N is the Weil pairing;*

have total variation distance at most

$$d_{TV}(\nu, \pi_k) \leq \sqrt{\frac{pN^3}{\ell^k}} \cdot (k+1) \ . \tag{5}$$

For our proof we fix the following notation, inspired from [48].

- Let $\zeta \in \mathbb{Z}/N\mathbb{Z}$ be the square root of ℓ^{-1} modulo N such that $\eta = \zeta^k$.
- Let $SS_\ell(p)$ be the category of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ and ℓ-isogenies between them, i.e. isogenies of degree a power of ℓ.
- Given a set X we denote by $L^2(X)$ the space of complex functions over X of finite L^2-norm, i.e. the space of complex functions f such that $\|f\|_2^2 := \sum_{x \in X} |f(x)|^2 < \infty$.
- Recall also that we fixed a supersingular elliptic curve E and a basis P, Q of $E[N]$. Then e_N is the Weil pairing, i.e. a bilinear, non-degenerate and alternating map from $E[N] \times E[N]$ to the group of N-th roots of unity.

Defining the isogeny graph with full level structure. We refer to [48] for the general definitions and results about categories of elements. For our proof we consider the functor

$$\mathcal{F}_N : SS_\ell(p) \to \text{Set}$$

such that:

- For any supersingular elliptic curve E' the set $\mathcal{F}_N(E')$ contains the pairs (P', Q') where P', Q' is a basis of $E'[N]$ such that $e_N(P', Q') = e_N(P, Q)$.
- Given any isogeny $\phi : E' \to E''$ of degree ℓ^r the map $\mathcal{F}_N(\phi)$ sends $(P', Q') \in \mathcal{F}_N(E')$ to $(\zeta^r\phi(P'), \zeta^r\phi(Q')) \in \mathcal{F}_N(E'')$.

It is immediate to verify that $\mathcal{F}_N$ defines a category of elements $\text{El}(\mathcal{F}_N)$ in the sense of [48, Def. 2.8]. Also, $\mathcal{F}_N$ is a functor with the $(\text{mod}\, N)$-congruence property ([48, Def. 3.7]). Namely, this means that given any supersingular elliptic curve E and two endomorphisms ϕ, ψ of degree a power of ℓ such that $\phi - \psi \in N \cdot \text{End}(E)$, then $\mathcal{F}_N(\phi) = \mathcal{F}_N(\psi)$.

We define the graph $\mathcal{G}$ associated to $\mathcal{F}_N$ according to [48, Def. 3.4].

- The vertices of $\mathcal{G}$ are the pairs (E, P, Q) where E is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ and P, Q is a basis of $E[N]$ such that $e(P, Q) = e(P_0, Q_0)$.

– The edges of $\mathcal{G}$ are ℓ^r-isogenies acting on the vertices as ϕ composed with multiplication by ζ^r.

In the graph there are at most $\frac{p-1}{24}(N^2-1)(N-1)$ vertices, since for each of the $\frac{p-1}{24}$ supersingular elliptic curves we can consider all the basis given by a change of basis of discriminant $1 \pmod N$. We also define as $\mathrm{Aut}(E,P,Q)$ the group of automorphisms of the vertex (E,P,Q), i.e. the group of automorphisms α of E such that $\alpha(P)=P$ and $\alpha(Q)=Q$. Note: even if (E,P,Q) is isomorphic to $(E,-P,-Q)$, still the multiplication by -1 is not an automorphism of (E,P,Q).
Spectral decomposition of $\mathcal{G}$. Let A_ℓ be the adjacency operator of the ℓ-isogeny graph $\mathcal{G}$, mapping a vertex (E,P,Q) to the sum of its ℓ-isogenous neighbors, according to the edges defined above by the functor $\mathcal{F}_N$. By [48, Thm. 3.10] we have the following:

1. The adjacency operator is a normal operator on $L^2(\mathcal{G})$, i.e. it admits a spectral decomposition.
2. The operator norm of A_ℓ on the subspace $L^2_{\deg}$ of $L^2(\mathcal{G})$ given by the constant functions over the subgraph $\mathcal{G}^1$ of vertices connected by edges of degree $1 \bmod N$ is $\ell+1$.
3. The operator norm of A_ℓ on the orthogonal complement $L^2_0(\mathcal{G})$ of $L^2_{\deg}$ is at most $2\sqrt{\ell}$.

Lemma 1. *The subspace $L^2_{\deg}$ of $L^2(\mathcal{G})$ is one dimensional.*

Proof (of Lem. 1). The lemma is a direct consequence of [48, Prop. 3.11]. To prove this, we need to look at the action of $G = (\mathrm{End}(E_0)/N\,\mathrm{End}(E_0))^\times$ over $\mathcal{F}_N(E_0)$, where E_0 is the supersingular elliptic curve with j-invariant 1728, as in [48, Prop. 3.11]. Since N is coprime with p, we have that $\mathrm{End}(E_0)/N\,\mathrm{End}(E_0)$ is isomorphic to $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ and $\mathcal{F}_N(E_0)$ is isomorphic to bases of $\mathbb{Z}/N\mathbb{Z}\times\mathbb{Z}/N\mathbb{Z}$ with fixed determinant. The action of G on $\mathcal{F}_N(E_0)$ is given by the multiplication of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ on the bases of $\mathbb{Z}/N\mathbb{Z}\times\mathbb{Z}/N\mathbb{Z}$, composed with the division by the determinant of the matrix.

It is then obvious that there is only one orbit for the action of G on $\mathcal{F}_N(E_0)$ and the stabilizer of any element is the group of scalar matrices, which has size $N-1$. We thus have that the graph $\mathcal{G}_{\deg}$ contains only one vertex, thus the map $\mathrm{Deg}:\mathcal{G}\to\mathcal{G}_{\deg}$ is constant. Since, by [48, Prop. 3.11(6)] all functions in $L^2_{\deg}$ factor through Deg, we conclude that $L^2_{\deg}$ is one dimensional.

Spectral gap for non-backtracking random walks. By the previous discussion, we can already conclude that a random walk in $\mathcal{G}$ asymptotically converges to the stationary distribution over the vertices of $\mathcal{G}$. However, we can improve the convergence rate by accounting for the non-backtracking condition, i.e. the fact that we consider only cyclic isogenies, as done in [7].

Considering a random walk of length k and let $A^{(k)}$ be the adjacency matrix of the graph of non-backtracking walks of length k. We compute $A^{(k)}$ recursively as follows:

– after 1 step we have $A^{(1)} = A_\ell$.

– after 2 steps we have $A^{(2)} = A_\ell^2 - (\ell + 1)$, since we need to remove the backtracking walks, which are exactly the ones that go from (E', P', Q') to $(E'', \zeta\phi(P'), \zeta\phi(Q'))$ and then back to $(E', \zeta^2\ell P', \zeta^2\ell Q')$. Note that at the first step there are $\ell + 1$ of them.

– after $k > 2$ steps we have $A^{(k)} = A_\ell A^{(k-1)} - \ell A^{(k-2)}$. This is because at the k-th step we start from E', P', Q', obtained after $k - 1$ steps, and we can make any of the $\ell + 1$ possible steps, excluding the one that goes to one of the ℓ previous vertices reached at the step $k - 2$, but multiplied by $\ell\zeta^2 = 1 \pmod N$.

Note that the recursive definition of $A^{(k)}$ is the same as the one from [7, Thm. 11], [21, Prop. 1.12] and [4, Lem. 2.3], so we can estimate the operator norm of $A^{(k)}$ using the same recursive argument.

The normalization of $A^{(k)}$ is given by $P^{(k)} := A^{(k)}/((\ell + 1)\ell^{k-1})$, that is diagonalizable with real eigenvalues on the same eigenvectors as A_ℓ and eigenvalue 1 on L^2_{deg}. To bound the operator norm on $L^2_0(\mathcal{G})$ we can use, as in the proofs of [7, Thm. 11], [4, Thm. 1.1] the property of Chebyshev polynomials in combination with the operator norm bounds on A_ℓ on $L^2_0(\mathcal{G})$, which is at most $2\sqrt{\ell}$, to show that the operator norm of $P^{(k)}$ on $L^2_0(\mathcal{G})$ is at most

$$\frac{1}{\sqrt{\ell^{k-1}(\ell+1)}} \cdot \left(\sqrt{\frac{\ell}{\ell+1}}(k+1) + \frac{1}{\sqrt{(\ell+1)\ell}}(k-1) \right) = \frac{(\ell+1)k + (\ell-1)}{\sqrt{\ell^k}(\ell+1)} \quad (6)$$

Bounding the distance. Let $f \in L^2(\mathcal{G})$ be the trivial distribution equal to 1 on E, P, Q and 0 elsewhere. We decompose $f = f_0 + f_1$ where $f_1 \in L_{\text{deg}}$ and f_0 is orthogonal to L_{deg}. To compute f_1 we use the operator F from [48, App. A.2]. By Lem. 1 L_{deg} is one dimensional, so Ff is the constant function and so the stationary distribution of the random walk defined by A_ℓ is

$$\frac{1}{\#\operatorname{Aut}(E, P, Q)} f_1(E, P, Q) = \nu \ . \quad (7)$$

Instead, π_k is exactly the distribution obtained by applying $P^{(k)}$ to the trivial distribution $f = f_0 + f_1 \in L^2(\mathcal{G})$. Let μ be the measure defined by $\mu(x) = \#\operatorname{Aut}(x)$ for any vertex x of $\mathcal{G}$. By the previous discussion, we can finally

bound using the classical argument, the distance between ν and π_k as follows:

$$\|\nu - \pi_k\|_1 = \sum_{x \in \mathcal{G}} \frac{1}{\mu(x)} |\nu(x) - \pi_k(x)| \tag{8}$$

$$= \sum_{x \in \mathcal{G}} \frac{1}{\mu(x)} \left| P^{(k)} f_0(x) + P^{(k)} f_1(x) - \nu \right| \tag{9}$$

$$\text{(By (7))} = \sum_{x \in \mathcal{G}} \frac{1}{\mu(x)} \left| P^{(k)} f_0(x) \right| \tag{10}$$

$$\text{(Cauchy-Schwarz)} \leq \sqrt{\sum_{x \in \mathcal{G}} \frac{1}{\mu(x)}} \cdot \left\| P^{(k)} f_0 \right\|_2 \tag{11}$$

$$\text{(Bound in (6))} \leq \sqrt{\#\mathcal{G}} \cdot \frac{(\ell+1)k + (\ell-1)}{\sqrt{\ell^k}(\ell+1)} \|f\|_2 \leq \sqrt{\frac{p}{24}} N^{3/2} \cdot \frac{k+1}{\ell^{k/2}} \tag{12}$$

This concludes the proof of Thm. 3.

A.3 New parameter sets for DeuringVUF-RK

The parameters for DeuringVUF given in [44] are not compatible with the requirements for Alg. 2 from Sec. 4.2, necessary to instantiate the Adapt algorithm. Specifically, this is due to the fact that $f = a = \lceil \log(N) \rceil$, thus we need to consider a slightly larger prime p such that $f > a$, for example the 272-bit prime given by

$$p = 611 \cdot 2^{262} - 1$$

with $N = (p-1)/17694602046$ given by

$$N = 25589351180461199199796458636782906876199489071940943588932462646724137 7$$

gives $f = 262$ and $a = 238$, thus $f - a = 24 > 0$. Note that $N \equiv 1 \pmod 8$, thus N satisfies the requirements from Sec. 4.4 for $\ell = 2$. As for [44, § 4.2], the prime p is chosen such that $f = \nu_2(p+1)$ is large enough to allow for a sufficiently large number of pushforwards in Alg. 2. We can select $\kappa = \iota$ and α to compute $I_{P_0} = \langle \alpha, N \rangle \mathcal{O}_0$ as:

$$5149745590329352664671349142607404356028189083743934380444615304170577 6+\pi \ .$$

According to the estimates from Cor. 2, we need to consider a random isogeny of degree $u_{\mathsf{cmp}} \cdot r_{\mathsf{rand}} \geq 1239$. Since $u_{\mathsf{cmp}} = f - a = 24$, we can select $r_{\mathsf{rand}} = 52$ to achieve this. The adjusting parameter from Fig. 8 is

$$\eta = 16668362671512805754381229567023226316678706602372287692 8 \ .$$

B Hint-based proof for SQIsign-RK unforgeability

We give now an alternative proof of the EUF-CMA security of SQIsign-RK to the one given in Sec. 5.3, not relying on intereactive isogeny oracles, but instead on the *non-interactive* hint-based framework of [3].

As for the proof of Thm. 2, it relies on the Multi-Instance Reset Lemma from [43], that we recall here for the reader's convenience.

Lemma 2 (Multi-Instance Reset Lemma, [43]). *Let $\mathcal{C}$ be a randomized algorithm that, on input a random instance I and a random element h of a finite set H returns a bit b and a side output* tr. *Let* acc *be the probability that $\mathcal{C}$ outputs* 1 *and* res *the probability that* $\mathsf{Rwd}_{\mathcal{C}}$ *outputs a non-$\perp$ value. Then, for any $N \geq 1$, we have*

$$\mathsf{res} \geq \left(1 - \left(1 - \mathsf{acc} + \frac{1}{|H|}\right)^N\right)^2 \tag{13}$$

Also, if $\mathcal{C}$ runs in time t, then $\mathsf{Rwd}_{\mathcal{C}}$ *runs in time $t' \approx 2Nt$ and there is $N \leq (\mathsf{acc} - 1/|H|)^{-1}$ such that:*

$$\frac{\mathsf{acc}}{t} - \frac{1}{t|H|} \leq 6\frac{\mathsf{res}}{t'} \ . \tag{14}$$

$\mathsf{Rwd}_{\mathcal{C}}(I_1, \cdots, I_N)$:
1: **for** $i = 1$ to N **do**:
2: Sample $h_i \xleftarrow{\$} H$ and randomness ρ_i for $\mathcal{C}$;
3: get $(b_i, \mathsf{tr}_i) \leftarrow \mathcal{C}(I_i, h_i)$ using ρ_i;
4: **if** $\exists i^* : b_{i^*} = 1$ **then**
5: **for** $j = 1$ to N **do**
6: Sample $h'_j \xleftarrow{\$} H$;
7: Get $(b'_j, \mathsf{tr}'_j) \leftarrow \mathcal{C}(I_{i^*}, h'_j)$ using ρ_{i^*};
8: **if** $\exists j^* : b'_{j^*} = 1$ and $I_{j^*} \neq I_{i^*}$ **then**
9: **return** $(i^*, \mathsf{tr}_{i^*}, \mathsf{tr}'_{j^*})$;
10: **return** $\perp$;

Fig. 9: Multi-Instance Reset Rewinder

Note that equation (14) is implied by the proof of [43, Lem. 3.5], by choosing $N = (\mathsf{acc} - 1/|H|)^{-1}$ or $N = 1$.

Recall the following variant of Prob. 1 from [3] that additionally considers isogeny hints.

*Problem 4 (q-**hint-OneEnd**$_p$, Problem 4 [3]).* Given a curve E sampled from the stationary distribution on the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$, and q hints $h_1, \ldots, h_q \leftarrow \mathcal{H}_E^{\mathsf{unif}}$, find non scalar endomorphism in efficient representation.

$\mathcal{H}_E^{\mathsf{unif}}$:

1: Sample an integer d from a weighted distribution on the integers $\leq 2\sqrt{2p}/\pi$ where each integer n, with prime factorization $n = \prod_{i=1}^{t} p_i^{e_i}$, has weight $\prod_{i=1}^{t}(p_i+1)^{e_i}$.
2: Sample an isogeny $\varphi_1 : E \to E_1$ uniformly among the (possibly non-cyclic) isogenies from E of degree d.
3: Sample $\varphi_2 : E_1 \to E_2$ of degree $2^a - d$, for $a = \lceil \log(d) \rceil$. **return** $h = (\varphi_1, \varphi_2)$.

$\mathcal{H}_E^{\mathsf{sim}}$: (same as [3, Exp. 3])

1: Sample $\varphi_1 : E \to E_1$ of degree $\mathsf{d_{ch}}$.
2: Sample $\varphi_2 : E_1 \to E_2$ such that:
3: a. E_2 is sampled from the stationary distribution on the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$;
4: b. The conditional distribution of φ_2 given E_2 is uniform over isogenies from E_1 to E_2 of degree $\leq 2\sqrt{2p}/\pi$.
5: Write $\deg(\varphi_2) = 2^n d'$ with d' odd.
6: Sample $\varphi_3 : E_2 \to E_3$ of degree $2^a - d'$ for $a = \lceil \log(d') \rceil$;
7: **return** $h = (\varphi_1, \varphi_2, \varphi_3)$.

RealHints(E, q):

1: **for** $s = 1$ **to** q_1 **do**
2: Set $\phi = id$, $E_s = E$ if $s = 0$
3: Sample $\phi : E \to E_s$ of degree d_r if $s > 0$
4: **for** $t = 1$ **to** q_2 **do**
5: $(\varphi_1, \varphi_2, \varphi_3) \leftarrow \mathcal{H}_{E_s}^{\mathsf{sim}}$
6: $h_{s,t} \leftarrow (\phi, \varphi_1, \varphi_2, \varphi_3)$
7: **return** $(h_{s,t})_{s,t \in [0,q_1] \times [1,q_2]}$

PushedHints(E, q_1, q_2):

1: **for** $s = 0$ **to** q_1 **do**
2: Set $\phi = id$, $E_s = E$ if $s = 0$
3: Sample $\phi : E \to E_s$ of degree d_r if $s > 0$
4: **for** $t = 1$ **to** q_2 **do**
5: Sample $\varphi_1 : E_s \to E_{\mathsf{ch}}$ of degree $\mathsf{d_{ch}}$
6: $h' \leftarrow \mathcal{H}_E^{\mathsf{unif}}$
7: $(\varphi_2, \varphi_3) \leftarrow \mathsf{PushHint}(E, h', \varphi_1 \circ \phi)$ ▷ *See [3, Alg. 6]*
8: $h_{s,t} \leftarrow (\phi, \varphi_1, \varphi_2, \varphi_3)$
9: **return** $(h_{s,t})_{s,t \in [0,q_1] \times [1,q_2]}$

Fig. 10: The hint distribution $\mathcal{H}_E^{\mathsf{unif}}$ and $\mathcal{H}_E^{\mathsf{sim}}$. Algorithms for the experiment in Prob. 5, inspired by [3, Exp. 4].

We refer to [3] for the definition of the hint distributions $\mathcal{H}_E^{\mathsf{unif}}$ and $\mathcal{H}_E^{\mathsf{sim}}$.

The EUF-CMA security SQIsign in [3], reduces to the hardness of Prob. 4 and to an additional indistinguishability assumption between two distributions of hints [3, Prob. 3]. The same hold for our scheme SQIsign-RK, with a slightly different definition of the hint distributions and so of the indistinguishability assumption.

*Problem 5 ((q_1, q_2)-**hint-dist**).* Let E be a curve sampled from the stationary distribution on the set of supersingular elliptic curves over $\mathbb{F}_{p^2}$ and $(h_{s,t})_{s,t\in[0,q_1]\times[1,q_2]}$ sampled with probability $\frac{1}{2}$ all from $\mathsf{RealHints}(E, q_1, q_2)$ and with probability $\frac{1}{2}$ all from $\mathsf{PushedHints}(E, q_1, q_2)$. Given E and the tuple, distinguish between the two distributions.

Before giving the security proof, we highlight the main differences with the results of [3] and the implications on the security reduction.

1. Because of the initial randomization step we need to consider additional hints, precisely by a factor of Q_r, the number of potential public key randomizations. Even though we won't need to actually provide them all to the adversary we still need to consider them in the security reduction, since computing pushforwards of hints would change their distribution. We consider this inevitable for a non-interactive proof, since it reflects the fact that the adversary may try to randomize the public key until a "good one" is obtained. We don't believe this to provide additional power to the adversary, given that the randomized public keys will be statistically close to the stationary distribution.
2. As noted also in Sec. 5.3, the scheme SQIsign-RK does not directly fit into the framework of the Fiat-Shamir transform with hits from [3], thus we to rely on the Multi-instance Reset Lemma from [3]. Since the lemmas requires N independent instances of the problem, we need to increase by a factor N the number of hints. We consider this to be an artifact of the proof due to the needed rewinding step.

In any case, note that the security the number of hints is still polynomial in the security parameter and, more importantly, at the current state of the art it is not taken into account for the parameter selection of SQIsign, since no attack is know that exploits the large degree isogeny hints. We conclude that even in the hint framework the unforgeability of SQIsign-RK holds under the same assumptions and parameters as the original SQIsign scheme.

Theorem 4. *Let p be a prime $\equiv 1 \pmod 4$ and N be a positive integer. Let $\mathcal{A}$ be a PPT adversary for the game G^{rkuf} (Fig. 1) against the SQIsign-RK scheme performing at most Q_s signing queries, Q_H random oracle queries to $\mathsf{H}_{\mathsf{SQI}}$ and Q_r random oracle queries to Expand. In the random oracle model, if (NQ_sQ_r)-**hint-OneEnd**$_p$ (Prob. 4) and (Q_r, Q_s)-**hint-dist** (Prob. 5) are hard, then $\mathcal{A}$'s advantage is negligible.*

Proof. Let $\mathcal{A}$ be a PPT adversary for the game G^{rkuf} (Fig. 1) against the SQIsign-RK scheme performing at most Q_s signing queries, Q_H random oracle queries to $\mathsf{H}_{\mathsf{SQI}}$ and Q_r random oracle queries to Expand. Let also $\mathsf{Adv}^{\mathcal{A}}$ be adversary's advantage in the unforgeability game.

We proceed by the following steps, mimicking the same proof given for Thm. 2 in Sec. 5.3.

- We first show how to simulate the queries to the signing oracle using the $\mathcal{H}_E^{\mathsf{sim}}$ distribution.
- Then we substitute the $\mathcal{H}_E^{\mathsf{sim}}$ distribution with the $\mathcal{H}_E^{\mathsf{unif}}$ one.
- Then we perform a random oracle guessing and obtain an algorithm to use as a subroutine for Lem. 2.
- Finally, we use the Multi-instance Reset Lemma to extract an endomorphism from the forgery and solve Prob. 4.

Simulation of the signing oracle with $\mathcal{H}_E^{\mathsf{sim}}$. For the first step we want to use the simulator from [3, Lem. 4.6] for the simulation of the signing oracle, rendering $\mathcal{A}$ to an intermediate adversary $\mathcal{B}$. Given as input a curve E_{pk} and a tuple of hints, $\mathcal{B}$ forwards E_{pk} to $\mathcal{A}$ and simulates the oracles as follows. Let $h_{s,t}$ be Q_rQ_s hints sampled from $\mathsf{RealHints}(E_{\mathsf{pk}}, Q_r, Q_s)$ and given to $\mathcal{B}$ as input. We handle $\mathcal{A}$'s queries as follows.

1. For Expand, on input E_{pk} and randomness rr_i never queried before we consider $s_i > 0$ so that $h_{s_i,1} = (\phi, \varphi_1, \varphi_2, \varphi_3)$ has not been used before and we reprogram $\mathsf{Expand}(E_{\mathsf{pk}}, \mathsf{rr}_i)$ to return the isogeny $\phi : E_{\mathsf{pk}} \to E'_{\mathsf{pk}}$. We store the pair (rr_i, ϕ) in a table $\mathcal{T}_E$ for future queries.
2. For $\mathsf{H}_{\mathsf{SQI}}$ we lazy sample the output of the random oracle and we keep track of the queries.
3. For the signing queries, on input msg_i and empty randomness $\perp$, we consider the hint $h_{0,t_i} = (id, \varphi_1, \varphi_2, \varphi_3)$ never used before. Then we call the simulator from [3, Lem. 4.6] on E_{pk} with the hint $(\varphi_1, \varphi_2, \varphi_3)$ to get an undistiguishable transcript $\mathsf{com}, \mathsf{chall}, \mathsf{resp}$. We reprogram $\mathsf{H}_{\mathsf{SQI}}(E_{\mathsf{pk}}, \mathsf{msg}_i, \mathsf{com})$ to return chall, Since the commitment com hash negligible min-entropy ([3, Lem. 4.5]) $\mathsf{H}_{\mathsf{SQI}}$ reprogramming fails with negligible probability. We return the corresponding signature.
4. For the signing queries, on input msg_j and randomness $\mathsf{rr}_i = \perp$, we consider $s_i > 0$ so that (rr_i, s_j) is in the table $\mathcal{T}_E$. We consider the hint $h_{s_i,t_j} = (\phi, \varphi_1, \varphi_2, \varphi_3)$ never used before. Let E'_{pk} be the codomain of ϕ. We call the simulator from [3, Lem. 4.6] on E'_{pk} with the hint $(\varphi_1, \varphi_2, \varphi_3)$ and proceed as in the previous case (Item 3).

By [3, Lem. 4.6] any PPT algorithm distinguishing the honest interaction with G^{rkuf} from the simulated one by $\mathcal{B}$ has negligible advantage, under the assumption on the hardness of the endormorphism ring problem, implied by the hardness of Prob. 4. Thus $\mathsf{Adv}^{\mathcal{B}} \stackrel{.}{\approx} \mathsf{Adv}^{\mathcal{A}}$, with $\mathsf{Adv}^{\mathcal{B}}$ being the advantage of $\mathcal{B}$ with input hints from $\mathsf{RealHints}(E_{\mathsf{pk}}, q_1, q_2)$.

Substitution of $\mathcal{H}_E^{\mathsf{sim}}$ with $\mathcal{H}_E^{\mathsf{unif}}$. Now, as done in [3, Thm. 3], we can substitute the input hints given to $\mathcal{B}$, instead of being sampled from $\mathsf{RealHints}(E, q_1, q_2)$, are sampled from $\mathsf{PushedHints}(E, q_1, q_2)$. Any PPT algorithm that can distinguish the two inputs with advantage $\mathsf{Adv}^{\mathsf{dist}}$ can be used to solve Prob. 5 with the same advantage, thus $\mathsf{Adv}^{\mathsf{dist}}$ is negligible under our assumptions. Thus, $\mathsf{Adv}'^{\mathcal{B}} \approx \mathsf{Adv}^{\mathcal{B}}$, with $\mathsf{Adv}'^{\mathcal{B}}$ being the advantage of $\mathcal{B}$ with input hints from $\mathsf{PushedHints}(E_{\mathsf{pk}}, q_1, q_2)$.

Random oracle guessing. We consider the intermediate game $\mathcal{C}$, taking as input the public key E_{pk}, $Q_r \cdot Q_s$ hints from $\mathcal{H}_{E_{\mathsf{pk}}}^{\mathsf{unif}}$ and a random string $h \in \mathsf{ChSet}$. Additionally, $\mathcal{C}$ controls the random oracle. First, $\mathcal{C}$ computes $\mathsf{PushedHints}(E_{\mathsf{pk}}, Q_r, Q_s)$ using the input hints, then interacts with $\mathcal{B}$ as in the same *Random oracle guessing* step in the proof of Thm. 2.

Namely:

1. $\mathcal{C}$ forwards E_{pk} to $\mathcal{B}$ and reprograms the random oracle as $\mathcal{B}$ does.
2. $\mathcal{C}$ chooses a random random oracle query and reprograms it to be h.
3. When $\mathcal{B}$ outputs a forgery, $\mathcal{C}$ checks if the random oracle query used in the forgery is the one reprogrammed to h and if it is a valid forgery, if not it outputs 0 and $\perp$. Otherwise, it returns 1 and the expanded forgery, $(E'_{\mathsf{pk}}, E_{\mathsf{com}}), (\phi_{\mathsf{chall}}, \sigma_{\mathsf{resp}}, \mathsf{rr})$.

Also, since $\mathcal{C}$ performs no more random oracle queries than $\mathcal{B}$ and the reprogrammed index is chosen independently from E_{pk} and $\mathcal{B}$ randomness, $\mathsf{Adv}^{\mathcal{C}} \doteq \mathsf{Adv}^{\mathcal{B}}/Q_H$, with $\mathsf{Adv}^{\mathcal{C}}$ being the probability that $\mathcal{C}$ outputs 1 and a valid forgery.

Rewinding and endomorphism extraction. For this last step, we proceed as in the rewinding step of the proof of Thm. 2, using Lem. 2 to extract an endomorphism from the forgery of $\mathcal{B}$, thus solving Prob. 4. There is however a notable difference with the generation of the N instances of the problem. In fact, for $\mathcal{C}$ the instance is given not only by a supersingular elliptic curve, but also by a tuple of $Q_r \cdot Q_s$ hints sampled from $\mathcal{H}_E^{\mathsf{unif}}$.

Thus the adversary $\mathcal{E}$ providing N instances for the rewinding algorithm $\mathcal{R}_{\mathcal{C}}$ need to take $N \cdot Q_r \cdot Q_s$ hints as input, and for each of the N instances to generate:

- sample a long enough isogeny walk $\tau_i : E_{\mathsf{pk}} \to E_{\mathsf{pk}}^{(i)}$ of large enough degree d, so that $E_{\mathsf{pk}}^{(i)}$ is to the stationary distribution.
- take $Q_r \cdot Q_s$ hints from $\mathcal{H}_{E_{\mathsf{pk}}}^{\mathsf{unif}}$ and push them via τ_i to get $Q_r \cdot Q_s$ hints. Since $\mathcal{H}_{E_{\mathsf{pk}}}^{\mathsf{unif}}$ is a pushable distribution by [3, Lem. 5.1] the resulting hints are distributed as $\mathcal{H}_{E_{\mathsf{pk}}^{(i)}}^{\mathsf{unif}}$.

By the Multi-Instance Reset Lemma, the probability that $i^* \geq 1$ is at least, thus the probability $\mathsf{Adv}^{\mathcal{E}}$ that $\mathcal{E}$ solves Prob. 4 is at least

$$\mathsf{Adv}^{\mathcal{E}} \geq \left(1 - \left(1 - \mathsf{Adv}^{\mathcal{C}} + \frac{1}{\mathsf{ChSet}}\right)^N\right)^2 \tag{15}$$

and the running time of $\mathcal{E}$ is $t_{\mathcal{E}} \approx 2Nt_{\mathcal{C}}$.

□

C Isogeny to endomorphism ring

In this section, we explain how to obtain a representation of the endomorphism ring $\mathrm{End}(E')$ when we are given (an efficient representation of) a smooth degree isogeny $\varphi : E \to E'$ and a representation of $\mathrm{End}(E)$. When the kernel ideal I_φ of φ is known, we can directly apply [25, Alg. 8] (inspired from [38, Alg. 4]) to obtain a $\mathbb{Z}$-basis of $\mathrm{End}(E')$ that we can evaluate on points of order coprime with $\deg(\varphi)$.

The problem then reduces to computing I_φ, *i.e.*, translating an isogeny into an ideal. In our application, we will apply IsoToEnd on isogenies of degree $2^\bullet$. Recall that our base prime is of the form $p = c2^f - 1$. This means that, using state-of-the-art techniques from [40], [1, Alg. 3.17] can only translate an isogeny $\varphi : E \to E'$ of degree $2^n \leq 2^f$ into an ideal (when $\mathrm{End}(E)$ is known), by accessing the 2^f-torsion of E defined over $\mathbb{F}_{p^2}$. However $\deg(\varphi) = 2^n$ is usually bigger than 2^f so standard techniques do not apply.

Following up the approach from [38, Alg. 9] and [32, Alg. 7], we cut the isogeny to translate into several pieces. Let us write $\varphi := \varphi_r \circ \cdots \circ \varphi_1$, the isogeny we want to translate with $\deg(\varphi_i) = 2^f$ for $1 \leq i \leq r-1$ and $\deg(\varphi_r) \leq 2^f$. Let us denote by I_i the ideal associated to φ_i for all $1 \leq i \leq r$.

The computation of I_1 is a direct application of [1, Alg. 3.17] to $\varphi_1 : E \to E_2$ and $\mathrm{End}(E)$. Now, from $\mathrm{End}(E)$ and $\varphi_1 : E \to E_2$, we can compute $\mathrm{End}(E_2)$ via [25, Alg. 8] (inspired from [38, Alg. 4]). In theory, we can then apply [1, Alg. 3.17] to $\varphi_2 : E_2 \to E_3$ and $\mathrm{End}(E_2)$ to obtain I_2. However, in practice, the representation of $\mathrm{End}(E_2)$ obtained via [25, Alg. 8] is a $\mathbb{Z}$-basis of endomorphisms that can only be evaluated on torsion points of odd order (using the isogeny path φ_1). But we need to evaluate the $\mathbb{Z}$-basis at points of 2^f-torsion to obtain I_2 via [1, Alg. 3.17].

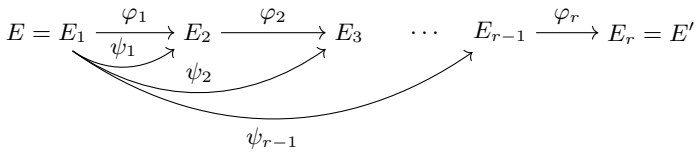


Fig. 11: Isogeny to ideal piecewise procedure.

To circumvent this difficulty, we use the Deuring correspondence (see Fig. 11). We can find an equivalent ideal $J_1 \sim I_1$ of odd norm and translate it into an isogeny $\psi_1 : E \to E_2$ using [8, Alg. 2]. Since $\deg(\psi_1)$ is odd, we can then compute $[\psi_1]^*\varphi_2$, or more exactly its kernel $\widehat{\psi}_1(\ker(\varphi_2))$. We can then obtain $[J_1]^*I_2$ from

this data and [1, Alg. 3.17], and finally compute $I_2 = [J_1]_*[J_1]^* I_2$. A similar approach applies to the following ideals $I_3, \cdots, I_r$. We summarize the whole procedure to compute I_φ in Alg. 3 which is also implemented in our proof of concept SageMath implementation given as supplemental material.

Algorithm 3 Big degree $\mathsf{IsoToIdeal}(\varphi : E \to E', \mathrm{End}(E))$

Input: A 2^n-isogeny $\varphi : E \to E'$ and its domain endomorphism ring $\mathrm{End}(E)$.
Output: The ideal I associated to φ.

Write $\varphi := \varphi_r \circ \cdots \circ \varphi_1$, where $\varphi_i : E_i \to E_{i+1}$ has degree $\leq 2^e$ for all $1 \leq i \leq r$;
$I_1 \leftarrow \mathsf{IsoToIdeal}(\varphi_1, \mathrm{End}(E))$ [1, Alg. 3.17];
Generate $J_1 \sim I_1$ of odd $\mathrm{nrd}(J_1)$;
$\psi_1 \leftarrow \mathsf{IdealToIsogeny}(J_1, E, \mathrm{End}(E))$ [8, Alg. 2];
for $i = 2, \ldots, r$ **do**
    Let $P_i \in E_i[2^e]$ be a generator of $\ker(\varphi_i)$;
    $Q_i \leftarrow \widehat{\psi}_{i-1}(P_i)$;
    Compute (a representation of) $[\psi_{i-1}]^* \varphi_i$ with kernel $\langle Q_i \rangle$;
    $[J_{i-1}]^* I_i \leftarrow \mathsf{IsoToIdeal}([\psi_{i-1}]^* \varphi_i, \mathrm{End}(E))$ [1, Alg. 3.17];
    $I_i \leftarrow [J_{i-1}]_* [J_{i-1}]^* I_i$;
    if $i \leq r - 1$ **then**
        Generate $J_i \sim (I_1 \cdots I_i)$ of odd $\mathrm{nrd}(J_i)$;
        $\psi_i \leftarrow \mathsf{IdealToIsogeny}(J_i, E, \mathrm{End}(E))$ [8, Alg. 2];
return $I_1 \cdots I_r$;