




Erebor and Durian: Full Anonymous Ring Signatures from Quaternions and Isogenies

Giacomo Borin¹, Yi-Fu Lai², and Antonin Leroux³

¹ IBM Research Europe, University of Zurich
giacomo.borin@ibm.com

² Ruhr-Universität Bochum
Yi-Fu.Lai@ruhr-uni-bochum.de

³ Direction Générale de l’Armement, Université de Rennes
antonin.leroux@polytechnique.org

Abstract. We construct two efficient post-quantum ring signatures with anonymity against full key exposure from isogenies, addressing the limitations of existing isogeny-based ring signatures.

First, we present an efficient concrete distinguisher for the SQIsign simulator when the signing key is provided using one transcript. This shows that turning SQIsign into an efficient full anonymous ring signature requires some new ideas.

Second, we propose a variant of SQIsign (Asiacrypt’20) that is resistant to the distinguisher attack with only a $\times 1.33$ increase in size and we render it to a ring signature, that we refer as **Erebor**. This variant introduces a new zero-knowledge assumption that ensures full anonymity. The efficiency of **Erebor** remains comparable to that of SQIsign, with only a proportional increase due to the ring size. This results in a signature size of 0.68 KB for 4 users and 1.35 KB for 8 users, making it the most compact post-quantum ring signature for up to 31 users.

Third, we revisit the GPS signature scheme (Asiacrypt’17), developing efficient subroutines to make the scheme more efficient and significantly reduce the resulting signature size. By integrating our scheme with the paradigm by Beullens, Katsumata, and Pintore (Asiacrypt’20), we achieve an efficient logarithmic ring signature, that we call **Durian**, resulting in a signature size of 9.87 KB for a ring of size 1024.

1 Introduction

Ring Signatures. Ring signatures, a cryptographic primitive introduced by Rivest, Shamir, and Tauman-Kalai [53], enable a member of a group (referred to as a ring) to sign a message on behalf of the entire group without revealing which specific member signed the message. The original application of ring signatures was to protect whistle-blowers, allowing them to leak information anonymously while ensuring the information’s credibility by proving it was released by someone within the group due to *unforgeability*. Today, ring signatures are widely utilized in various fields, such as electronic voting systems [41], confidential transactions in blockchain technology [48,62,26], secure messaging [31], deniable key exchanges [9] and deniable AKEM [28]. In many applications, *full anonymity* of the underlying ring signature is essential, ensuring the signer’s anonymity even if all signing keys are exposed.

Ring signatures exist for a variety of classical assumptions [53,2,7,30,61]. However, these number-theoretic assumptions can be solved by a quantum computer in polynomial time [56], rendering these schemes insecure against adversaries equipped with sufficiently powerful quantum computers. To address this issue, many *post-quantum* ring signature schemes have been proposed [33,42,61,6,5,26,28]. Among the state-of-the-art proposals are lattice-based instances of the *linear*⁴ ring signatures Gandalf [28] and DualRing [61], which needs 1.2 KB for 2 users and 4.7 KB for 8 users respectively and grow linearly with the ring size. The *logarithmic*⁵ ring signature SMILE [43] requires 18 KB for a ring of size 1024. Among the post-quantum

⁴ The signature size grows linear to the ring size.

⁵ The signature size grows logarithmic to the ring size.

proposals, there are two constructions from isogenies [6,5], which are *linkable* and *accountable* ring signatures respectively, by using the isogeny group actions [11].

The isogeny problem, which lies at the heart of isogeny-based cryptography, asserts that given two isogenous elliptic curves, it is hard to compute an isogeny between them. Imposing restrictions on the elliptic curves leads to the isogeny group action, which offers richer algebraic properties and has proven to be a versatile branch in isogeny-based cryptography [38,6,23,5,32]. However, due to the innate structure of *abelian* group actions, it suffers from the subexponential time attacks [35,36], so the efficient instantiations of the ring signature [6,5] does not meet the quantum security of NIST 1 [51]. To have efficient instantiations by scaling the underlying schemes has been an open problem [19,13] and is currently a bottleneck for its constructions and applications. Meanwhile, translating these constructions to the general isogeny case by removing the use of group action has been recognized as a non-trivial task. This brings us to the main question of this work:

*Can we have efficient ring signatures from isogenies
that provides both full anonymity and sufficient post-quantum security?*

Isogeny Proof of Knowledge. Isogeny zero-knowledge proof of knowledge (ZKPoK) for an isogeny problem is an active research area in isogeny research [20,29,60,58,18]. Before the SIDH attacks [10,54,44], incorporating auxiliary information like torsion points was a common research object to consider [20,60,58,18]. The GPS signature scheme [29] is another example of an isogeny ZKPoK, where the signature size is nearly tens of KB to one hundred KB. However, due to the algorithm’s high complexity, the scheme remains theoretical. The state-of-art works of isogeny ZKPoK owe credit to distinct approaches showcased in recent papers [3,14], both exhibiting comparable performance metrics in proof size and runtime. Yet, both methodologies entail proof sizes of at least a few hundred KB.

Besides, the prominent isogeny-based signature schemes [21,22,17], known as SQIsign and SQIsignHD, operate on a sigma-protocol framework without employing parallel repetitions. This simplifies the proof process while still demonstrating “partial” knowledge of the endomorphism ring of a supersingular curve E . Full knowledge of this ring is equivalent to knowing an isogeny between E and a specific E_0 with a j -invariant of 0 or 1728. The recent improvements of SQIsign2Ds share the same feature [46,4]. Notably, SQIsign stands out for its compactness among NIST submissions for post-quantum signatures [12]. These schemes are natural candidates for adaptation into ring signatures.

Methodology. It is believed that the schemes mentioned above can be transformed into 1-out-of-many proofs or ring signatures using standard approaches [16,2,33,61]. However, this is not always the case. This limitation arises from the design of the simulators used in the constructions [21,22,17].

The simulator of a signature scheme is crucial for constructing a ring signature. For example, the simulators for SQIsignHD and SQIsign2Ds rely on access to an oracle, making it infeasible to simulate transcripts for a ring signature in a real-world setting. Similarly, the simulator for SQIsign cannot generate a transcript with a prefix challenge, rendering it incompatible with the sequential approach [2,61]. It is folklore that if the signing key is given, an efficient distinguisher for the SQIsign simulator exists. However, a rigorous analysis has never been given in the literature. In this work, we present an efficient algorithm demonstrating that it is possible to distinguish simulated transcripts, which precludes the use of existing methods to achieve fully anonymous ring signatures with SQIsign.

1.1 Contributions.

1. We present a concrete distinguisher in theorem 3.3 for the simulator of SQIsign when the signing key is provided. Supported by the experimental result, the algorithm is efficient, requiring only *one* transcript to distinguish. We stress again that this does not constitute an attack on SQIsign, as it necessitates the secret key to execute, and that the core ideas of this distinguisher were already known to the community, but never considered relevant for a real-world construction.

2. We propose Erebor ⁶, a linear ring signature based on a variant of SQIsign that is resistant to the aforementioned distinguisher attack and introduce a new assumption for zero-knowledge. Unlike the original SQIsign, this variant is compatible with both parallel and sequential OR proofs [16,2]. The resulting linear ring signatures offer full anonymity based on the new assumption, for which we provide a security argument. This leads to the most compact post-quantum ring signatures with full anonymity with a ring size less than 32. As an independent interest, we provide a shorter version considering anonymity without key exposure.
3. We revisit the GPS signature scheme. By tweaking the scheme and developing efficient subroutines, we make the scheme feasible and significantly reduce the resulting signature size. Additionally, by integrating our new scheme with an adaptation of the group action paradigm introduced in [6], we achieve an efficient logarithmic ring signature Durian ⁷. This results in the most compact logarithmic post-quantum ring signatures, providing full anonymity in a statistical sense.

1.2 Technical Overview

Due to the Deuring correspondence (Table 1), an isogeny between two supersingular curves E and E' corresponds to a connecting ideal, which serves as both a left $\text{End}(E)$ -ideal and a right $\text{End}(E')$ -ideal within the quaternion algebra. For simplicity, we may occasionally interchange the objects “curve” and “endomorphism ring” and the terms “ideal” and “isogeny” when the context is clear.

We now explain our contributions in detail. In the context of signatures derived from non-interactive proof of knowledge identification protocols we can see ring signatures as a special case of an one-out-of-many proof (namely, R-proofs): to prove of one knowledge out of many public statements or problems. Classical techniques like sequential and parallel OR-proofs [16,2] exploit the same simulator used to prove the honest-verifier zero knowledge property. For identification protocols achieving statistical indistinguishability between honest transcripts and simulated ones this immediately implies statistical full-anonymity, while for protocols relying on computational assumptions to prove the indistinguishability, like [21], there are two major differences:

- we need the same computational assumption to prove the anonymity,
- to achieve full-anonymity we need indistinguishability to hold even with access to the secret key.

We start by formally showing that for SQIsign this last point does not hold.

Distinguisher. The high-level idea of the distinguisher is to use the secret key to do the “reverse engineer” to recover the randomness used in the signing algorithm by exploiting the Eichler orders’ properties. Roughly, the signing algorithm of SQIsign proceeds as follows. The protocol is to prove knowledge of the endomorphism ring of a curve E_{pk} . This is equivalent to proving knowledge of an ideal between E_{pk} and E_0 , where E_0 has a j -invariant of 1728.

The main algorithm, `SigningKLPT`, takes as input an ideal I and the secret ideal and returns a random, equivalent ideal of a power-smooth norm. Here, I is the ideal connecting the public curve E_{pk} and a challenge curve E_{ch} chosen by the verifier. Importantly, the randomness of the ideal returned by `SigningKLPT` hides information about the secret ideal. In detail, I is first randomized within a class group to obtain \tilde{I} . Next, the algorithm finds an equivalent ideal for \tilde{I} with a power-smooth norm by a few subroutines. Finally, the resulting ideal is translated into an isogeny between E_{pk} and E_{ch} and sent to the verifier together with the commitment curve and the challenge isogeny.

In contrast, the simulator procedure is much simpler. It generates a random isogeny $\sigma' : E_{\text{pk}} \rightarrow E'_{\text{ch}}$ of a specific degree and then computes a random challenge isogeny $\hat{\phi}'_{\text{ch}} : E'_{\text{ch}} \rightarrow E'_{\text{cmt}}$ of a specific degree. The simulator outputs a simulated transcript as $(E'_{\text{cmt}}, \phi'_{\text{ch}}, \sigma')$. The main difference between the transcripts is the way to generate the isogeny σ' . The indistinguishability ensures the computational zero-knowledge property of SQIsign.

⁶ Short for “Eichler order RE-randomizing-Based OR-proof.”

⁷ Short for “DeUring correspondence-based RIng signature with full ANonymity.”

Our distinguisher proceeds as follows. By assuming access to the secret ideal, we can translate both isogenies to left $\text{End}(E_{\text{pk}})$ -ideals. We may assume the distributions of the ideals are uniformly random over each support, denoted by S_{real} and S_{sim} , respectively. We note that even though $S_{\text{real}} \subset S_{\text{sim}}$ and S_{real} is negligible compared to S_{sim} , the size of S_{real} is still exponentially large in the security parameter. Hence, enumerating the ideals to distinguish by querying the oracle will be infeasible. On the other hand, it suffices to determine if the resulting ideals are in S_{real} to distinguish.

In the case of the real transcript, we observe three facts: 1). The ideal, translated from the isogeny, is the output of **SigningKLPT** and stays in the same class as \tilde{I} . 2). The procedure for finding the equivalent ideal of a power-smooth norm does not depend on the representative of a class. 3). We can invoke a meet-in-the-middle type approach to recover the randomness used in the previous procedure.

The last step is feasible because the former part of the equivalent-ideal-finding procedure has only polynomially-many solutions (see the estimation of 6) and the latter part has a specific structure (due to the strong approximation) to derive the output. For a more detailed explanation, refer to Item 2. Therefore, by fixing this ideal and running the equivalent-ideal-finding subroutines of **SigningKLPT**, we can recover the randomness used in **SigningKLPT**. In contrast, when running on input the simulated transcripts, the distinguisher will not terminate. Supported by the implementation, the distinguisher succeeds with an overwhelming probability using just one transcript.

Linear-size Ring Signature. The blueprint under Erebor design is the classical AOS framework [2], which provides simple and efficient ring signatures from any identification protocol. To have a full-anonymous ring signature, we need to modify **SQISign** in two ways. The primary goal is to address the distinguishability issue when the secret key is available. We introduce a new KLPT variant (Algorithm 3) for the signing algorithm. The high-level idea is to randomize and find an equivalent ideal for the secret ideal before executing **SigningKLPT**. This gives a better zero-knowledge than **SQISign** for two reasons. First, since now the pullback would send an endomorphism to random morphisms, it is hard to recover the left \mathcal{O}_0 -class using in the signing algorithm. Furthermore, this increases the possibilities for each intermediate variable by an exponential factor by choosing a sufficiently large degree for the equivalent ideal. Hence, our new signing KLPT algorithm makes the abovementioned distinguisher fail. As a result, we obtain an **SQISign** variant with a better zero-knowledge guarantee, incurring only minor overheads in efficiency and signature size. We provide a careful counting argument to analyze the new assumption and conjecture its hardness in Section 4.3.

Then, to apply the AOS framework [2], we tweak the **SQISign** diagram: having the challenge starting from the public key E_{pk} instead of from E_0 . In this way we can produce a simulated transcript for any given challenge isogeny. With minor modification we can employ the same building blocks of **SQISign** and achieve the same final results. As a result, we present the most compact full-anonymous post-quantum ring signature for up to 31 users.

Logarithmic-size Ring Signature. The GPS signature scheme [29] is based on a parallel-repeated sigma protocol with a challenge space of size 2. The prover shows the knowledge of an isogeny between E_0 and E , where the endomorphism ring of E_0 is known.

At a high level, the prover selects a subgroup S of E of power-smooth size, computes the codomain curve of the isogeny with the kernel S , and commits to this curve, denoted as E' . Depending on the challenge bit from $\{0, 1\}$, the prover reveals an isogeny path from either E_0 or E to E' . Revealing S suffices to compute the isogeny between E and E' , which is also simulatable since it does not require a secret key.

When revealing the isogeny between E_0 and E' , the prover uses the secret isogeny between E_0 and E to produce the isogeny between E_0 and E_1 . Revealing the composed isogeny will leak the secret key, so the prover has to compute the endomorphism ring $\text{End}(E')$ using the isogeny and the known $\text{End}(E_0)$. Here, we can use the *Ramanujan* property of the supersingular isogeny graph to simulate the transcript. Then, the prover computes the connecting ideal between $\text{End}(E_0)$ and $\text{End}(E')$ and translates the ideal to an isogeny from E_0 to E' with a power-smooth degree. This translation is the primary source of inefficiency.

Here, we adopt a different approach. Instead of revealing the isogeny between E_0 and E' , the prover reveals the optimal connecting ideal, which has the smallest norm by using the lattice reduction in dimension 4. The benefit of using this approach is twofold. First, this does not require the norm to be power-smooth, thereby

avoiding the lengthy loop in generating the response. Second, the optimal ideal can be represented using approximately $\log_2(p)$, which is nearly optimal given that there are roughly $O(p)$ isomorphism classes for supersingular curves. By generalizing the algorithms in [22], we can efficiently compute this representation, estimating the process to take less than 10ms. To reconstruct the curve E' in our scheme, the verifier computes the isogeny using a simple adaptation of the new ideal-to-isogeny algorithm developed in [4, Algorithm 3]. This computation is estimated to take less than 40 ms.

Beullens, Katsumata and Pintore [6] provide a group-action-based framework for logarithmic ring signatures. Although we are not using group action here, it is straightforward to construct a logarithmic ring signature using our improved signature scheme based on the same principles. Given E_0 and E_1, \dots, E_N , to prove the knowledge of an isogeny between E_0 and E_I for some $I \in [N]$, the prover computes an isogeny for each E_i where $i \in [N]$ by randomly choosing a subgroup of smooth size over each E_i . The prover then shuffles and commits to the codomain curves E'_i . Depending on the challenge bit in $\{0, 1\}$, the prover reveals either all subgroups or an optimal connecting ideal between $\text{End}(E_0)$ and E'_I . We utilize standard optimization techniques to improve the signature size, as detailed in section 5. Surprisingly, the resulting logarithmic ring signature is only slightly larger than the group-action counterpart while achieving NIST level 1 security, as shown in table 4.

2 Preliminaries

Notations. We write $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ for the sets of natural numbers, integers, and rational numbers. For $N \in \mathbb{N}$, $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ denote the projective space modulo N . For $M \in \mathbb{N}$ we let $[M] := \{1, \dots, M\}$. For an ideal I , let $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$ denote the left and the right order of I respectively.

2.1 Sigma protocols

Definition 2.1 (Sigma Protocol). *A sigma protocol Π_Σ is a three-move identification protocol for a NP relation R consists of oracle-calling PPT algorithms Gen and $(P = (P_1, P_2), V = (V_1, V_2))$, where V_2 is deterministic. We assume Gen, P_1 and P_2 share states and so does V_1 and V_2 . Let ChSet denote the challenge space. Then, Π_Σ proceeds as follows.*

- The prover, gets a valid relation $(x, w) \leftarrow \text{Gen}(1^\lambda)$ and publish x ;
- The prover, on input $(x, w) \in R$, runs $\text{com} \leftarrow P_1^O(x, w)$ and sends a commitment com to the verifier.
- The verifier runs $\text{ch} \xleftarrow{s} V_1^O(1^\lambda)$, drawing a random challenge from ChSet , and sends it to the prover.
- The prover, given ch , runs $\text{rsp} \leftarrow P_2^O(x, w, \text{ch})$ and returns a response rsp to the verifier.
- The verifier runs $V_2^O(x, \text{com}, \text{ch}, \text{rsp})$ and outputs 1 (accept) or 0 (reject).

Here, O is modeled as a random oracle. For simplicity, we often drop O from the superscript when it is clear from the context. We assume the statement x is always given as input to both the prover and the verifier. The protocol transcript $(\text{com}, \text{ch}, \text{rsp})$ is said to be valid in case $V_2(\text{com}, \text{ch}, \text{rsp})$ outputs 1.

We consider the following properties for a Σ -protocol:

Correctness. A sigma protocol Π_Σ is said to be *correct* if for all $(x, w) \in R$ and the prover and the verifier both follow the protocol specification, the verifier always outputs 1.

Honest Verifier Zero-Knowledge. We have a few distinct notions for zero-knowledge. We start with the standard one, then we consider the concepts more relevant to the SQsign protocol and our constructions.

We say Π_Σ is *{statistically, computationally} honest-verifier-zero-knowledge (HVZK)* for relation R if there exists a PPT simulator \mathcal{S}^O with access to a random oracle O such that for any $\lambda \in \mathbb{N}$, pair $(x, w) \in R$

and any $\{\text{computationally unbounded, PPT}\}$ adversary \mathcal{A} that makes at most a polynomial number of queries to \mathcal{O} , we have

$$\text{Adv}_{\Pi_{\Sigma}}^{\text{HVZK}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}}(P^{\mathcal{O}}(x, w)) = 1] - \Pr[\mathcal{A}^{\mathcal{O}}(\mathcal{S}^{\mathcal{O}}(x)) = 1]| = \text{negl}(\lambda), \quad (1)$$

where $P = (P_1, P_2)$ is a prover running on (x, w) and the probability is taken over the randomness used by (P, V) and by the random oracle.

- Π_{Σ} is said to be *special HVZK* if the challenge $\text{ch} \in \text{ChSet}$ is fixed in advance for both the prover and the simulator and (1) holds for any $\text{ch} \in \text{ChSet}$.
- Π_{Σ} is said to be *strong HVZK* if (1) holds conditioned on that $(x, w) \leftarrow \text{Gen}(1^{\lambda})$ and the \mathcal{A} is has access to w .
- Π_{Σ} is said to be *weak HVZK* if (1) holds conditioned on that $(x, w) \leftarrow \text{Gen}(1^{\lambda})$ and the \mathcal{A} is has no access to w .

Looking ahead, we will use computationally special strong HVZK and statistically special HVZK properties respectively to construct ring signatures with full anonymity. Intuitively, the indistinguishability provided by these properties ensures anonymity even when the secret key (i.e. w) is exposed. In contrast, we will also show that the sigma protocol of SQIsign , which has been shown to be weak HVZK, does not satisfy *strong HVZK* and hence cannot be transformed into a fully anonymous ring signature using existing paradigms.

Special Soundness. We say a sigma protocol Π_{Σ} has special soundness if there exists a polynomial-time extraction algorithm Extract such that, given a statement x and any two valid transcripts $(\text{com}, \text{ch}, \text{rsp})$ and $(\text{com}, \text{ch}', \text{rsp}')$ relative to x and such that $\text{ch} \neq \text{ch}'$, outputs a witness w satisfying $(x, w) \in R$.

High Min-Entropy. We say a sigma protocol Π_{Σ} has $\alpha(\lambda)$ min-entropy if for any $(x, w) \in R$, and a possibly computationally-unbounded adversary \mathcal{A} , we have

$$\Pr[\text{com} = \text{com}' | \text{com} \leftarrow P_1^{\mathcal{O}}(x, w), \text{com}' \leftarrow \mathcal{A}^{\mathcal{O}}(x, w)] \leq 2^{-\alpha},$$

where the probability is taken over the randomness used by P_1 and by the random oracle. We say Π_{Σ} has *high min-entropy* if $2^{-\alpha}$ is negligible in λ .

A sigma protocol can be rendered to a digital signature via the well-known Fiat-Shamir transform [27] and substituting the random oracle \mathcal{O} with a cryptographic hash function.

2.2 Ring Signature

We give here basic definitions for identification protocols, ring signatures. Then we show how to construct the latter from AOS sequential OR-proofs [2], providing proofs of security tailored to our situations.

Definition 2.2 (Ring signature). A ring signature scheme Π_{RS} consists of four PPT algorithms (RS.Setup, RS.KeyGen, RS.Sign, RS.Verify) such that:

- RS.Setup(1^{λ}) \rightarrow pp : On input a security parameter 1^{λ} , it returns public parameters pp used by the scheme.
- RS.KeyGen(pp, rr) \rightarrow (pk, sk) : On input the public parameters pp and a randomness rr , it outputs a pair of public and secret keys (pk, sk) .
- RS.Sign($\text{sk}, \text{rr}, \text{m}, \text{R}$) \rightarrow σ : On input a secret key sk , a randomness rr , a message m , and a list of public keys, i.e., a ring, $\text{R} = \{\text{pk}_1, \dots, \text{pk}_N\}$, it outputs a signature σ .
- RS.Verify($\text{R}, \text{m}, \sigma$) \rightarrow $1/0$: On input a ring $\text{R} = \{\text{pk}_1, \dots, \text{pk}_N\}$, a message m , and a signature σ , it outputs either 1 (accept) or 0 (reject).

Correctness: For every security parameter $\lambda \in \mathbb{N}$, $N = \text{poly}(\lambda)$, $j \in [N]$, and every message m the following holds:

$$\Pr \left[\text{RS.Verify}(R, m, \sigma) = 1 \mid \begin{array}{l} \text{pp} \leftarrow \text{RS.Setup}(1^\lambda), \\ (\text{pk}_i, \text{sk}_i) \leftarrow \text{RS.KeyGen}(\text{pp}, \text{rr}_i) \quad \forall i \in [N], \\ R := (\text{pk}_1, \dots, \text{pk}_N), \\ \sigma \leftarrow \text{RS.Sign}(\text{sk}_j, m, R). \end{array} \right] = 1.$$

Anonymity: A ring signature scheme Π_{RS} is anonymous if, for all $\lambda \in \mathbb{N}$ and $N = \text{poly}(\lambda)$, any PPT adversary \mathcal{A} has at most negligible advantage in the following game played against a challenger.

- (i) The challenger runs $\text{pp} \leftarrow \text{RS.Setup}(1^\lambda)$ and $(\text{pk}_i, \text{sk}_i) \leftarrow \text{RS.KeyGen}(\text{pp}, \text{rr}_i)$ for all $i \in [N]$ using the randomness rr_i . It also samples a random bit $b \leftarrow \{0, 1\}$;
- (ii) The challenger provides pp to \mathcal{A} ;
- (iii) \mathcal{A} outputs a challenge (R, m, i_0, i_1) to the challenger, where the ring R must contain pk_{i_0} and pk_{i_1} .
- (iv) The challenger then runs $\sigma^* \leftarrow \text{RS.Sign}(\text{sk}_{i_b}, \text{rr}_b^*, m, R)$, and provides σ^* to \mathcal{A} ;
- (v) \mathcal{A} outputs a guess b^* . If $b^* = b$, we say the adversary \mathcal{A} wins.

The advantage of \mathcal{A} is defined as

$$\text{Adv}_{\Pi_{\text{RS}}}^{\text{Anon}}(\mathcal{A}) := |\Pr[\mathcal{A} \text{ wins}] - 1/2|.$$

The scheme is *full-anonymous* or *anonymous against full key exposure* if any PPT adversary \mathcal{A} has still negligible advantage in the game where at Item ii the challenger provides also $\{\text{rr}_i\}_{i \in [N]}$ to \mathcal{A} .

Unforgeability (UF-CMA): A ring signature scheme Π_{RS} is *unforgeable (with respect to insider corruption)* if, for all $\lambda \in \mathbb{N}$ and $N = \text{poly}(\lambda)$, any PPT adversary \mathcal{A} has at most negligible advantage in the following game played against a challenger.

- (i) The challenger runs $\text{pp} \leftarrow \text{RS.Setup}(1^\lambda)$ and generates key pairs $(\text{pk}_i, \text{sk}_i) = \text{RS.KeyGen}(\text{pp}; \text{rr}_i)$ for all $i \in [N]$ using random coins rr_i . It sets $\text{PK} := \{\text{pk}_i\}_{i \in [N]}$ and initializes two empty sets \mathcal{S} and \mathcal{C} .
- (ii) The challenger provides pp and PK to \mathcal{A} ;
- (iii) \mathcal{A} can make signing and corruption queries an arbitrary polynomial number of times:
 - (**sign**, i, m, R): The challenger checks if $\text{pk}_i \in R$ and if so it computes the signature $\sigma \leftarrow \text{RS.Sign}(\text{sk}_i, m, R)$. The challenger provides σ to \mathcal{A} and adds (i, m, R) to \mathcal{S} ;
 - (**corrupt**, i): The challenger adds pk_i to \mathcal{C} and returns rr_i to \mathcal{A} .
- (iv) \mathcal{A} outputs (R^*, m^*, σ^*) . If $R^* \subset \text{PK} \setminus \mathcal{C}$, $(\cdot, m^*, R^*) \notin \mathcal{S}$, and $\text{RS.Verify}(R^*, m^*, \sigma^*) = 1$, then we say the adversary \mathcal{A} wins.

The advantage of \mathcal{A} is defined as $\text{Adv}_{\text{RS}}^{\text{Unf}}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}]$.

In [2] the authors show how to render a set of 3-pass identification protocols satisfying the Special HVZK property to a ring signature using a circular version of the Fiat-Shamir transform [27]. Since we focus in the application of this construction to the SQlsign protocol to simplify the exposition we only consider the case in which all the considered identification protocols are the same. For the signature definition we consider as public parameters pp the security parameter λ , the relation R , algorithms $\text{Gen}, \text{P}_1, \text{P}_2, \text{V}_2$, plus the hash function $\text{H} : \{0, 1\}^* \rightarrow \text{ChSet}$ (that takes the role of V_1) and the simulator \mathcal{S} from the Special HVZK property. Note that it is important that \mathcal{S} generates the transcript given a predetermined challenge $\text{ch} \in \text{ChSet}$.

The specification are in Algorithm 1. If the Σ -protocol is commitment recoverable (i.e. we can recover the commitment from the challenge and the response) we can avoid inserting $\text{com}_1, \dots, \text{com}_N$ in the output signature.

A proof for the security of the construction can be found in [61]. We generalize the results to our case for completeness, since we involve Σ -protocols with different zero-knowledge notions. The proofs are quite straightforward and we provide them in Appendix A. As for the signature definition we focus on the case of all Σ protocols Π^{ld} being equal.

Proposition 2.3. *If Π^{ld} satisfies the special weak (resp., strong) computational HVZK property the ring signature scheme (Algorithm 1) is anonymous (resp., full-anonymous) in the programmable random oracle model.*

Proposition 2.4. *If Π^{ld} satisfies Definition A.1 the ring signature scheme (Algorithm 1) is unforgeable (UF-CMA) in the programmable random oracle model.*

Algorithm 1 AOS Sequential Ring Signature from [2]

RS.KeyGen(pp) :

- 1: Get $x, w \leftarrow \text{Gen}(1^\lambda)$
- 2: Assign $pk, sk \leftarrow x, w$
- 3: **return** (pk, sk) .

RS.Sign(sk_l, m, R) :

- 1: Get $com_l \leftarrow P_1(pk_l, sk_l)$;
- 2: Set $ch_{l+1} \leftarrow H(com_l, R, m, pk_{l+1})$;
- 3: **Parse** $(pk_1, \dots, pk_N) \leftarrow R$
- 4: **for** $i = l + 1, \dots, N, 1, \dots, l - 1$ **do**
- 5: Get $com_i, rsp_i \leftarrow \mathcal{S}(pk_i, ch_i)$;
- 6: Set $ch_{i+1} \leftarrow H(com_i, R, m, pk_{i+1})$;
- 7: Get $rsp_l \leftarrow P_2(sk_l, com_l, ch_l)$;
- 8: **return** $\sigma = (ch_1, rsp_1, \dots, rsp_N, com_1, \dots, com_N)$.

RS.Verify(R, m, σ) :

- 1: **for** $i = 1, \dots, N$ **do**
 - 2: **if not** $V_2(pk_i, com_i, ch_i, rsp_i)$ **then**
 - 3: **return reject**;
 - 4: $ch_{i+1} \leftarrow H(com_i, R, m, pk_{i+1})$;
 - 5: **if** $ch_1 = ch_{N+1}$ **then**
 - 6: **return accept**.
 - 7: **else**
 - 8: **return reject**.
-

2.3 Isogenies and Quaternions

In this section, we recall several useful mathematical definitions. Below, we assume some familiarity of the reader with basic notions on elliptic curves, isogenies, quaternion algebras and their link through the Deuring correspondence. We refer the reader to [57,59] for a more complete treatment of the overall theory, and to [39, Chapter 2] for a presentation of the Deuring correspondence as we use it. We give a brief overview as follows.

The Deuring correspondence is a mathematical result linking integral lattices of $\mathcal{B}_{p,\infty}$, the quaternion algebra ramified at p and ∞ to supersingular elliptic curves and their isogenies. To any isomorphism class of supersingular supersingular elliptic curves (up to Galois conjugacy) the Deuring correspondence associates the isomorphism class of its endomorphism ring which is an isomorphism class of maximal orders. For this reason, in this work, we often *implicitly consider curves and orders up to isomorphisms*.

Quaternion algebras, orders and ideals. This paragraph is almost a verbatim of [22]. The endomorphism rings of supersingular elliptic curves over \mathbb{F}_{p^2} are isomorphic to maximal orders of $B_{p,\infty}$, the quaternion algebra ramified at p and ∞ . We fix a basis $1, i, j, k$ of $B_{p,\infty}$, satisfying $i^2 = -q, j^2 = -p$ and $k = ij = -ji$ for some positive integer q . The canonical involution of conjugation sends an element $\alpha = a + ib + jc + kd$ to $\bar{\alpha} = a - (ib + jc + kd)$. A fractional ideal I is a \mathbb{Z} -lattice of rank four inside $B_{p,\infty}$. We define $n(\alpha) = \alpha\bar{\alpha}$. For an ideal I , we denote by $n(I)$ the norm of I as the largest rational number such that $n(\alpha) \in n(I)\mathbb{Z}$ for any $\alpha \in I$. Given fractional ideals I and J , if $J \subseteq I$ then the index $[I : J]$ is defined to be the order of the finite quotient group I/J . We define the ideal conjugate $\bar{I} = \{\bar{\alpha}, \alpha \in I\}$. An order \mathcal{O} is a subring of $B_{p,\infty}$ that is also a fractional ideal. An order is called *maximal* when it is not contained in any other larger order. The left order of a fractional ideal is defined as $\mathcal{O}_L(I) = \{\alpha \in B_{p,\infty} \mid \alpha I \subseteq I\}$ and similarly for the right order $\mathcal{O}_R(I)$. Then I is said to be an $(\mathcal{O}_L(I), \mathcal{O}_R(I))$ -ideal or a left $\mathcal{O}_L(I)$ -ideal. A fractional ideal is *integral* if it is contained in its left order, or equivalently in its right order; we refer to integral ideals hereafter as ideals. An ideal can be written as $I = \mathcal{O}_L(I)\alpha + \mathcal{O}_L(I)n(I) = \mathcal{O}_L(I)\langle\alpha, n(I)\rangle$ for some $\alpha \in \mathcal{O}_L(I)$. Two left \mathcal{O} -ideals I and J are equivalent if there exists $\beta \in B_{p,\infty}^\times$, such that $I = J\beta$. For a given \mathcal{O} , this defines equivalence classes of left \mathcal{O} -ideals, and we denote the set of such classes by $\text{Cl}(\mathcal{O})$. Also, for any ideal K and any $\alpha \in B_{p,\infty}^\times$, we write $\chi_I(\alpha) = K\bar{\alpha}/n(K)$. Ideals equivalent to K are precisely the ideals $\chi_I(\alpha)$ with $\alpha \in I \setminus \{0\}$.

Through the notion of kernel ideal, it is possible to associate an isogeny $\varphi : E \rightarrow E'$ with an ideal I_φ of left order \mathcal{O} and right order \mathcal{O}' where $I_\varphi = \{\alpha \in \mathcal{O}_0 : \alpha(P) = 0 \text{ for all } P \in \ker(\varphi)\}$ and $\mathcal{O} \cong \text{End}(E)$ and $\mathcal{O}' \cong \text{End}(E')$. We will keep this notation I_φ throughout this document.

Special Extremal Order. A *special extremal order* is an order \mathcal{O}_0 in $\mathcal{B}_{p,\infty}$ which contains a suborder of the form $R + jR$, where $R = \mathbb{Z}[\omega] \subset \mathbb{Q}(i)$ is a quadratic order and ω has minimal discriminant. When $p \equiv 3 \pmod{4}$, we have the *special extremal order* $\mathcal{O}_0 = \langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \rangle$, with $i^2 = -1$, $j^2 = -p$ and $k = ij$. It is isomorphic to the endomorphism ring $\text{End}(E_0)$ of the elliptic curve of j -invariant 1728. For the rest of the paper, we fix this *special extremal order* \mathcal{O}_0 , with subring $\mathbb{Z}[\omega]$, and the corresponding elliptic curve E_0 .

Eichler Orders An Eichler order is the intersection of two maximal orders inside $\mathcal{B}_{p,\infty}$. In our settings we consider the case $\mathcal{D} = \mathcal{O}_0 \cap \mathcal{O} = \mathbb{Z} + I$, with $\mathcal{O}_0 \cap \mathcal{O}$ being the endomorphism rings of the supersingular elliptic curves E_0, E linked by the cyclic isogeny $\phi_I : E_0 \rightarrow E$ where the kernel of ϕ_I is $E_0[I] := \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0 \text{ for all } \alpha \in I\}$. Endomorphisms contained in Eichler orders have the nice properties to remain endomorphisms when pushed by ϕ_I from E_0 to E , thus if we consider two equivalent left \mathcal{O}_0 ideals J_1, J_2 of norms coprimes to $\mathfrak{n}(I)$ such that $J_1 = \chi_{J_2}(\beta)$ for $\beta \in J_2 \cap \mathcal{D}$ then the ideals remain equivalent when pushed through I , i.e. $[I]_* J_1 \sim [I]_* J_2$. Because of these properties Eichler orders are essential for performing calculations on maximal orders different from \mathcal{O}_0 , more on this can be read in [21,59,39].

Supersingular j -invariants over \mathbb{F}_{p^2}	Maximal orders in $\mathcal{B}_{p,\infty}$
$j(E)$ (up to Galois conjugacy)	$\mathcal{O} \cong \text{End}(E)$ (up to isomorphism)
(E_1, φ) with $\varphi : E \rightarrow E_1$	I_φ integral left \mathcal{O} -ideal and right \mathcal{O}_1 -ideal
$\theta \in \text{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\text{deg}(\varphi)$	$n(I_\varphi)$
$\hat{\varphi}$	\bar{I}_φ
$\varphi : E \rightarrow E_1, \psi : E \rightarrow E_1$	Equivalent ideals $I_\varphi \sim I_\psi$
Supersingular j -invariants over \mathbb{F}_{p^2}	$\text{Cl}(\mathcal{O})$
$\tau \circ \rho : E \rightarrow E_1 \rightarrow E_2$	$I_{\tau \circ \rho} = I_\rho \cdot I_\tau$
N -isogenies (up to isomorphism)	$\text{Cl}(\mathfrak{D})$, with Eichler order \mathfrak{D} of level N

Table 1: The Deuring correspondence, a summary given in [22].

The Effective Deuring Correspondence. For our applications, we need to make this theoretical correspondence effective as shown in Table 1. In particular, the most important task for us is to take an ideal and compute the corresponding isogeny. There now exists several distinct variants of this algorithm [21,22,40,4,49]. In this work we are going to need two flavours of it. The first one targets the case where the norm is odd, whereas the second one requires the norm of the ideal to be translated to be a long power of 2.

The odd generic case was recently addressed in [4, Algorithm 3] with an algorithm that takes in input a left ideal I , and outputs an efficient way to evaluate $\varphi_I : E \rightarrow E_I$ on any point of $E(\mathbb{F}_{p^2})$. Note that the algorithm described in [4] imposes a strong restriction on the domain E (it needs to be a very specific curve E_0), however, a generic algorithm can be derived from the restricted one by applying it twice, once between E_0 and E and once between E_0 and E_I , and then composing the results to be able to evaluate φ_I . In the rest of this work, we call this algorithm `AnyIdealTolsogeny`, and we assume that it takes an ideal I between maximal orders in $\mathcal{B}_{p,\infty}$ and outputs the domain E and codomain E_I of φ_I , and a representation F of φ_I allowing to evaluate φ_I on any points of $E(\mathbb{F}_{p^k})$ in a polynomial (in $\log p$ and $\log \mathfrak{n}(I)$) number of operations over \mathbb{F}_{p^k} .

An algorithm for the power of two case is at the heart of the signing procedure of SQIsign [21]. It was improved several times since then: first in [22], and then later on in [40,46] using dimension 2 isogenies, we refer to `IdealTolso`(J, I_ϕ) as the algorithm taking as input a left \mathcal{O}_0 -ideal I_ϕ , with $\phi : E_0 \rightarrow E$, and an ideal J such that $\mathcal{O}_L(J) = \mathcal{O}_R(I_\phi)$ of norm a power of 2 that returns the isogeny $\phi_J : E \rightarrow E'$.

We do not provide a full description of all these ideal-to-isogenies algorithms because they are quite technical and the state of the art bit unstable (there have been a lot of recent improvements), and the inner details are not really relevant for our work anyway as we only need to use them as black-boxes. We address the reader to the various references we gave for more details.

We will also need a translation algorithm that works from isogenies to ideal when the degree is a power of two. For that task, we will use the algorithm described in [17, Appendix A.4]. In the rest of this work, this algorithm is denoted by `IsoToIdeal`.

Counting Equivalent Isogenies Thanks to the Deuring correspondence we can associate any left \mathcal{O} -ideal I of reduced norm d to an isogeny $\phi_I : E \rightarrow E'$ with codomain supersingular elliptic curve E with $\text{End}(E) \simeq \mathcal{O}$ of degree d .

Using [59, Lemma 42.2.8] for any $\phi_I : E \rightarrow E'$ we have an isomorphism of left \mathcal{O} -modules:

$$\begin{aligned} \phi_I^* : \text{hom}(E, E') &\rightarrow I \\ \psi &\mapsto \hat{\psi} \circ \phi_I . \end{aligned}$$

Our main tool is the *Van Der Corput's inequality* used on a lattice $\Lambda \subset \mathbb{R}^4$ [15] (an improvement of the Minkowski inequality):

$$\#\{\vec{v} \in \Lambda \mid \|\vec{v}\| \leq r\} \geq 2 \left\lfloor \frac{\pi^2}{32} \frac{r^4}{\text{Vol}(\Lambda)} \right\rfloor + 1 . \quad (2)$$

Given two supersingular elliptic curves E, E' lets label as $\text{Iso}_B(E, E') \subseteq \text{hom}(E, E')$ the set of isogenies in with domain E and codomain E' with prime norm lower than B . We want now to lower-bound the cardinality of $\text{Iso}_B(E, E')$, for that we need to use the normalized norm map $\mathfrak{n}_I(\alpha) = \frac{\mathfrak{n}(\alpha)}{\mathfrak{n}(I)}$, for $\alpha \in I$, to induce a metric on the lattice in such a way that $\mathfrak{n}_I(\phi_I^*(\psi)) = \text{deg}(\psi)$. To effectively use inequality (2) we need to compute the discriminant of the lattice with respect to \mathfrak{n}_I and we proceed as in [34]. Thanks to [59, Theorem 15.5.5] we know that $\det(\mathcal{O}_0) = p^2$ since it is maximal, and by [59, Lemma 15.2.15] we can derive $\det(I)$ from the index of the ideal in the order, since $\det(I) = |\mathcal{O}/I|^2 \det(\mathcal{O}) = \mathfrak{n}(I)^4 \det(\mathcal{O})$. Since we are in dimension 4 lattices we can conclude that with respect to \mathfrak{n}_I the volume of I is p^2 .

To use (2) we consider the norm $\|\cdot\| = \sqrt{\mathfrak{n}_I(\cdot)}$, then we use the bound on $r = \sqrt{B}$ and $\text{Vol}(\Lambda) = p$ (the square root of the discriminant). So we have that there are at least $0.61 \frac{B^2}{p}$ isogenies of degree less than B . Under the heuristic that degrees are distributed uniformly we know that the probability of it being prime can be approximated as $\log(B)^{-1}$ using the prime counting function, so we have at least

$$\frac{0.61}{\log(B)} \frac{B^2}{p} \quad (3)$$

prime degree isogenies in $\text{hom}(E, E')$.

Supersingular Isogeny Graphs. Let $p \geq 5$ be a prime number. For any $\ell \neq p$, we have an ℓ -isogeny graph where each vertex corresponds to the j -invariant of a supersingular graph and each edge corresponds to an ℓ -isogeny between the two vertices (i.e. the supersingular curves). The graph can be viewed as undirected due to the existence of the dual isogeny. An ℓ -isogeny graph is full-connected and $\ell + 1$ -regular.

Moreover, the graph is *Ramanujan* so for a random walk from any vertex in the graph converges to the stationary distribution fast. We conclude the property with the following theorem from [50].

Theorem 2.5. *There is a bound $n = O(\log_\ell(p) + \log_\ell(2^\lambda))$ such that the statistical distance between the stationary distribution over an ℓ -isogeny graph and the end point distribution of a random walk starting from any distribution of length not less than n have statistical distance is negligible bounded by $\text{negl}(\lambda)$.*

The recent work [3] showed that *Ramanujan* property holds also for the ℓ -isogeny graph of elliptic curves with d Borel level structures, i.e. the graph with vertices the pairs (E, ϕ) with E a supersingular elliptic curve and $\phi : E \rightarrow \star$ a cyclic isogeny of degree d not divisible by p , up to isomorphism, with edges the ℓ -isogenies linking the curves and pushing the d -isogenies one to the other.

Theorem 2.6 (Theorem 11 [3]). *Given any distribution π on the ℓ -isogeny graph of elliptic curves with d Borel level structures, then the statistical distance between the distribution obtained after a random walk of length k and the stationary distribution on the graph is bounded by:*

$$\frac{\sqrt{3}}{2} K^{-1} \frac{(\ell+1)(k+1) - 2}{(\ell+1)\sqrt{\ell^k}}, \quad (4)$$

with $K = \left(\frac{(p-1)d}{12} \prod_q \left(1 + \frac{1}{q}\right) \right)^{-1/2}$, for q ranging over the prime divisors of d .

As shown in [34,25,50], there are many equivalent forms of the original isogeny problem on the supersingular isogeny graph. Here we consider the two following problems, on which our schemes based. Notably, they are equivalent to original isogeny problem [25,50].

Problem 2.7. Given E_0 where $\text{End}(E_0)$ is a special extremal order in $B_{p,\infty}$, the *supersingular endomorphism ring problem* on a supersingular elliptic curve E requires to find an ideal I which is a left- $\text{End}(E_0)$ ideal and a right- $\text{End}(E)$ ideal.

Problem 2.8. The *supersingular endomorphism problem* on a supersingular elliptic curve E requires to find a non-scalar smooth endomorphism $\alpha : E \rightarrow E$.

3 Zero-Knowledge for SQISign

In this section, we investigate the zero-knowledge property of the SQISign identification scheme. The main result of this section is to present an efficient distinguisher for SQISign when the secret key is given.

To start, we sketch the SQISign scheme. The core of the protocol is to prove knowledge of the endomorphism ring of a curve E_{pk} . The prover commits to another curve E_{cmt} and receives a challenge isogeny $\phi_{\text{ch}} : E_{\text{cmt}} \rightarrow E_{\text{ch}}$. The prover must then provide a cyclic isogeny from E_{pk} to E_{cmt} that factors through ϕ_{ch} . Also, we sketch the simulator as follows: it computes a random isogeny $\sigma' : E_{\text{pk}} \rightarrow E'_{\text{ch}}$ of a specific degree and then computes a random challenge isogeny $\hat{\phi}'_{\text{ch}} : E'_{\text{ch}} \rightarrow E'_{\text{cmt}}$ of a specific degree. The simulator outputs a simulated transcript as $(E'_{\text{cmt}}, \hat{\phi}'_{\text{ch}}, \sigma')$. For a more detailed description, refer to [21].

Clearly, the simulation above is not *special* HVZK because the challenge ϕ_{ch} necessitates the existence of the curve E'_{ch} in advance for the simulation. The limitation does not constitute any immediate issue for a signature scheme but rendering it unusable to define a ring signature with Algorithm 1 for instance. Then, we will show that it is *computationally weak* HVZK instead of the strong one. To understand these limitations we go more into detail for the core procedure `SigningKLPT`, recalling some security results from literature and showing how to distinguish simulated transcripts when having access to the secret key.

Procedure of SigningKLPT. Algorithm 2 is the main signing algorithm of SQISign introduced in [21], which can be viewed as a modification of the KLPT algorithm from [34]. The algorithm consists of four main subroutines:

- `EquivalentRandomEichlerIdeal`(I, N_ζ): on input a left \mathcal{O} -ideal and $N_\zeta \in \mathbb{N}$ returning an equivalent ideal uniformly distributed in the class set $\text{Cl}_{\mathcal{D}}(\mathcal{O})$ of norm coprime to N_ζ ;
- `EquivalentPrimeIdeal` $_{\mathcal{O}_0}(I)$: on input a left- \mathcal{O}_0 ideal returning the smallest equivalent ideal to I of prime norm;
- `FullRepresentInteger` $_{\mathcal{O}_0}(M)$: on input $M \in \mathbb{Z}$ and $M > p$ returning $\gamma = x + yi + z \frac{i+j}{2} + t \frac{1+k}{2}$ with $\mathfrak{n}(\gamma) = M$;
- `FullStrongApproximation` $_{\mathbb{F}}(C, D, N)$: on input $C, D \in \mathbb{Z}$, a (semi)prime N , returning $\mu_1 \in \mathcal{O}_0$ such that $\mu = \lambda(C + iD)j + N\mu_1$ has norm dividing \mathbb{F} .

The parameters e_0, e_1 , and $e = e_0 + e_1$ are properly chosen to ensure the termination of the algorithm. Going into details about the particular algorithms involved is out of the scope of the paper, so we refer the reader to [21,12,22] for a more detailed treatment.

Algorithm 2 SigningKLPT(I, I_ζ)

Require: I left \mathcal{O} -ideal, I_ζ left \mathcal{O}_0 -ideal, right \mathcal{O} -ideal of coprime norms.

Ensure: $J \sim I$ of norm ℓ^e .

- 1: Get $\mathcal{C} \leftarrow \text{EquivalentRandomEichlerIdeal}(I, N_\zeta)$; \triangleright Uniformly distributed in $\text{Cl}_{\mathcal{D}}(\mathcal{O})$
 - 2: Set $\mathcal{C}' \leftarrow [I_\zeta]_* \mathcal{C}$;
 - 3: Get $L \leftarrow \text{EquivalentPrimeIdeal}_{\mathcal{O}_0}(\mathcal{C}')$ of prime norm $N_L = \mathfrak{n}(L)$;
 - 4: Store $\delta \in \mathcal{C}'$ such that $L = \chi_{\mathcal{C}'}(\delta)$;
 - 5: Compute $\gamma \leftarrow \text{FullRepresentInteger}_{\mathcal{O}_0}(N_L \ell^{e_0})$;
 - 6: Find $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N_L \mathbb{Z})$ with $\gamma j(C_0 + iD_0) \in L$;
 - 7: Find $(C_1 : D_1) \in \mathbb{P}^1(\mathbb{Z}/N_\zeta \mathbb{Z})$ such that $\gamma j(C_1 + \omega D_1) \delta \in \mathbb{Z} + I_\zeta = \mathcal{D}$;
 - 8: Compute $C = \text{CRT}_{N_\zeta, N}(C_0, C_1)$ and $D = \text{CRT}_{N_\zeta, N}(D_0, D_1)$
 - 9: Fix $N \leftarrow N_L N_\zeta$ and $e_1 = e - e_0$;
 - 10: Get $\mu \leftarrow \text{FullStrongApproximation}_{\ell^{e_1}}(C, D, N)$ $\triangleright \mu = \lambda(C + iD)j + N\mu_1$ of norm ℓ^{e_1}
 - 11: Set $\beta = \gamma\mu$; $\triangleright \mathfrak{n}(\beta) = N\ell^e$, $\beta \in L$ and $\beta\delta \in \mathcal{D}$
 - 12: **return** $J = [I_\zeta]_* \chi_L(\beta)$ $\triangleright J = [I_\zeta]_* \chi_{\mathcal{C}'}(\beta\delta)$
-

Distribution of SigningKLPT's outputs. We recall the characterization of the distribution of the output of SigningKLPT from [21, Section 7.2] and [22, Section 6]. Let $\zeta : E_0 \rightarrow E$ of degree N_ζ , \mathcal{O} be the endomorphism ring of E and \mathcal{D} be the Eichler order $\mathcal{O}_0 \cap \mathcal{O}$. We consider $\mathcal{U}_{\mathcal{C}, N_\zeta}$, equivalently \mathcal{U}_{L, N_ζ} , the set of all isogenies ι from the isomorphism class of E_0 of degree D_{rsp} such that $\hat{\iota} \circ \phi_L = \beta \in L$ where \mathcal{C} and L are the intermediate variables defined as in Algorithm 2.

Proposition 3.1 (Prop 10 and Lemma 14 [21]). *The set*

$$\mathcal{P}_{N_\zeta} := \bigcup_{\mathcal{C} \in \text{Cl}(\mathcal{O}_0)} \mathcal{U}_{\mathcal{C}, N_\zeta} \quad (5)$$

can be computed from the sole knowledge of N_ζ . Moreover, under the heuristic assumptions from [21, Section 7.3] the output distribution of SigningKLPT(I, I_ζ), on input I drawn uniformly from the non-trivial classes in $\text{Cl}(\mathcal{O})$, is statistically indistinguishable from the uniform distribution on the set

$$\{[I_\zeta]_* I_L \mid \iota \in \mathcal{P}_{N_\zeta}\} .$$

Hence, the security assumption of SQsign zero-knowledge can thereby be summarised as follows:

Problem 3.2. Let p be a prime, and D_{rsp} a smooth integer. Let $\zeta : E_0 \rightarrow E$ be a random isogeny drawn from a probability distribution on the set of cyclic isogenies with domain E_0 of degree N_ζ .

The problem is, given $p, D_{\text{rsp}}, E, N_\zeta$, to distinguish which of the following cases is, given access to an oracle that outputs isogenies $\sigma : E \rightarrow \star$ of degree D_{rsp} sampled uniformly at random:

1. From a set of cyclic isogenies of degree D_{rsp} ; or
2. From $[\zeta]_* \mathcal{P}_{N_\zeta}$, where \mathcal{P}_{N_ζ} is defined in eq. (5).

Our distinguisher is based on two following facts.

1. L , the output of $\text{EquivalentPrimeIdeal}_{\mathcal{O}_0}(\mathcal{C}')$, is deterministic up to $\mathcal{C} \in \text{Cl}_{\mathcal{D}}(\mathcal{O})$ drawn in Line 1;
2. β computed in Line 11 as $\gamma \cdot \mu$ has the following traits:
 - (a) γ is one of the possible output of $\text{FullRepresentInteger}(N\ell^{e_0})$ with $N_L = \mathfrak{n}(L)$ (Line 5). Note that from [34] we can estimate the number of solutions to

$$\frac{\sqrt{N_L \ell^{e_0}}}{\sqrt{p} \log(p) h(R)} \quad (6)$$

with $h(R)$ the class number of the ring $R = \mathbb{Z} \oplus \mathbb{Z}i$;

- (b) $\mu = (C + iD)j \in Rj \pmod{NN_\zeta \mathcal{O}_0}$, with $p(C^2 + D^2)\ell^{e_1}$ being a quadratic residue modulo $N_L N_\zeta$ and $\gamma(C + iD)j \in L$ modulo N_L (Line 6).

Theorem 3.3. *Let the parameters to be specified as `SQISign` ([12]). There exists a polynomial-time algorithm solving Problem 3.2 with only one query when the secret $\zeta : E_0 \rightarrow E$ is given. Equivalently if ζ given, we can distinguish a simulated transcripts from the real ones.*

Proof. Let $\sigma : E \rightarrow E_{\text{ch}}$ be the received isogeny, thanks to ζ we can compute the pullback $\iota = [\zeta]^* \sigma$ and the associated ideal I_ι using `IsoToIdeal`. Even though \mathcal{P}_{N_τ} has a size of $\tilde{\Theta}(pN_\tau)$, which is exponentially large in the security parameter λ , it remains a set of negligible cardinality relative to the number of cyclic isogenies of degree D_{rsp} due to the choice of parameter. Therefore, for a uniformly sampled cyclic isogeny σ of degree D_{rsp} , the probability that $\iota \in \mathcal{P}_{N_\tau}$ is negligible. Hence, to solve Problem 3.2, we only need to efficiently verify whether $\iota \in \mathcal{P}_{N_\tau}$ or not.

Observe that the output I' of `SigningKLPT` is $[I_\zeta]_* \chi_L(\beta)$ for $\beta \in \mathcal{D}$ (where L is defined in Line 3), so, if actually $\iota \in \mathcal{P}_{N_\tau}$ we can compute the deterministic prime norm ideal $L = \text{EquivalentPrimeIdeal}(I_\iota)$, that only depends on the equivalence class of I_ι . We define β as the unique element of L such that $\chi_L(\beta) = I_\iota$.

Then, given L , we can enumerate in polynomial time all possible $\gamma \leftarrow \text{FullRepresentInteger}(N\ell^{e_0})$ where the number of possible solutions of γ is polynomial thanks to Equation (6). For each possible γ we can compute $\mu \leftarrow \gamma^{-1}\beta$. If $\iota \in \mathcal{P}_{N_\tau}$ then $\mu \pmod{(NN_\tau)} \in Rj$ and we can rewrite it as $Ci + Dij$. Then we check if it satisfies the condition from step Item 2b. If this is true we output that σ comes from the second isogeny.

Remark 3.4. Even in the case in which we are given another connecting isogeny $\kappa : E_0 \rightarrow E$ we can still recover $\zeta : E_0 \rightarrow E$ if N_ζ is smaller than \sqrt{p} . In fact in this case, with high probability, ζ is the isogeny of smallest degree connecting $E_0 \rightarrow E$, that can be recovered by looking at the shortest vector in the ideal I_κ .

We provide our proof-of-concept implementation in SageMath. The experimental validates Theorem 3.3 and only requires a few seconds to distinguish. Our proof-of-concept implementation in SageMath can be found in `giacomoborin/RingSQISign-poc`. Note that the folder is written as a fork of the SageMath `SQISign` implementation `LearningToSQI/SQISign-SageMath` that provides the mathematical functionalities used in the scheme.

4 Linear Ring Signatures

The primary goal of this section is to present a fully anonymous ring signature. To achieve this, we introduce a new KLPT variant for the signing algorithm to address the distinguishability issues described in the previous section. Subsequently, we propose a new `SQISign` variant that offers a better zero-knowledge guarantee with mild overhead. To apply the AOS construction (Algorithm 1), we slightly modify the `SQISign` framework to enable the simulation of a signature with a given challenge isogeny. Finally, we then provide an analysis of the new zero-knowledge assumption.

These two modifications are independent of each other. We begin with the simple modification of the `SQISign` framework, which is more straightforward, to streamline the presentation.

Remark 4.1. As described in Section 1, the `SQISignHD` protocol [17], a variant of `SQISign` that leverages efficient representations involving higher-dimensional isogenies, cannot be used to construct ring signatures using known paradigms in the literature. This limitation boils down to its simulation procedure requiring the uniform generation of random isogenies of arbitrary degrees, and there is no known method to construct such an oracle in polynomial time. The same limitation applies to other HD variants [4,46,24].

4.1 Computational Special Zero-knowledge Variant

There are several ways to get a `SQISign` variant with *Special Honest-verifier Zero-knowledge*, arguably the simpler one is to generate the challenge starting from the public key E_{pk} as in Figure 1. The choice to compute

the challenge isogeny from the commitment curve in the original scheme was probably motivated by efficiency, but it is not at all a necessity, for example in other SQIsign variants relying on higher dimensional isogenies we are already dealing with a diagram as in Figure 1, like [4,52]. For the one dimensional case, the protocol can be modified as follows.

Protocol 1 *We assume Setup is run as in SQIsign, then:*

- **Key Generation:** Sample a random isogeny walk $\tau_{\text{sk}} : E_0 \rightarrow E_{\text{pk}}$ of degree $\approx p$ and return the public key E_{pk} and the secret key τ_{sk} ;
- **Commitment:** Sample a random isogeny walk $\psi : E_0 \rightarrow E_{\text{cmt}}$ of prime degree bounded by an integer B_{cmt} and commit to E_{cmt} .
- **Challenge:** Sample a random isogeny $\phi_{\text{ch}} : E_{\text{pk}} \rightarrow E_{\text{ch}}$ of degree $D_c \approx 2^\lambda$ and send ϕ_{ch} as a challenge;
- **Response:** Consider the ideal I associated to the composition $\phi_{\text{ch}} \circ \tau_{\text{sk}} \circ \hat{\psi}$, get an equivalent ideal of power-smooth norm D_{rsp} via executing $\text{SigningKLPT}(I, I_\psi)$. Then translate it to an isogeny $\sigma : E_{\text{cmt}} \rightarrow E_{\text{ch}}$ such that $\hat{\phi}_{\text{ch}} \circ \sigma$ of degree D_{rsp} and being cyclic. Then, return σ as response.
- **Verification:** Check that σ is an isogeny from E_{cmt} to the expected codomain E_{ch} and $\hat{\phi}_{\text{ch}} \circ \sigma$ is cyclic.

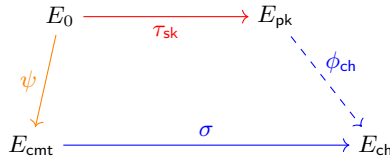


Fig. 1: Special variant.

To analyze the underlying security of Protocol 1 we consider a slightly different characterization analogue to Problem 3.2 to take into account the runtime generation of the connecting isogeny.

Problem 4.2. Let p be a prime, and D_{rsp} a smooth integer and a supersingular elliptic curve E' . The problem ask to distinguish between isogenies $\sigma : E \rightarrow E'$ of degree D_{rsp} sampled either

1. uniformly random between cyclic isogenies of degree D_{rsp} with E a uniformly random supersingular elliptic curve;
2. uniformly random in $[\zeta]_* \mathcal{P}_{N_\zeta}$, where \mathcal{P}_{N_ζ} is defined in (5), where $\zeta : E_0 \rightarrow E$ is a random isogeny drawn from a probability distribution on the set of cyclic isogenies with domain E_0 , and N_ζ is its degree.

Proposition 4.3. *Protocol 1 is correct, special sound for the relation defined in Problem 2.8, and computationally special weak honest-verifier zero-knowledge basing on the hardness of Problem 4.2 and the heuristic assumptions from [21, Section 7.3].*

Proof. The correctness is implied by the correctness of the SigningKLPT procedure. We prove the special soundness essentially in the same way as in [21]. Given two valid transcripts for the same commitment: $(E_{\text{cmt}}, \phi_{\text{ch}}, \sigma)$ and $(E_{\text{cmt}}, \phi'_{\text{ch}}, \sigma')$ with $\phi_{\text{ch}} \neq \phi'_{\text{ch}}$, then the composition

$$\hat{\phi}'_{\text{ch}} \circ \sigma' \circ \hat{\sigma} \circ \phi_{\text{ch}}$$

is a by definition an endomorphism of smooth degree $(D_c D_{\text{rsp}})^2$. We claim this is a non-scalar endomorphism. Since they are valid transcripts the endomorphism is a composition of cyclic isogenies $\hat{\phi}'_{\text{ch}} \circ \sigma'$ and the dual of $\hat{\phi}_{\text{ch}} \circ \sigma$. Suppose for the purpose of a contradiction that the endomorphism is scalar, then we have

$$\hat{\phi}'_{\text{ch}} \circ \sigma' = \hat{\phi}_{\text{ch}} \circ \sigma ,$$

due to the uniqueness of the dual isogeny. This contradicts to the fact that $\phi_{\text{ch}} \neq \phi'_{\text{ch}}$. Hence, the protocol has special soundness for the relation defined by the Supersingular Endomorphism Problem on E_{pk} (Problem 2.8).

Regarding computational special zero-knowledge, we first consider the simple simulator $\mathcal{S}(E_{\text{pk}}, D_{\text{rsp}}, \phi_{\text{ch}})$ as follows. On input an arbitrary challenge $\phi_{\text{ch}} : E_{\text{pk}} \rightarrow E_{\text{ch}}$, it samples uniformly $\sigma' : E_{\text{ch}} \rightarrow E$ of degree D_{rsp} and outputs $(E, \phi_{\text{ch}}, \sigma')$. We consider now Problem 4.2 with $E' = E_{\text{ch}}$. The simulator distribution is clearly the same as the first distribution of Problem 4.2, while the original output is equivalent to second distribution due to Proposition 3.1 under the same heuristic assumptions as **SQSign**. \square

It is clear that this simple variant is still not “strong” HVZK for the same reason described in the previous section. Indeed, the commitment isogeny ψ can be recovered from the knowledge of the endomorphism ring of E_{pk} , we can thereby construct a distinguisher, with access to the secret key, using again Theorem 3.3 to distinguish simulated transcripts.

4.2 Connecting Isogeny Randomization

This subsection presents a variant of the KLPT algorithm providing a better zero-knowledge guarantee and the strong HVZK property. Our key idea here is to randomize the connecting ideal between the starting curve E_0 and the commitment one E_{cmt} prior to **SigningKLPT**. In this way, we will have exponentially many possible connecting isogeny candidates. From the perspective of Deuring correspondence, this implies that even if we know the endomorphism ring of E_{cmt} because of $\hat{\sigma} \circ \phi_{\text{ch}} \circ \tau_{\text{sk}}$, we still cannot individuate the connecting ideal used for **SigningKLPT**.

Accordingly, we can implement this strategy by modifying Protocol 1 in the response phase. Prior to the execution of **SigningKLPT**, we take a uniformly random equivalent ideal to I_ψ of prime norm bounded by B_{cmt} , which corresponds to a random connecting isogeny of prime degree. We select B_{cmt} in such a way that there are exponentially many possibilities. We summarize this process via a new procedure **RSigningKLPT** in Algorithm 3.

Algorithm 3 **RSigningKLPT**(J, I, B_{cmt})

Require: A left \mathcal{O} -ideal J , and I , a left \mathcal{O}_0 -ideal and right \mathcal{O} -ideal, $B_{\text{cmt}} > \sqrt{p}$

Ensure: $J' \sim J$ of norm ℓ^e

- 1: Compute a reduced basis $\{\alpha_1, \dots, \alpha_4\}$ of I ,
- 2: Fix m_i accordingly to the basis;
- 3: **repeat**
- 4: Sample $x_i \in [-m_i, m_i]$ for $i = 1, 2, 3, 4$;
- 5: Set $\alpha \leftarrow \sum_{i=1}^4 x_i \alpha_i$
- 6: **until** $n(\alpha) \leq n(I)B_{\text{cmt}}$
- 7: Set $I_\psi \leftarrow I \frac{\alpha}{n(I)}$;
- 8: Set $J \leftarrow \frac{\alpha}{n(I)} J$;
- 9: **return** **SigningKLPT**(J, I_ψ).

\triangleright We get $\psi \in \text{Iso}_{B_{\text{cmt}}}(E_0, \psi)$
 \triangleright Ensure $\mathcal{O}_L(J) = \mathcal{O}_R(I_\psi)$

Selection of m_i The selection of the integers m_i in Line 2 is crucial both to achieve efficiency and a uniform distribution of ψ in $\text{Iso}_{B_{\text{cmt}}}(E_0, E_{\text{cmt}})$. Since by triangular inequality we know that $n(\alpha) \leq \sum_{i=1}^4 m_i^2 \lambda_i$, with $\lambda_i = n(\alpha_i)$, we can expect to fix m_i slightly larger than $\sqrt{B_{\text{cmt}} n(I) / 4 \lambda_i}$ (say twice the size). To do that we consider the *Gram-Schmidt orthogonalization* $\{\alpha_1^*, \dots, \alpha_4^*\}$ and the *Gram-Schmidt coefficient* μ_{ij} for our basis (see e.g. [47]), with respect to the inner product induced by the reduced trace, in such a way that

$$\alpha_i = \alpha_i^* + \sum_{j < i} \mu_{ij} \alpha_j^* .$$

Then we fix $m_4 = \sqrt{B_{\text{cmt}}n(I)/\lambda_4^*}$ and recursively $m_i = \sqrt{B_{\text{cmt}}n(I)/\lambda_i^*} + \sum_{j>i} |\mu_{ji}| m_j$ for $i = 3, 2, 1$ (with $\lambda_i^* = n(\alpha_i^*)$). In this way we can ensure that if $n(\alpha) \leq B_{\text{cmt}}n(I)$ then $|x_i| \leq m_i$. Anyway for our subsequent security reduction we need the following assumption:

Assumption 1 *The distribution of the isogenies obtained at Line 7 are statistically close to the uniform distribution on $\text{Iso}_{B_{\text{cmt}}}(E_0, E_{\text{cmt}})$.*

4.3 Security.

We now argue about the *strong* HVZK property of the new protocol instantiated with this additional randomization step by reducing it to a new version of Problem 4.2 that consider the new randomization step. Then we analysis why the knowledge of a connecting isogeny cannot lead to a distinguisher attack as for Theorem 3.3.

Problem 4.4. Let $p \equiv 1 \pmod{4}$ be a prime, B_{cmt} an integer, D_{rsp} a smooth integer. Given E_0 elliptic curve of j -invariant 1728 and E a random curve of known endomorphism ring \mathcal{O} , and a cyclic left \mathcal{O} -ideal I_η of norm D_{rsp} , distinguish from which of the two set I_η has been uniformly sampled:

- all cyclic left \mathcal{O} -ideals of norm D_{rsp} ;
- the union:

$$\{[I_\psi]_* I_\iota \mid \iota \in \mathcal{P}_{\text{deg}(\psi)}, \psi \in \text{Iso}_{B_{\text{cmt}}}(E_0, E)\} . \quad (7)$$

where $\mathcal{P}_{\text{deg}(\psi)}$ is defined as in (5).

Lemma 4.5. *For Protocol 1 used with Algorithm 3 the strong computational HVZK reduces to the hardness of Problem 4.4 if the heuristic assumptions from [21, Section 7.3] and Assumption 1 hold.*

Proof. To prove the proposition we consider the same simulator from the proof of Proposition 4.3 that outputs a uniformly random isogeny of degree D_{rsp} and we show that a distinguisher for this simulator, aided with the secret key $\tau_{\text{sk}} : E_0 \rightarrow E_{\text{pk}}$, can be render to a distinguisher for Problem 4.4. Given the curve E and an input left \mathcal{O} -ideal I_ζ we can translate it to a cyclic isogeny $\zeta : E \rightarrow E'_{\text{ch}}$, we generate then an isogeny $\hat{\phi}'_{\text{ch}} : E'_{\text{ch}} \rightarrow E'_{\text{pk}}$ and we compute a connecting isogeny $\tau'_{\text{sk}} : E_0 \rightarrow E'_{\text{pk}}$ using the knowledge of the endomorphism ring. So we have the valid transcript $(E, \phi'_{\text{ch}}, \zeta)$ associated to the key $\tau'_{\text{sk}} : E_0 \rightarrow E'_{\text{pk}}$ that we can feed to the HVZK distinguisher.

We argue now about the statistical indistinguishability of the inputs for the two cases. While it is clear that for the uniformly random one the distributions are the same, while for the second one we need to argue that the output of Algorithm 3 is statistically indistinguishable from the uniform distribution on the set from (7). This is immediate from the construction of the protocol, in fact the set is indexed over the connecting isogenies of prime bounded degree $\text{Iso}_{B_{\text{cmt}}}(E_0, E)$, that results from the first randomization step in Algorithm 3 and are statistically indistinguishable thanks to Assumption 1. Fixed an isogeny ψ , so equivalently an ideal I_ψ , we need to prove that the output ideal distribution of SigningKLPT is statistically indistinguishable from the uniform distribution on:

$$\{[I_\psi]_* I_\iota \mid \iota \in \mathcal{P}_{\text{deg}(\psi)}\} ,$$

that is implied by Proposition 4.3 under the assumptions from [21, Section 7.3]. □

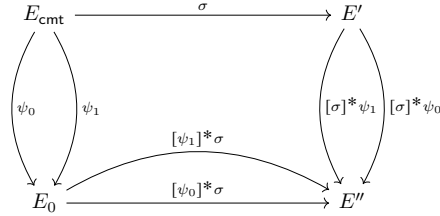
Security Analysis We argue now about the hardness of Problem 4.4. The set in (7) is constructed by using multiple of pushforward of the sets \mathcal{P}_{N_ζ} , so all the security arguments from [21, Appendix B] and [22] still applies. Also, the hardness of the problem is immediately related to the size of the set $\text{Iso}_{B_{\text{cmt}}}(E_0, E)$, so to the bound B_{cmt} . In fact we can consider as a distinguisher that search through all the possible connecting isogeny of prime degree and apply theorem 3.3 (let C be the computational cost of this attack). The computational cost of this algorithm is

$$O(\#\text{Iso}_{B_{\text{cmt}}}(E_0, E) \cdot C) .$$

For this reason we choose B_{cmt} accordingly to (3) such that $\#\text{Iso}_{B_{\text{cmt}}}(E_0, E)$ is exponential, in particular we go for the conservative choice of taking it bigger than 2^λ .

We analyse further how the randomization thwarts the distinguisher. Note that the first step of the distinguisher is the recover of the prime norm ideal L obtained from $\text{EquivalentPrimeIdeal}(\mathcal{C})$. However the correctness of this step relies on individuating the correct equivalence class \mathcal{C} via $[I_\psi]_* I_\sigma$, but this happens only with negligible probability if we are pulling through a random connecting isogeny. To see this consider the response isogeny $\sigma : \psi \rightarrow E'$ of large degree D_{rsp} , two random isogenies $\psi_0, \psi_1 \in \text{Iso}_{B_{\text{cmt}}}(E_0, E)$ (let ψ_0 be the one used to generate the response) and the ideals I_0, I_1 associated to the pullback isogenies $[\psi_0]^* \sigma, [\psi_1]^* \sigma$.

Thanks to the Deuring correspondence we know that the ideals I_0, I_1 are equivalent if and only if $[\psi_0]^* \sigma, [\psi_1]^* \sigma$ share the same codomain, but, by the commutativity property of the diagrams, this is equivalent to ask that $[\sigma]^* \psi_0, [\sigma]^* \psi_1$ share the same codomain, as you can see from the following scheme.



Now, observe that the pushforward isogenies $[\sigma]^* \psi_0, [\sigma]^* \psi_1$ distribution is statistically indistinguishable from the uniform distribution of cyclic isogenies with the same respective degrees. To prove this, since we have that $D_{\text{rsp}} = \ell^e = O(p^3 B_{\text{cmt}}^3)$ (more on that in Section 4.4), we can use Theorem 2.6 and bound the statistical distance from the uniform distribution is bounded by the negligible quantity

$$O\left(\frac{p B_{\text{cmt}}}{\sqrt{p^3 B_{\text{cmt}}^3}}\right) = O\left((p B_{\text{cmt}})^{-\frac{1}{2}}\right).$$

Hence, since the pushforward isogenies are random walks in the supersingular isogeny graph of degree bigger than \sqrt{p} , also their domains distributions are statically close to uniform by Theorem 2.5. We conclude that I_0 and I_1 are equivalent with negligible probability. Suppose however that in some way we recover even partial information about the ideal L or the involved endomorphism, we still cannot use it to start a *meet-in-the-middle* like attack since the isogenies are of prime degree. Given the previous discussion we conclude that we have reasonable evidence that Problem 4.4 is hard.

4.4 Our Ring Signature Construction

We can finally define our first ring signature **Erebor** obtained by applying Algorithm 1 to Protocol 1. Observe that both the optimizations from [55,22] known for KLPT-based **SQI**sign can be used for our ring signature. We have in Algorithm 4 a *full-anonymous* version **Erebor-full** that uses Algorithm 3. To sample the challenge we use an hash function H that on input a curve E' , a ring R , a message M , and a curve E returns the coordinates of a cyclic subgroup of E of order D_c with respect to a deterministically generated torsion basis of $E[D_c]$. For compactness we do not go into details on the various translations between kernels, isogenies and ideals.

As for [21] we take a prime $p \approx 2^{2\lambda}$ to ensure the hardness of Problem 2.8. The public keys are the j -invariant of the supersingular curves (so we need $N \cdot 4 \log_2(p)$ bits to represent them). Each transcript of Protocol 1 $(E_{\text{cmt}}, \phi_{\text{ch}}, \sigma)$ can be compressed to $(\ker(\phi_{\text{ch}}), \hat{\sigma})$ since $\ker(\phi_{\text{ch}})$ is a subgroup of E_{pk} and the commitment E_{cmt} can be recovered as the codomain of $\hat{\sigma}$. Thus the signature size is:

$$\underbrace{\text{cost}(\phi_{\text{ch}})}_{\text{ch}_1} + N \cdot \underbrace{\text{cost}(\hat{\sigma})}_{\text{rsp}_\bullet}. \quad (8)$$

We can bound the final degree D_{rsp} for the response σ using Lemma 11. of [21], so

$$D_{\text{rsp}} = 3 \log(p) + 3 \log(B_{\text{cmt}}) + O(\log \log(p)) .$$

We fix $B_{\text{cmt}} \geq 2^{\frac{3}{2}} p^{\frac{1}{2}}$ (in this way from the estimation 3 we have at least 2^λ prime degree equivalent isogenies) and get $\log(D_{\text{rsp}}) \approx \frac{9}{2} \log(p) + \frac{\lambda}{2} + O(\log \log(p))$.

Note that if we are not interested in achieving *full-anonymity* we do not need to use Algorithm 3 at Line 9 and we can actually employ in the commitment generation (Line 1) the same procedure used in the optimized KeyGen from [21, Section 8.3]. In this way we get a much smaller degree for the commitment isogeny $N_\psi \approx p^{\frac{1}{4}}$, so the response degree is again $\log(D_{\text{rsp}}) \approx \frac{15}{4} \log(p)$, resulting in a shorter signature. We refer to this construction as *Erebor-short*.

Thanks to the previous discussions on Protocol 1 we can finally prove:

Theorem 4.6. *The Erebor-full (resp., short) ring signature scheme is unforgeable and full-anonymous (resp., anonymous) over the programmable random oracle model if Problems 4.4 and 2.8 (resp., Problems 4.2 and 2.8) are hard under the same heuristic assumptions from [21, Section 7.3] and Assumption 1.*

Proof. Regarding *Erebor-full*, since the underlying sigma protocol (Protocol 1) is special sound and strong special HVZK by Proposition 4.3. Also, *Erebor-short* has weak special HVZK as explained above. By Proposition 2.4, we prove the results. \square

Remark 4.7. We point out that a similar linear ring signature construction can be achieved using the parallel OR-proof [16], however this would lean an increase in signature length by a factor of $(N - 1)\text{cost}(\phi_{\text{ch}})$.

Algorithm 4 Erebor-full Ring Signature Scheme

RS.Setup(1^λ) :

- 1: Get SQI-friendly prime $p \approx 2^{2\lambda}$
- 2: Set $B_{\text{cmt}} \geq 2^{\frac{3}{2}} p^{\frac{1}{2}}$;
- 3: **return** (p, B_{cmt}) .

RS.Sign($I_{\tau_{\text{sk}}}^{(l)}$, $M, R = \{E_{\text{pk}}^{(1)}, \dots, E_{\text{pk}}^{(N)}\}$) :

- 1: Sample $\psi : E_0 \rightarrow E_{\text{cmt}}^{(l)}$ and I_ψ ;
 \triangleright For short take $N_\psi \approx p^{1/4}$
 - 2: Set $K_{l+1} \leftarrow H(E_{\text{cmt}}^{(l)}, R, M, E_{\text{pk}}^{(l+1)})$;
 - 3: **for** $i = l + 1, \dots, N, 1, \dots, l - 1$ **do**
 - 4: Get $\phi_{\text{ch}}^{(i)} : E_{\text{pk}}^{(i)} \rightarrow E_{\text{ch}}^{(i)}$
 - 5: Sample $\hat{\sigma}^{(i)} : E_{\text{ch}}^{(i)} \rightarrow E_{\text{cmt}}^{(i)}$;
 - 6: Set $K_{i+1} \leftarrow H(E_{\text{cmt}}^{(i)}, R, M, E_{\text{pk}}^{(i+1)})$;
 - 7: Get $\phi_{\text{ch}}^{(l)} : E_{\text{pk}}^{(l)} \rightarrow E_{\text{ch}}^{(l)}$ and $I_{\phi_{\text{ch}}^{(l)}}$;
 - 8: Set $J \leftarrow \bar{I}_\psi \cdot I_{\tau_{\text{sk}}}^{(l)} \cdot I_{\phi_{\text{ch}}^{(l)}}$
 - 9: Get $J' \leftarrow \text{RSigningKLPT}(J, I_\psi, B_{\text{cmt}})$;
 \triangleright For short use SigningKLPT
 - 10: Get $\sigma^{(l)} \leftarrow \text{IdealTolso}(J', I_\psi)$;
 - 11: **return** $\sigma = (K_1, \hat{\sigma}^{(1)}, \dots, \hat{\sigma}^{(N)})$.
-

RS.KeyGen(pp) :

- 1: Sample $I_{\tau_{\text{sk}}}$ of norm $2^{2\lambda}$
- 2: Compute $\tau_{\text{sk}} : E_0 \rightarrow E_{\text{pk}}$;
- 3: **return** $(E_{\text{pk}}, I_{\tau_{\text{sk}}})$.

RS.Verify(R, m, σ) :

- 1: **for** $i = 1, \dots, r$ **do**
 - 2: Check $\sigma^{(i)}$ degree;
 - 3: Get $\phi_{\text{ch}}^{(i)} : E_{\text{pk}}^{(i)} \rightarrow E_{\text{ch}}^{(i)}$ from K_i ;
 - 4: Compute $\hat{\sigma}^{(i)} \circ \phi_{\text{ch}}^{(i)} : E_{\text{pk}}^{(i)} \rightarrow E^{(i)}$;
 - 5: **if** $\hat{\sigma}^{(i)} \circ \phi_{\text{ch}}^{(i)}$ is not cyclic **then**
 - 6: | **return reject**;
 - 7: $K_{i+1} \leftarrow H(E^{(i)}, R, M, E_{\text{pk}}^{(i+1)})$;
 - 8: **if** $K_1 = K_{N+1}$ **then**
 - 9: | **return accept**.
 - 10: **else**
 - 11: | **return reject**.
-

5 Logarithmic Ring Signatures

In this section we describe how to construct logarithmic ring signatures based on the endomorphism ring problem (Problem 2.7).

Unlike our other construction introduced in Section 4, this ring signature protocol is not directly based on SQIsign. In fact, it is conceptually closer to the GPS signature [29], and we can improve it with the recent algorithmic improvements obtained by the use of higher dimensional isogenies and in particular the recent algorithm to translate ideal to isogenies from [4]. This algorithm is the most important subroutine of an efficient method `OrderToCurve` to compute the curve associated through the Deuring correspondence to a given maximal order that is at the heart of our signing method.

5.1 Compressed Endomorphism Ring Representation.

In this section, we give a heuristic method to encode the isomorphism class of maximal order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ in approximately $\log p$ bits. This method works very well in practice and is very efficient. Also, note that it is essentially optimal as there are $O(p)$ distinct isomorphism classes of maximal order in $\mathcal{B}_{p,\infty}$.

A maximal order \mathcal{O} in $\mathcal{B}_{p,\infty}$ is a dimension-4 lattice contained inside $\mathcal{B}_{p,\infty}$. Thus, it can be given by a basis of 4 elements. Since 1 is always contained in any order \mathcal{O} , we always know one basis element, and so 3 other elements of $\mathcal{B}_{p,\infty}^*$ suffice to define \mathcal{O} . Each element of $\mathcal{B}_{p,\infty}$ can be given by four coefficients in \mathbb{Q} as their coefficients in the basis $1, i, j, k$ of $\mathcal{B}_{p,\infty}$. However, this representation is not very compact. The best bound on the coefficients of the smallest basis of \mathcal{O} allows us to get a representation of size $\approx 3 \log p$ at best. As we explained already this is not optimal, as there are less than p maximal orders inside $\mathcal{B}_{p,\infty}$.

Our idea to obtain an optimal compression is to use an ideal connecting \mathcal{O} (or rather an order isomorphic to \mathcal{O}) and \mathcal{O}_0 , i.e. an ideal whose left order is \mathcal{O}_0 and right order is isomorphic to \mathcal{O} , where \mathcal{O}_0 is the special extremal order in $\mathcal{B}_{p,\infty}$ (for instance when $p \equiv 3 \pmod{4}$, we have $\mathcal{O}_0 = \langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \rangle$).

It can be shown that there always exists such an ideal of norm $N < \sqrt{p}$ (see [34, Section 3.1] for instance). The purpose of the rest of this section is to show that it is possible to encode an \mathcal{O}_0 ideal of norm N in approximately $\log N$ bits.

The main result behind our representation method is the following lemma. We recall that a primitive quaternion element is an element γ contained in an order \mathcal{O} such that there does not exist any integer $k > 1$ such that $\gamma/k \in \mathcal{O}$.

Lemma 5.1. *Let p be a prime, and N be any integer coprime to p . Let \mathcal{O}_0 be a maximal order of $\mathcal{B}_{p,\infty}$ and let I be a left \mathcal{O}_0 -ideal of norm N and let ι, γ be two primitive quaternion elements in \mathcal{O}_0 such that $n(\iota)$ is coprime to N , $\gamma \notin \mathbb{Z}[i]$ and $\gcd(n(\gamma), N^2) = N$. Then, there exists $(a : b) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $I = \mathcal{O}_0 \langle \gamma(a + \iota b), N \rangle$ and $\gcd(a, b, N) = 1$.*

Proof. By the definition of γ , the ideal $\mathcal{O}_0 \langle \gamma, N \rangle$, has norm N . We adapt the proof of [22, Lemma 8] to the case of generic N easily as the norm is not required to be of the form ℓ^f in the proof. Together with existence shown in the proof of [22, Proposition 9], we know there exists a, b such that $\gamma(a + \iota b) \in I$ and $\gcd(a, b) = 1$. This implies $I = \mathcal{O}_0 \langle \gamma(a + \iota b), N \rangle$; otherwise, there exists $n \in \mathbb{N}$ dividing N such that $E[n] \subset \ker(\gamma(a + \iota b))$, which is not possible since $\gcd(a, b, N) = 1$ and ι, γ are primitive. Since multiplying both a and b by any integer coprime to N will not change that fact, we can consider $(a : b)$ as an element in $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. \square

When \mathcal{O}_0 is a special extremal order, it is easy to devise a way to generate a ι and a γ satisfying the requirements of Lemma 5.1 from \mathcal{O}_0 and N . Then, finding suitable a, b can be done by linear algebra modulo N as is done in [22, Algorithm 4].

Thus, when \mathcal{O}_0 is canonical, giving N and an element $(a : b) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ suffices to determine uniquely any ideal of norm N and recover it efficiently.

It only remains to show that we can give the data of an integer N and an element $(a : b) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ with $\gcd(a, b) = 1$ in $\approx 2 \log N$ bits.

Let $a_0 = \gcd(a, N)$ and $b_0 = \gcd(b, N)$ and let $c < N$ be an integer such that $c = (b/b_0)(a/a_0)^{-1} \pmod{N}$. Then, $(a : b) = (a_0 : b_0 c)$, and $N, (a : b)$ can be represented by the four integers $N/(a_0 b_0), a_0, b_0, c$. It is easy to see that this four elements can be represented in $\approx 2 \log N$ bits.

We give a precise description of the algorithm `CompressMaxOrder` in the section as in Algorithm 5. We also require following efficient algorithms as subroutines, which can be found in [21].

- $\text{ConnectingIdeal}(\mathcal{O}_1, \mathcal{O}_2)$: on input two maximal orders $\mathcal{O}_1, \mathcal{O}_2$ in $B_{p,\infty}$ returning a connecting ideal;
- $\text{FullRepresentInteger}_{\mathcal{O}_0}(M)$: on input $M \in \mathbb{Z}$ and $M > p$ returning $\gamma = x + yi + z\frac{i+j}{2} + t\frac{1+k}{2}$ with $n(\gamma) = M$;
- $\text{EquivalentIdeal}_{\mathcal{O}_0}(I)$: on input a left \mathcal{O}_0 -ideal I returning the equivalent left \mathcal{O}_0 -ideal of the smallest norm.

Algorithm 5 $\text{CompressMaxOrder}_{\mathcal{O}_0}$

Require: $\mathcal{O} \subset B_{p,\infty}$ a maximal order.

Ensure: a compressed representation of \mathcal{O}

- 1: $I \leftarrow \text{ConnectingIdeal}(\mathcal{O}_0, \mathcal{O})$.
 - 2: $J \leftarrow \text{EquivalentIdeal}_{\mathcal{O}_0}(I)$.
 - 3: $N \leftarrow n(J)$.
 - 4: $\gamma \leftarrow \text{FullRepresentInteger}_{\mathcal{O}_0}(N(p+1))$ \triangleright Primitive γ
 - 5: Take random elements in \mathcal{O}_0 until finding an element ι of norm coprime to N .
 - 6: Compute values a, b with $\gcd(a, b, N) = 1$ such that $J = \mathcal{O}_0\langle\gamma(a + \iota b), N\rangle$.
 - 7: $a_0 \leftarrow \gcd(a, N)$, $b_0 \leftarrow \gcd(b, N)$, $c \leftarrow (b/b_0)(a/a_0)^{-1} \pmod N$.
 - 8: **return** $N/(a_0b_0), a_0, b_0, c$.
-

The length of the output of CompressMaxOrder depends on N , which can be bounded above by \sqrt{p} as we explained already. Looking ahead, we will use our compression technique to represent the isogeny between the curves corresponding to \mathcal{O}_0 and \mathcal{O}_1 respectively. Since CompressMaxOrder uses the connecting ideal of the smallest norm and does not depend on the representative of the input order, the representation does not leak the information of the exact isogeny used to reach the corresponding curves.

If CompressMaxOrder is running in such a way that the “random” choices made to generate γ and ι are deterministic (meaning that anyone running the computation for the same values of p, N, \mathcal{O}_0 will end up with the same γ and ι), then it is possible to decompress an output of CompressMaxOrder to find the maximal order given in input (or at least a maximal order in the same isomorphism class which is enough for us). This yields the decompression algorithm that we describe as Algorithm 6.

Algorithm 6 $\text{DecompressMaxOrder}$

Require: a compressed representation s of \mathcal{O} .

Ensure: a maximal order $\mathcal{O} \in B_{p,\infty}$

- 1: Parse s as four integers x, a', b', c' .
 - 2: $N \leftarrow xa'b'$
 - 3: $\gamma \leftarrow \text{FullRepresentInteger}_{\mathcal{O}_0}(N(p+1))$ \triangleright Primitive γ
 - 4: Take random elements in \mathcal{O}_0 until finding an element ι of norm coprime to N .
 - 5: Compute $J = \mathcal{O}_0\langle\gamma(a' + \iota b'c'), N\rangle$.
 - 6: **return** $\mathcal{O}_R(J)$.
-

5.2 Maximal Order to Curve

In this section, we give a brief description of an algorithm OrderToCurve to compute an elliptic curve whose endomorphism ring is isomorphic to a maximal order $\mathcal{O} \subset B_{p,\infty}$ given in input. This algorithm is quite easily built from AnyIdealTolsogeny algorithm introduced as [4, Algorithm 3] (see Section 2.3 for more details on this algorithm).

Algorithm 7 OrderToCurve

Require: a maximal order $\mathcal{O} \in \mathcal{B}_{p,\infty}$

Ensure: a curve E with $\text{End}(E) \cong \mathcal{O}$

- 1: Compute $I = I(\mathcal{O}_0, \mathcal{O})$ the ideal connecting \mathcal{O}_0 and \mathcal{O} .
 - 2: $E_0, E, F \leftarrow \text{AnyIdealTolsogeny}(I)$
 - 3: **return** E .
-

5.3 Our Sigma Protocol for Logarithmic Size Ring Signature

We consider the “OR relation” defined in Problem 2.7, where the prover proves the possession of an ideal which is a connecting ideal between E_0 and one of E_1, \dots, E_M . Concretely,

$$R_{E_0} = \{(\{E_i\}_{i \in [M]}, (l, I)) \mid \mathcal{O}_L(I) \cong \text{End}(E_0) \wedge \mathcal{O}_R(I) \cong \text{End}(E_l)\}.$$

Before introducing the base OR sigma protocol for the logarithmic ring signature. We need the following efficient algorithms as the ingredients:

- $\text{KerGen}_D(E, r)$: on input a smooth number smooth D , a curve E and $r \in [D + 1]$ returning a cyclic kernel over E of degree $D \approx p$, which is deterministic conditioned on r . This can be instantiated by generating a canonical basis $\{P, Q\}$ for $E[D]$ and determining the kernel with r .
- $\text{IsoToIdeal}_I(\phi)$: on input a left \mathcal{O}_0 -ideal I and $\phi : E \rightarrow E'$ where $\mathcal{O}_R(I) \cong \text{End}(E')$ returning an \mathcal{O}_0 -ideal J such that $\mathcal{O}_R(J) \cong \text{End}(E')$.
- A pseudorandom number generator. We will instantiate it by using a random oracle $\text{O}(\text{Expand}\|\cdot)$.
- A collision-resistant hash function. We will use this to compress the input for the Merkle tree function. We will instantiate it by using a random oracle $\text{O}(\text{Com}\|\cdot)$.
- $\text{MerkleTree}(\cdot), \text{getMerklePath}(\cdot), \text{ReconstructRoot}(\cdot)$: the collision-resistant Merkle Tree function set that are able to hide the index for any node and path pair given by $\text{getMerklePath}(\cdot)$ is given. We use the instantiation in [6] together with its properties, which is can be instantiated by using a collision-resistant hash function from $\{0, 1\}^\lambda$ to $\{0, 1\}^{2\lambda}$. We give a brief overview in Appendix B.

We now sketch the base OR sigma protocol as shown in Figure 2.

Theorem 5.2. *The sigma protocol Π_Σ described in Figure 2 has correctness and λ min-entropy.*

Proof. When the challenge is $\text{ch} = 0$, the prover sends the seed to the verifier. The computation of the verifier will result in the same commitment (root) in this case.

When $\text{ch} = 1$, the prover sends $(s_{\mathcal{O}}, \text{path}, \text{bits}_l)$ to the verifier, where $s_{\mathcal{O}} \leftarrow \text{CompressMaxOrder}(\mathcal{O}_R(J))$ and $\mathcal{O}_R(J) \cong \text{End}(E'_l)$. For \mathcal{O}' , the output of $\text{DecompressMaxOrder}(s_{\mathcal{O}})$, we have $\mathcal{O}' \cong \text{End}(E'_l)$ by the Deuring’s correspondence, since $\text{DecompressMaxOrder}$ obtains \mathcal{O}' by reconstructing a connecting ideal equivalent to J . Hence, $\text{OrderToCurve}(\mathcal{O}')$ gives a curve isomorphic to E'_l and results in the same j -invariant. Hence, ReconstructRoot will produce the same commitment. Besides, since we model $\text{O}(\text{Com}\|\cdot)$ as a random, the scheme has λ entropy. \square

Theorem 5.3. *The Π_Σ depicted in Figure 2 has Special Soundness if $\text{O}(\text{Com}\|\cdot)$ and $\text{MerkleTree}(\cdot)$ is collision resistant.*

Proof. Given two valid transcripts $(\text{com}, 0, \text{rsp}_0)$ and $(\text{com}, 1, \text{rsp}_1)$ under the same statement $(\{E_i\}_{i \in [M]})$, the extractor Extract proceeds as follows.

1. Generate by using the transcript $(\text{com}, 0, \text{rsp}_0)$ and following the prover’s first round procedure, obtain $\phi_i, \mathbf{C}_i \leftarrow \text{O}(\text{Com}\|_j(E'_i)\|\text{bits}_i)$ for each $i \in [N]$ and the root $(\text{com}, \text{tree}) \leftarrow \text{MerkleTree}(\mathbf{C}_1, \dots, \mathbf{C}_N)$.
2. Generate by using the transcript $(\text{com}, 1, \text{rsp}_1)$ and following the verification procedure, obtain $\mathcal{O}' \leftarrow \text{DecompressMaxOrder}(s), E' \leftarrow \text{OrderToCurve}(\mathcal{O}')$ and $\tilde{\mathbf{C}} \leftarrow \text{O}(\text{Com}\|_j(E')\|\text{bits})$.
Also, extract the ideal \tilde{J} during the execution of $\text{DecompressMaxOrder}$.

<p>round 1: $P_1^O(\{E_i\}_{i \in [M]}, (l, I))$</p> <ol style="list-style-type: none"> 1: seed $\xleftarrow{\\$} \{0, 1\}^\lambda$ 2: $(r, \text{bits}_1, \dots, \text{bits}_M) \leftarrow O(\text{Expand} \parallel \text{seed})$ 3: for i from 1 to M do 4: $E'_i, \phi'_i \leftarrow \text{KerGen}_D(E_i, r)$ 5: $\mathbf{C}_i \leftarrow O(\text{Com} \parallel j(E'_i) \parallel \text{bits}_i)$ 6: $(\text{root}, \text{tree}) \leftarrow \text{MerkleTree}(\mathbf{C}_1, \dots, \mathbf{C}_M)$ 7: Prover sends com \leftarrow root to Verifier. <p>round 2: $V_1^O(\text{com})$</p> <ol style="list-style-type: none"> 1: ch $\xleftarrow{\\$} \{0, 1\}$ 2: Verifier sends ch to Prover. <p>round 3: $P_2^O((l, I), \text{ch})$</p> <ol style="list-style-type: none"> 1: if ch = 1 then 2: $J \leftarrow \text{IsoToldeal}_I(\phi'_i)$ 3: path $\leftarrow \text{getMerklePath}(\text{tree}, l)$ 4: $s_{\mathcal{O}} \leftarrow \text{CompressMaxOrder}(\mathcal{O}_R(J))$ 5: rsp $\leftarrow (s_{\mathcal{O}}, \text{path}, \text{bits}_i)$ 6: else 7: rsp \leftarrow seed 8: Prover sends rsp to Verifier 	<p style="text-align: right;">\triangleright Sample $r' \in [D + 1]$ and $\text{bits}_i \in \{0, 1\}^\lambda$</p> <p style="text-align: right;">\triangleright Create commitments $\mathbf{C}_i \in \{0, 1\}^{2\lambda}$</p> <p>Verification: $V_2^O(\text{com}, \text{ch}, \text{rsp})$</p> <ol style="list-style-type: none"> 1: $(\text{root}, \text{ch}) \leftarrow (\text{com}, \text{ch})$ 2: if ch = 1 then 3: $(s, \text{path}, \text{bits}) \leftarrow \text{rsp}$ 4: $\mathcal{O}' \leftarrow \text{DecompressMaxOrder}(s)$ 5: $E' \leftarrow \text{OrderToCurve}(\mathcal{O}')$ 6: $\tilde{\mathbf{C}} \leftarrow O(\text{Com} \parallel j(E') \parallel \text{bits})$ 7: $\tilde{\text{root}} \leftarrow \text{ReconstructRoot}(\tilde{\mathbf{C}}, \text{path})$ 8: Verifier accepts if $\tilde{\text{root}} = \text{root}$ 9: else 10: Repeat round 1 with seed \leftarrow rsp 11: Output accept if the computation results in root, otherwise reject
--	---

Fig. 2: Construction of the base OR sigma protocol $\Pi_\Sigma = (P' = (P'_1, P'_2), V' = (V'_1, V'_2))$ for the relation R_{sig} . Informally, $O(\text{Expand} \parallel \cdot)$ and $O(\text{Com} \parallel \cdot)$ are a PRG and a commitment scheme instantiated by the random oracle, respectively.

3. Find $\tilde{l} \in [N]$ such that $\mathbf{C}_{\tilde{l}} = \tilde{\mathbf{C}}$ and assert $j(E') = j(E'_{\tilde{l}})$.
4. Return \tilde{I} where the ideal $\tilde{I} \leftarrow \text{IsoToldeal}_{\tilde{j}}(\hat{\phi}_{\tilde{l}})$.

In Item 3, if such an index \tilde{I} does not exist, then a collision occurs in the Merkle tree as shown in [6, Lemma 2.9]. Similarly, if the first condition is satisfied and $j(E') \neq j(E'_{\tilde{l}})$, a collision is detected for $O(\text{Com} \parallel \cdot)$. It suffices to show that \tilde{I} is a connecting ideal between $\text{End}(E_0)$ and $\text{End}(E_I)$. Given that \tilde{J} is a connecting ideal between $\text{End}(E_0)$ and $\text{End}(E'_I)$ and $\hat{\phi}_I : E'_I \rightarrow E_I$, the mapping $\text{IsoToldeal}_{\tilde{j}}(\hat{\phi}_I)$ provides the connecting ideal between $\text{End}(E_0)$ and $\text{End}(E_I)$. Therefore, the scheme demonstrates special soundness. \square

Theorem 5.4. *The scheme Π_Σ depicted in Figure 2 is statistically special honest-verifier zero-knowledge. Concretely, there exists a simulator \mathcal{S} such that for any statement and witness in the relation and computationally-unbounded adversary \mathcal{A} with at most Q queries to the random oracle, we have*

$$\text{Adv}_{\Pi_\Sigma}^{\text{HVZK}}(\mathcal{A}) \leq Q/2^\lambda + \text{negl}(\lambda)$$

for some negligible function $\text{negl}(\lambda)$.

Proof. Let $\mathbf{x} = \{E_i\}_{i \in [M]}$ be a statement and fixed. The simulator \mathcal{S} with access to the random oracle \mathcal{O} runs on input \mathbf{x} and $\text{ch} \in \{0, 1\}$ as follows.

1. When $\text{ch} = 0$, \mathcal{S} follows the same procedure as a real prover and outputs a transcript $(\text{com}, 0, \text{rsp})$ where a witness is not required in this case.
2. When $\text{ch} = 1$, the simulator generates a cyclic isogeny from E_0 of degree D uniformly at random and computes the codomain curve E' , generates bits $\xleftarrow{\$} \{0, 1\}^\lambda$, and computes $\mathbf{C}_1 = O(\text{Com} \parallel j(E') \parallel \text{bits})$. \mathcal{S} computes the connecting ideal J by using IsoToldeal such that $\mathcal{O}_R(J) \cong \text{End}(E')$, then computes $s_{\mathcal{O}}$ by using CompressMaxOrder . Next, \mathcal{S} generates dummy commitments $\mathbf{C}_2, \dots, \mathbf{C}_M \xleftarrow{\$} \{0, 1\}^{2\lambda}$ and computes $(\text{root}, \text{tree}) \leftarrow \text{MerkleTree}(\mathbf{C}_1, \dots, \mathbf{C}_M)$. Then, \mathcal{S} computes $\text{path} \leftarrow (\text{tree}, 1)$. Set $\text{com} = \text{root}$ and $\text{rsp} = (s_{\mathcal{O}}, \text{path}, \text{bits})$. \mathcal{S} returns $(\text{com}, 1, \text{rsp})$.

Clearly, for the case $\text{ch} = 0$, the simulation is perfect. To show the transcripts are indistinguishable, we use a hybrid argument by introducing a series of simulators $\mathcal{S}_0 = P, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 = \mathcal{S}$. In each simulation, we gradually modify the transcripts from \mathcal{S}_0 , identical to the real prover P using a witness, to \mathcal{S}_3 , identical to \mathcal{S} witness. For $i \in \{1, 2, 3\}$, we define the advantage of \mathcal{S}_i to be

$$\text{Adv}_i(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}}(\mathcal{S}_{i-1}^{\mathcal{O}}(x, w, 1)) = 1] - \Pr[\mathcal{A}^{\mathcal{O}}(\mathcal{S}_i^{\mathcal{O}}(x, w, 1)) = 1]| .$$

\mathcal{S}_1 proceeds the same as a real prover P except that it is not using $\mathcal{O}(\text{Expand}\|\cdot)$ and $\mathcal{O}(\text{Com}\|\cdot)$ to generate $(r, \text{bits}_1, \dots, \text{bits}_M)$ and $\{\mathbf{C}_i\}_{i \in [M] - \{I\}}$. Instead, it samples the elements uniformly at random from the corresponding domain. Since $\mathcal{O}(\text{Expand}\|\cdot)$ and $\mathcal{O}(\text{Com}\|\cdot)$ are modeled by a random oracle, the elements follows the same distribution except for those that have been queried. Since seed and bits_i for each i have λ bits of min-entropy, we have $\text{Adv}_1(\mathcal{A}) \leq Q/2^\lambda$.

\mathcal{S}_2 proceeds the same as \mathcal{S}_1 except that it is not using generating E'_l from E_l and is not using the secret ideal I between E_0 and E_l to generate the ideal J . Instead, it generates a cyclic isogeny from E_0 of degree D uniformly at random and computes the codomain curve E'_l . Also, it is able to compute J by using IsoToIdeal without using I . Due to the choice of D , by Theorem 2.5, the statistical distances of the output distributions is bounded by $\text{negl}(\lambda)$ for some negligible function $\text{negl}(\lambda)$. We have $\text{Adv}_1(\mathcal{A}) \leq \text{Adv}_2(\mathcal{A}) + \text{negl}(\lambda)$.

$\mathcal{S}_3 = \mathcal{S}$ is using 1 as the index to generate the root instead of $l \in [M]$. By Lemma B.2, the output follows the same distribution. Hence, we have $\text{Adv}_2(\mathcal{A}) = \text{Adv}_3(\mathcal{A})$. By a union bound, we have $\text{Adv}_{\Pi_{\Sigma}^{\text{HVZK}}}(\mathcal{A}) \leq Q/2^\lambda + \text{negl}(\lambda)$. \square

6 Instantiations

We estimate the parameters and the performance for **Erebor** and **Durian**, summarized in Tables 3 and 4 with a comparison with the state-of-art ring signature schemes in the post-quantum literature and the isogeny literature.

Linear-size Ring Signature. For **Erebor** the final signature parameters and the timings depends on the particular prime p and the subroutines used during the computations. We try to provide some size estimates below as functions of p, e , where $2^e = D_{\text{rsp}}$, the security parameter λ and the number of users in the ring N . We write $\text{cost}(x)$ for the size of a compressed representation of the data x . Most of the compression techniques mentioned below are rather standard for **SQIsign**.

Since the starting curve of ϕ_{ch} is the public key we can represent it with the coordinates of a generator of the kernel subgroup, so $\text{cost}(\phi_{\text{ch}}) = \lambda \approx \log_2(D_c)$. Moreover, $\hat{\sigma} : E_{\text{ch}} \rightarrow E_{\text{cmt}}$ can be compressed too as described in [21, Section 8.5] $e + 4(\lceil e/f \rceil - 1)$, where f is the exponent of the maximum available power-of-2 torsion (i.e. the maximum such that $2^f \mid (p^2 - 1)$), also, all the other compression techniques from [55] can be used in this context, obtaining different trade off of efficiency vs. compression. For level of security NIST I we consider the prime p_{3923} from [22], we get $\text{cost}(\hat{\sigma}) = 170\text{B}$ for the *full* version and 130B for the *short* one.

With respect to timings, we need to perform $N - 1$ simulations, that means computing a degree $D_c \cdot D_{\text{rsp}}$ isogeny (we need it to compute $\hat{\sigma} \circ \phi_{\text{ch}}$ to get the commitment), and one generation of a response, that has the same cost of performing a **SQIsign** KLPT based signature, possibly adjusted to the increased value for D_{rsp} . It is important to observe that if we need to generate ψ of small degree, as in the **KeyGen** procedure of **SQIsign**, the cost of the commitment generation greatly increases. The verification cost correspond, as for the simulation, to the computation of N degree $D_c \cdot D_{\text{rsp}}$ isogenies.

For NIST I, we provide in Table 2 an estimate of the costs as a function of the number of users N in millions of cycles. As a baseline, we take the numbers provided in [22] for the optimized **C** implementation with the prime p_{3923} scaled linearly with the rate of the length of the response in the variant of **Erebor** we consider over the length of the response in **SQIsign**.

Note that these estimates give a lower-bound on the actual efficiency. Leroux in [40], and Onuki and Nakagawa in [49] described improved variant of **SQIsign**'s ideal-to-isogeny translation method using high dimensional isogenies that should outperform the approach in [22]. Since no competitive implementation was provided yet for these new algorithms, we rather use the results from [22].

NIST 1	Signing (MC)	Ver. (MC)	Signature size (B)
Erebor- <i>full</i>	$(N - 1) \cdot 41 + 2683$	$N \cdot 41$	$16 + N \cdot 170$
Erebor- <i>short</i>	$(N - 1) \cdot 30 + 2408$	$N \cdot 30$	$16 + N \cdot 130$

Table 2: Size in bytes (B) and efficiency estimates in Millions of CPU cycles (MC) for Erebor.

Logarithmic-size Ring Signature. For Durian in Section 5, we can take the underlying prime the same as [4] $p = 5 \cdot 2^{248} - 1$ where we can find a smooth torsion subgroup easily and the execution of KerGen is fast when the degree D is a power of 2 smaller than 2^{248} . We remark that the signature size of Durian is solely based on the parameter p and the security parameter λ regardless of other parameters like D . For the ring size N , when the challenge is 0, the response has λ bits. When the challenge is 1, the response is approximately $\log_2(p) + \lceil \log_2(N) \rceil \cdot 2\lambda + \lambda$ where $\log_2(p)$ upper-bounds the output of CompressMaxOrder. Hence, the signature is expected to take $(\log_2(p) + \lceil \log_2(N) \rceil \cdot 2\lambda + \lambda) \cdot \lambda/2$ bits. However, if we use the standard Fiat-Shamir with unbalanced challenge space technique to mitigate the overhead incurred by the increasing ring size, as in [6] (see [5,37] for obtaining a tight reduction without rewinding) and a seed tree to compress the seeds for the zero challenges [6] (see [8, Appendix B] for a precise upperbound to the size). Here, the challenge space consists of n -bits strings of Hamming weight k . We consider $N = 2$, $N = 8$ and $N = 2^{10}$. By taking $\lambda = 128$ and $\log_2(p) = 251$ targeting NIST 1, if we choose mild parameters $(n, k) = (193, 35)$, $(193, 35)$ and $(455, 23)$ respectively such that $\binom{n}{k} > 2^\lambda$ to obtain more compact signatures of 4.08KB, 6.29KB and 9.87KB respectively. The choice will not slow down the overall performance too much.

We did not implement Durian but we can estimate the running time based on extrapolation from the recent C implementation from [4] (exact cycle counts for the underlying operations were not presented so we don't have a more precise estimate). The operations: KerGen, IsoToldeal and CompressMaxOrder operations do not take more than 10ms each on average to generate a response. During verification, each execution of the DecompressMaxOrder and OrderToCurve operations takes less than 40 ms on average.

Therefore, for $N = 2$, we estimate that Durian would take approximately 4.2 seconds for signing. Unfortunately, due to the slow subroutines DecompressMaxOrder and OrderToCurve, verification would require about 4.5 seconds. When $N = 8$, we estimate the signing time to be 15 seconds and the verification time to be 14 seconds. When $N = 1024$, it will take tens of minutes to 1 hour to sign and verify. Despite these times, we expect Durian to be faster than its isogeny group action counterpart, Calamari, where each group action takes 40 ms.

	pk size	sk size	N			Hardness Assumption	Security Level
			2	4	8		
Erebor- <i>full</i> [4]	0.06	0.03	0.35	0.68	1.35	Problems 4.4 and 2.8	NIST 1
Gandalf [28]	0.89	*	1.2	2.4	4.8	R-NTRU, R-SIS	NIST 1
DualRing [61]	2.84	0.23	4.56	4.64	4.74	M-LWE, M-SIS	NIST 1

Table 3: A comparison between full anonymous ring signatures in the literature where the signature size grows linearly in the ring size. The size unit is in KB. N represents the ring size. Problem 2.8, the supersingular endomorphism problem, is equivalent to the isogeny problem [25,50].

Acknowledgement

We thank Luca De Feo for his advice all along the project. Yi-Fu Lai is supported by the European Union (ERC AdG REWORC - 101054911). Giacomo Borin is supported by SNSF Consolidator Grant CryptonIs 213766.

	pk size	sk size	N			Hardness Assumption	Security Level
			2	2^3	2^{10}		
Durian (Figure 2)	0.06	0.8	4.08	6.29	9.87	Problem 2.7	NIST 1
SMILE ⁺ [43]	2.00	1.73	/	/	17.27	M-SIS, M-LWE	NIST 1
Calamari [6]	0.06	0.03	3.5	5.4	9.6	GAIP	≥ 60

Table 4: A comparison between full anonymous logarithmic-size ring signatures in the literature. The size unit is in KB. N represents the ring size. SMILE provides logarithmic ring signature with size of form $N = 32^i$ and has 15.96KB for $N = 32$. Problem 2.7, the supersingular endomorphism ring problem, is tightly equivalent to the isogeny problem [25].

References

1. Abdalla, M., An, J.H., Bellare, M., Nampreppe, C.: From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 418–433. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7_28
2. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer, Heidelberg (Dec 2002). https://doi.org/10.1007/3-540-36178-2_26
3. Basso, A., Codogni, G., Connolly, D., De Feo, L., Fouotsa, T.B., Lido, G.M., Morrison, T., Panny, L., Patranabis, S., Wesolowski, B.: Supersingular curves you can trust. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part II. LNCS, vol. 14005, pp. 405–437. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30617-4_14
4. Basso, A., Feo, L.D., Dartois, P., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-west: The fast, the small, and the safer. Cryptology ePrint Archive, Paper 2024/760 (2024), <https://eprint.iacr.org/2024/760>, <https://eprint.iacr.org/2024/760>
5. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 95–126. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_4
6. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falaf: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 464–492. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_16
7. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_26
8. Borin, G., Persichetti, E., Santini, P., Pintore, F., Reijnders, K.: A guide to the design of digital signatures based on cryptographic group actions. Cryptology ePrint Archive, Paper 2023/718 (2023), <https://eprint.iacr.org/2023/718>, <https://eprint.iacr.org/2023/718>
9. Brendel, J., Fiedler, R., Günther, F., Janson, C., Stebila, D.: Post-quantum asynchronous deniable key exchange and the Signal handshake. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part II. LNCS, vol. 13178, pp. 3–34. Springer, Heidelberg (Mar 2022). https://doi.org/10.1007/978-3-030-97131-1_1
10. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_15
11. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 395–427. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03332-3_15
12. Chavez-Saab, J., Corte-Real Santos, M., De Feo, L., Komada Eriksen, J., Hess, B., Kohel, D., Leroux, A., Longa, P., Meyer, M., Panny, L., Patranabis, S., Petit, C., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: Squisign specification. <https://squisign.org/spec/squisign-20230601.pdf> (2023), accessed: 2023-10-04
13. Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: Group action from 2-dimensional isogenies. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part II. LNCS, vol. 14603, pp. 190–216. Springer, Heidelberg (Apr 2024). https://doi.org/10.1007/978-3-031-57725-3_7
14. Cong, K., Lai, Y.F., Levin, S.: Efficient isogeny proofs using generic techniques. In: Tibouchi, M., Wang, X. (eds.) ACNS 23, Part II. LNCS, vol. 13906, pp. 248–275. Springer, Heidelberg (Jun 2023). https://doi.org/10.1007/978-3-031-33491-7_10

15. van der Corput, J.: Verallgemeinerung einer mordellschen beweisermethode in der geometrie der zahlen. *Acta Arithmetica* **1**(1), 62–66 (1935)
16. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) *CRYPTO'94*. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_19
17. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQISignHD: New dimensions in cryptography. In: Joye, M., Leander, G. (eds.) *EUROCRYPT 2024*, Part I. LNCS, vol. 14651, pp. 3–32. Springer, Heidelberg (May 2024). https://doi.org/10.1007/978-3-031-58716-0_1
18. De Feo, L., Dobson, S., Galbraith, S.D., Zobernig, L.: SIDH proof of knowledge. In: Agrawal, S., Lin, D. (eds.) *ASIACRYPT 2022*, Part II. LNCS, vol. 13792, pp. 310–339. Springer, Heidelberg (Dec 2022). https://doi.org/10.1007/978-3-031-22966-4_11
19. De Feo, L., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: SCALLOP: Scaling the CSI-FiSh. In: Boldyreva, A., Kolesnikov, V. (eds.) *PKC 2023*, Part I. LNCS, vol. 13940, pp. 345–375. Springer, Heidelberg (May 2023). https://doi.org/10.1007/978-3-031-31368-4_3
20. De Feo, L., Jao, D., Plüt, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology* **8**(3), 209–247 (2014)
21. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) *ASIACRYPT 2020*, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_3
22. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In: Hazay, C., Stam, M. (eds.) *EUROCRYPT 2023*, Part V. LNCS, vol. 14008, pp. 659–690. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_3
23. De Feo, L., Meyer, M.: Threshold schemes from isogeny assumptions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) *PKC 2020*, Part II. LNCS, vol. 12111, pp. 187–212. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45388-6_7
24. Duparc, M., Fouotsa, T.B.: Sqprime: A dimension 2 variant of sqsignhd with non-smooth challenge isogenies. *Cryptology ePrint Archive* (2024)
25. Eisenträger, K., Hallgren, S., Lauter, K.E., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) *EUROCRYPT 2018*, Part III. LNCS, vol. 10822, pp. 329–368. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_11
26. Esgin, M.F., Steinfeld, R., Zhao, R.K.: MatRiCT⁺: More efficient post-quantum private blockchain payments. In: 2022 IEEE Symposium on Security and Privacy. pp. 1281–1298. IEEE Computer Society Press (May 2022). <https://doi.org/10.1109/SP46214.2022.9833655>
27. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) *Advances in Cryptology — CRYPTO' 86*. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987)
28. Gajland, P., Janneck, J., Kiltz, E.: Ring signatures for deniable AKEM: Gandalf's fellowship. *Cryptology ePrint Archive*, Paper 2024/890 (2024), <https://eprint.iacr.org/2024/890>, <https://eprint.iacr.org/2024/890>
29. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. In: Takagi, T., Peyrin, T. (eds.) *ASIACRYPT 2017*, Part I. LNCS, vol. 10624, pp. 3–33. Springer, Heidelberg (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_1
30. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Oswald, E., Fischlin, M. (eds.) *EUROCRYPT 2015*, Part II. LNCS, vol. 9057, pp. 253–280. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_9
31. Hashimoto, K., Katsumata, S., Kwiatkowski, K., Prest, T.: An efficient and generic construction for Signal's handshake (X3DH): Post-quantum, state leakage secure, and deniable. In: Garay, J. (ed.) *PKC 2021*, Part II. LNCS, vol. 12711, pp. 410–440. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75248-4_15
32. Katsumata, S., Lai, Y.F., LeGrow, J.T., Qin, L.: CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In: Handschuh, H., Lysyanskaya, A. (eds.) *CRYPTO 2023*, Part III. LNCS, vol. 14083, pp. 729–761. Springer, Heidelberg (Aug 2023). https://doi.org/10.1007/978-3-031-38548-3_4
33. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) *ACM CCS 2018*. pp. 525–537. ACM Press (Oct 2018). <https://doi.org/10.1145/3243734.3243805>
34. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. *LMS Journal of Computation and Mathematics* **17**(A), 418–432 (2014)
35. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing* **35**(1), 170–188 (2005)

36. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In: Severini, S., Brandão, F.G.S.L. (eds.) TQC 2013. LIPIcs, vol. 22, pp. 20–34. Schloss Dagstuhl (2013)
37. Lai, Y.F.: CAPYBARA and TSUBAKI: Verifiable random functions from group actions and isogenies. Cryptology ePrint Archive, Report 2023/182 (2023), <https://eprint.iacr.org/2023/182>
38. Lai, Y.F., Galbraith, S.D., Delpech de Saint Guilhem, C.: Compact, efficient and UC-secure isogeny-based oblivious transfer. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 213–241. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-77870-5_8
39. Leroux, A.: Quaternion Algebra and isogeny-based cryptography. Ph.D. thesis, PhD thesis, Ecole doctorale de l’Institut Polytechnique de Paris (2022)
40. Leroux, A.: Verifiable random function from the deuring correspondence and higher dimensional isogenies. Cryptology ePrint Archive, Paper 2023/1251 (2023), <https://eprint.iacr.org/2023/1251>, <https://eprint.iacr.org/2023/1251>
41. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 04. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (Jul 2004). https://doi.org/10.1007/978-3-540-27800-9_28
42. Lu, X., Au, M.H., Zhang, Z.: Raptor: A practical lattice-based (linkable) ring signature. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 19. LNCS, vol. 11464, pp. 110–130. Springer, Heidelberg (Jun 2019). https://doi.org/10.1007/978-3-030-21568-2_6
43. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: SMILE: Set membership from ideal lattices with applications to ring signatures and confidential transactions. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 611–640. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84245-1_21
44. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 448–471. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_16
45. Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO’87. LNCS, vol. 293, pp. 369–378. Springer, Heidelberg (Aug 1988). https://doi.org/10.1007/3-540-48184-2_32
46. Nakagawa, K., Onuki, H.: SQIsign2D-east: A new signature scheme using 2-dimensional isogenies. Cryptology ePrint Archive, Paper 2024/771 (2024), <https://eprint.iacr.org/2024/771>, <https://eprint.iacr.org/2024/771>
47. Nguyen, P.Q., Stehlé, D.: Low-dimensional lattice basis reduction revisited. ACM Transactions on algorithms (TALG) **5**(4), 1–48 (2009)
48. Noether, S.: Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098 (2015), <https://eprint.iacr.org/2015/1098>
49. Onuki, H., Nakagawa, K.: Ideal-to-isogeny algorithm using 2-dimensional isogenies and its application to sqisign. Cryptology ePrint Archive (2024)
50. Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VI. LNCS, vol. 14656, pp. 388–417. Springer, Heidelberg (May 2024). https://doi.org/10.1007/978-3-031-58751-1_14
51. Peikert, C.: He gives C-sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 463–492. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45724-2_16
52. Renan, F., Kutas, P.: SQIAsignHD: SQIsignHD adaptor signature. Cryptology ePrint Archive, Paper 2024/561 (2024), <https://eprint.iacr.org/2024/561>, <https://eprint.iacr.org/2024/561>
53. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_32
54. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_17
55. Santos, M.C.R., Eriksen, J.K., Meyer, M., Reijnders, K.: AprèsSQI: Extra fast verification for SQIsign using extension-field signing. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 63–93. Springer, Heidelberg (May 2024). https://doi.org/10.1007/978-3-031-58716-0_3
56. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994). <https://doi.org/10.1109/SFCS.1994.365700>
57. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
58. Urbanik, D., Jao, D.: New techniques for SIDH-based NIKE. Journal of Mathematical Cryptology **14**(1), 120–128 (2020)
59. Voight, J.: Quaternion algebras. Springer Nature (2021)

60. Yoo, Y., Azarderakhsh, R., Jalali, A., Jao, D., Soukharev, V.: A post-quantum digital signature scheme based on supersingular isogenies. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 163–181. Springer, Heidelberg (Apr 2017). https://doi.org/10.1007/978-3-319-70972-7_9
61. Yuen, T.H., Esgin, M.F., Liu, J.K., Au, M.H., Ding, Z.: DualRing: Generic construction of ring signatures with efficient instantiations. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 251–281. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84242-0_10
62. Yuen, T.H., Sun, S., Liu, J.K., Au, M.H., Esgin, M.F., Zhang, Q., Gu, D.: RingCT 3.0 for blockchain confidential transaction: Shorter size and stronger security. In: Bonneau, J., Heninger, N. (eds.) FC 2020. LNCS, vol. 12059, pp. 464–483. Springer, Heidelberg (Feb 2020). https://doi.org/10.1007/978-3-030-51280-4_25

A Proofs for Algorithm 1 Security

A sigma protocol can be rendered to a digital signature via the well-known Fiat-Shamir transform [27] and substituting the random oracle \mathcal{O} with a cryptographic hash function H . We consider another useful security definition for identification protocols, that, as proven in [1], is equivalent to the security of the signature obtained via the Fiat-Shamir transform and it is implied by the previously given definitions (see e.g. [12, Section 9.1]).

Definition A.1 ([1]). *A sigma-protocol Π_Σ is secure against impersonation under passive attacks if, for any polynomial time adversary \mathcal{A} , the probability of winning the following impersonator game is negligible in λ :*

- (i) *The challenger sample a random pair $(x, w) \leftarrow \text{Gen}(1^\lambda)$ and send x to \mathcal{A} ;*
- (ii) *\mathcal{A} can query a polynomial number of valid transcripts;*
- (iii) *\mathcal{A} send a valid commitment com^* to the challenger;*
- (iv) *the challenger send a uniformly random $\text{ch}^* \in \text{ChSet}$;*
- (v) *\mathcal{A} output a response rsp^* , \mathcal{A} wins if $\mathbb{V}_2(x, \text{com}, \text{ch}, \text{rsp}) = 1$ (accepts).*

For convenience we restate the Proposition 2.3 and 2.4, then provide the proofs using basic game based arguments.

Proposition A.2. *If Π^{ld} satisfies the special weak (strong) computational HVZK property the AOS ring signature (Algorithm 1) is anonymous (against key exposure) in the programmable random oracle model.*

Proof (Proof of Proposition 2.3). Consider and adversary \mathcal{A} against the anonymity property playing the game G . We start a modified version of the game G^* in which using the random oracle we generate the signature σ^* without using sk_{i_b} . This can be done by sampling ch_{i_b} at random, obtaining $\text{com}_{i_b}, \text{rsp}_{i_b} \leftarrow \mathcal{S}(\text{ch}_{i_b})$ and then reprogramming the random oracle so that $H(\text{com}_{i_b-1}, \mathbb{R}, m, \text{pk}_{i_b}) \rightarrow \text{ch}_{i_b}$. If a record for that input already exists we can just restart the signature simulation process.

Since G^* outputs are independent of b necessarily $\Pr[\mathcal{A} \text{ wins } G^*] = \frac{1}{2}$, so

$$\text{Adv}_{\Pi_{\text{RS}}}^{\text{Anon}}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins } G] - \Pr[\mathcal{A} \text{ wins } G^*]|,$$

but the only difference between the two games is the use of the simulator \mathcal{S} , so $|\Pr[\mathcal{A} \text{ wins } G] - \Pr[\mathcal{A} \text{ wins } G^*]|$ is bounded by the advantage against the weak HVZK. It is immediate to notice that if we also feed the secret keys (i.e. the witnesses) to \mathcal{A} we are in the same case of the strong HVZK. \square

Proposition A.3. *If Π^{ld} satisfies Definition A.1 the AOS ring signature (Algorithm 1) is unforgeable (UF-CMA) in the programmable random oracle model.*

Proof (Proof of Proposition 2.4).

Consider and adversary \mathcal{A} against the UF-CMA property playing the game G . In the reprogramming random oracle model we show how to render it to an adversary against the impersonation game G_{imp} ([1, Definition 2.1]) for the sigma protocol involved in the ring signature. Let ϵ be the probability of winning this game and $q_{\text{H}}, q_{\text{sig}}$ the number of respectively random oracle and signing queries.

We start the impersonator game and we receive from the challenger the public key (i.e. statement) pk_{imp} , we then query the impersonator challenger for q_{sig} valid transcripts. Then we simulate the UF-CMA game in this way:

- (i) We generate key pairs $(\text{pk}_i, \text{sk}_i) = \text{RS.KeyGen}(\text{pp}; \text{rr}_i)$ for all $i \in [N]$ but for one random index i_{imp} , and we fix $\text{pk}_{i_{\text{imp}}} \leftarrow \text{pk}_{\text{imp}}$. We set $\text{PK} := \{\text{pk}_i\}_{i \in [N]}$ and initializes two empty sets \mathcal{S} and \mathcal{C} .
- (ii) The challenger provides PK to \mathcal{A} ;
- (iii) \mathcal{A} can make signing and corruption queries an arbitrary polynomial number of times:
 - $(\text{sign}, \text{pk}_i, \text{m}, \text{R})$: if $i \neq i_{\text{imp}}$ and $\text{pk}_i \in \text{R}$ we sign the message and return the signature σ to \mathcal{A} . If $i = i_{\text{imp}}$ we use one of the valid transcripts $(\text{com}_q, \text{ch}_q, \text{rsp}_q)$ previously queried to simulate the signature via reprogramming the random oracle. We surely have enough of them since the number of signing queries is bounded. We start the signature procedure committing to com_q , then we reprogram the random oracle so that the last query output ch_q , and return it to \mathcal{A} . We then always add $(i, \text{m}, \text{R}, \sigma)$ to \mathcal{S} .
 - $(\text{corrupt}, i)$: If $i = i_{\text{imp}}$ we declare failure and exit, otherwise we add pk_i to \mathcal{C} and returns rr_i to \mathcal{A} .

The simulation of H as a random oracle is our main tool for the reduction. We can assume without loss of generality that the queries are always of the form $\text{com}_*, \text{R}_*, \text{M}_*, \text{pk}_*$. We also keep a *time-ordered registry* \mathcal{R} of any query $(\text{com}_*, \text{R}_*, \text{M}_*, \text{pk}_*, \text{ch}_*)$, with ch_* being the output. Also at the start we select at random two of the q_{H} queries q_1, q_2 to be reprogrammed.

At the q_1 -th query we take the commitment com_{q_1} . If it is valid for pk_{imp} we send it to the challenger and receive the challenge ch_{imp} . We then reprogram the q_2 -th query to output ch_{imp} if pk_{imp} is queried.

If the adversary \mathcal{A} outputs a valid forgery $\sigma = (\text{ch}_1^*, \text{rsp}_1^*, \dots, \text{rsp}_r^*, \text{com}_1^*, \dots, \text{com}_r^*)$ for M^*, R^* we then look in the register at the oracle queries. Also we index the keys with respect to the order in R .

Because of the ring structure of the oracle calls there must exist at least one *reverse index* i_R such that $(\text{com}_{i_R}^*, \text{R}^*, \text{M}^*, \text{pk}_{i_R+1})$ has been queried before $(\text{com}_{i_R-1}^*, \text{R}^*, \text{M}^*, \text{pk}_{i_R})$. Since the public key are all generated by Gen with probability at least $1/n$ the reverse index is the one associated to pk_{imp} ⁸. Also, with probability $\binom{q_{\text{H}}}{2}^{-1}$ these two queries are exactly q_1, q_2 , so $\text{com}_{i_R}^*$ is the one sent to the challenger and $\text{ch}_{i_R} = \text{ch}_{\text{imp}}$ is the received challenge. The validity of the final signature implies that $(\text{pk}_{\text{imp}}, \text{com}_{i_R}^*, \text{ch}_{\text{imp}}, \text{rsp}_{i_R}^*)$ is a valid transcript, that we can send to the challenger, so

$$\text{Adv}_{\text{RS}}^{\text{Unf}}(\mathcal{A}) \leq n \cdot \binom{q_{\text{H}}}{2} \cdot \epsilon. \quad (9)$$

□

B Index-hiding Merkle trees

The definition an *index-hiding* Merkle tree is taken almost verbatim from [6]. Merkle trees [45] allow one to hash a list of elements $A = (a_0, \dots, a_N)$ into one hash value (often called the *root*). At a later point, one can efficiently prove to a third party that an element a_i was included at a certain position in the list A . In the following, we consider a slight modification of the standard Merkle tree construction, such that one can prove that a single element a_i was included in the tree without revealing its position in the list.

Formally, the Merkle tree technique consists of three algorithms (MerkleTree , getMerklePath , ReconstructRoot) with access to a common hash function $\text{H}_{\text{Coll}} : \{0, 1\}^* \rightarrow \{0, 1\}^{2^\lambda}$.

- $\text{MerkleTree}(A) \rightarrow (\text{root}, \text{tree})$: On input a list of 2^k elements $A = (a_1, \dots, a_{2^k})$, with $k \in \mathbb{N}$, it constructs a binary tree of height k with $\{l_i = \text{H}_{\text{Coll}}(a_i)\}_{i \in [2^k]}$ as its leaf nodes, and where every internal node h with children h_{left} and h_{right} equals the hash digest of a concatenation of its two children. While it is standard to consider the concatenation $h_{\text{left}} \| h_{\text{right}}$, we consider a variation which consists in ordering the two children according to the lexicographical order (or any other total order on binary strings). We denote by $(h_{\text{left}}, h_{\text{right}})_{\text{lex}}$ this modified concatenation. The algorithm then outputs the root root of the Merkle tree, as well as a description of the entire tree tree .

⁸ note that here we take into consideration also the case in which $\text{pk}_{\text{imp}} \notin \text{R}^*$

- $\text{getMerklePath}(\text{tree}, I) \rightarrow \text{path}$: On input the description of a Merkle tree tree and an index $i \in [2^k]$, it outputs the list path , which contains the sibling of l_i (i.e. a node, different from l_i , that has the same parent as l_i), as well as the sibling of any ancestor of l_i , ordered by decreasing height.
- $\text{ReconstructRoot}(a, \text{path}) \rightarrow \text{root}$: On input an element a in the list of elements $A = (a_1, \dots, a_{2^k})$ and $\text{path} = (n_1, \dots, n_k)$, it outputs a reconstructed root $\text{root}' = h_k$, which is calculated by putting $h_0 = \text{H}_{\text{Coll}}(a)$ and defining h_i for $i \in [k]$ recursively as $h_i = \text{H}_{\text{Coll}}((h_{i-1}, n_i)_{\text{lex}})$.

If the hash function H_{Coll} that is used in the Merkle tree is collision-resistant, then the following easy lemma implies that the Merkle tree construction is *binding*, i.e. that one cannot construct a path that “proves” that a value $b \notin A = (a_1, \dots, a_N)$ is part of the list A that was used to construct the Merkle tree without breaking the collision-resistance of the underlying hash function H_{Coll} .

Lemma B.1 (Binding for Merkle Tree). *There is an efficient extractor algorithm that, given the description tree of a Merkle tree (having root root and constructed using the list of elements A) and (b, path) such that $b \notin A$ and $\text{ReconstructRoot}(b, \text{path}) = \text{root}$, outputs a collision for the hash function H_{Coll} .*

The use of the lexicographical order to concatenate two children nodes in the Merkle tree construction implies that the output path of the getMerklePath algorithm information-theoretically hides the index $i \in [N]$ given as input. Formally, we have the following.

Lemma B.2 (Index Hiding for Merkle Tree). *Let $N \in \mathbb{N}$ be a power of 2, D, D' be two arbitrary distributions over $\{0, 1\}^*$ and D_I , with $I \in [N]$, be the distribution defined as*

$$D_I = \left[(a_I, \text{path}, \text{root}) \left| \begin{array}{l} a_I \leftarrow D, \\ a_i \leftarrow D' \quad \forall 1 \leq i \neq I \leq N, \\ (\text{tree}, \text{root}) \leftarrow \text{MerkleTree}(A), \\ \text{path} \leftarrow \text{getMerklePath}(\text{tree}, I) \end{array} \right. \right]$$

where $A = (a_1, \dots, a_N)$. Then we have $D_I = D_J$ for all $I, J \in [N]$.