SQIPrime: A Dimension 2 Variant of SQISignHD with Non-smooth Challenge Isogenies

Max Duparc(✉) and Tako Boris Fouotsa

EPFL, Lausanne, Switzerland
{max.duparc,tako.fouotsa}@epfl.ch

Abstract. We introduce SQIPrime, a post-quantum digital signature scheme based on the Deuring correspondence and Kani's Lemma. Compared to its predecessors that are SQISign and especially SQISignHD, SQIPrime further expands the use of high dimensional isogenies, already in use in the verification in SQISignHD, to all its subroutines. In doing so, it no longer relies on smooth degree isogenies (of dimension 1). Intriguingly, this includes the challenge isogeny which is also a non-smooth degree isogeny, but has an accessible kernel. The fact that the isogenies do not have rational kernel allows to fit more rational power 2 torsion points which are necessary when computing and representing the response isogeny. SQIPrime operates with prime numbers of the form $p = 2^\alpha f - 1$.

We describe two variants of SQIPrime. SQIPrime4D which incorporates the novelties described above and uses dimension 4 isogenies to represent the response isogeny. The runtime of higher dimensional isogeny computation is exponential in the dimension, hence the smaller the dimension the better for efficiency. The second variant, SQIPrime2D, solely uses dimension 2 isogenies. This is achieved by setting the degree of the secret isogeny to be equal to that of the challenge isogeny and further exploiting Kani's Lemma. SQIPrime2D is more efficient compared to SQIPrime4D and to SQISignHD, at the cost of being comparatively less compact, but still very compact compared to non isogeny based post-quantum signatures.

Keywords: Isogenies · SQISign · SQISignHD · Kani's Lemma · SQIPrime

1 Introduction

The interest of isogeny based signature schemes is that they provide compact post-quantum signatures. This property, which comes at the cost of a greater computational cost, motivated their research. Among the early propositions of isogeny based signature schemes such as [6,16,49], was GPS [27] that specifically relied on Deuring correspondence [20]. Its ideas were expanded and improved in 2020 by De Feo, Kohel, Leroux, Petit and Wesolowski to create

K.-M. Chung and Y. Sasaki (Eds.): ASIACRYPT 2024, LNCS 15486, pp. 396–429, 2025.
https://doi.org/10.1007/978-981-96-0891-1_13

the SQISign protocol in [18]. As of today, SQISign is the only isogeny based candidate at the NIST [38] post-quantum cryptography standardization effort. In 2023, Dartois, Leroux, Robert and Wesolowski proposed SQISignHD [11], a variant of SQISign utilising Kani's Lemma [29] for verification. Both SQISign (and follow-ups [19,46]) and SQISignHD are, as of today, the two most compact post-quantum signatures, of respective size 177B for SQISign and 109B for SQISignHD for 128 bits of security.

Kani's Lemma and high dimensional isogenies (originally used in [8,34,44] to prove that SIDH [17,28] was insecure by leveraging accessible images of torsion points) are used in SQISignHD to solve some drawbacks of SQISign as they can be used to represent isogenies of non-smooth degree, which significantly simplifies the signature part of SQISignHD, at the cost of a more complex verification. The emergence of SQISignHD is part of a broader trend in Isogeny Based Cryptography, consisting in leveraging the new capabilities enabled by Kani's Lemma, a trend that birthed many new cryptographic schemes such as SQISignHD [11], FESTA and QFESTA [3,36], IS-CUBE [35], SCALLOP-HD [9], DeuringVRF [33], SILBE [22] or POKE [1]. Kani's lemma has also been recently used to design a new ideal-to-isogeny algorithm [39] for the SQISign signature scheme.

As mentioned above, the main input in SQISignHD is the use of high dimensional isogenies to represent the response. In SQISign, the secret key is an isogeny $\tau : E_0 \to E_A$, where E_0 has j–invariant 1728. The commitment is a curve E_1 obtained by computing an isogeny $\psi : E_0 \to E_1$ and the challenge is an isogeny $\varphi : E_1 \to E_2$. The response is an isogeny $\sigma : E_A \to E_2$ (see left-hand side of Fig. 1). The isogeny σ is in fact a long smooth isogeny of degree roughly $p^{15/4}$, obtained through a more efficient variant [18,19] of the KLPT algorithm [30]. The use of the KLPT algorithm and the fact that the degree of the response isogeny σ is roughly $p^{15/4}$ implies that one needs to use primes with as much accessible (defined over a small extension of $\mathbb{F}_p$) smooth torsion as possible. This is one of the biggest constraints in SQISign that was solved in SQIsignHD.

The attacks [8,34,44] on SIDH/SIKE (and any other isogeny based protocol revealing images of smooth order torsion points such as [10,14,25]) led to a new method for representing isogenies of generic degree [43]. In fact, an evaluation of an isogeny on torsion points of large (with respect to the degree of the isogeny) smooth order is a representation of this isogeny. In SQISignHD, from the knowledge of the endomorphism rings of the curves at play, the signer samples a relatively short (but non-smooth) response isogeny σ and evaluates it on torsion points of smooth order. This evaluation is then returned to the verifier as the response. Since this evaluation represents the isogeny, the verifier can efficiently check that the data received represents an isogeny $\sigma : E_1 \to E_2$. Note that here, the response goes from E_1 to E_2 while the challenge goes from E_A to E_2, this change is made for a more convenient implementation. This brings several relaxations, among which the change of the base prime p to an SIDH prime: $p = 2^a 3^b f - 1$. In SQISign, the most computationally involved part is transforming the ideal obtained from KLPT into an isogeny, this is done during the signing process. In SQISignHD, signing is somewhat easier since the KLPT algorithm is avoided, but the verification is computationally involved. In fact, in order to validate that the evaluation returned by the signer represents an isogeny $\sigma : E_1 \rightarrow E_2$, one needs to compute and evaluate an isogeny in higher dimension: 2, 4 or 8 in general. The smaller the dimension, the more efficient the computation and the evaluation are. In SQISignHD, the verification uses dimension 4 isogenies. There is a huge efficiency gap between dimension 4 isogenies and dimension 2 isogenies [11,12,31,45]. Hence, in the quest for better efficiency, it becomes natural to ask the following question:

Can one design a variant of SQISignHD that uses only dimension 1 and/or dimension 2 isogenies?

Contributions. In this paper, we answer the question above in the affirmative, by describing SQIPrime, a derivative of SQISignHD. To do so, we first extend the use of Kani's Lemma to both key generation and commitment, by adapting the **RandIsogImages** algorithm from QFESTA [36]. Next, we modify the challenge isogeny generation in such a way that the verifier can use non-smooth degree isogenies, by sampling solely the kernel generator of this isogeny. The signer/prover can then use the techniques introduced by Leroux [33] to compute this challenge isogeny and include it in the response. As a consequence, we use primes of the form $p = 2^\alpha f - 1 = 2Nq + 1$ where q is the degree of the challenge. These changes induce numerous adaptations and optimizations throughout the protocol. In order to ease understanding and not apply all the numerous changes at once, we propose two variants of SQIPrime: SQIPrime4D and SQIPrime2D.

In SQIPrime4D, we incorporate the most basic changes to SQISignHD, without necessarily aiming for a better efficiency. These changes include: the use of an adaptation (**KaniDoublePath**, Sect. 3.1) of the **RandIsogImages** algorithm from QFESTA [36] for key generation and commitment, and the use of a non-smooth degree isogeny for commitment. More precisely, let $\tau : E_0 \rightarrow E_A$, $\psi : E_0 \rightarrow E_1$, $\varphi : E_A \rightarrow E_2$ and $\sigma : E_1 \rightarrow E_2$ be the secret, commitment, challenge and response isogenies in SQISignHD. In SQIPrime4D, τ and ψ are generated using the **KaniDoublePath** algorithm. For the challenge, the verifier samples a uniformly random scalar $a \in \mathbb{Z}_q$ where q is the degree of the commitment isogeny. The scalar a defines a point $C = P + [a]Q$ where (P, Q) is a specified basis of $E_A[q]$. The signer/prover uses the techniques in the DeuringVRF [33] to translate C into its corresponding ideal I_φ, which is in fact the ideal corresponding to the challenge isogeny $\varphi : E_A \rightarrow E_2$. From here, they recover the endomorphism ring of E_2, solve for a short isogeny $\sigma : E_2 \rightarrow E_1$ (note that this is the dual of the response in the original SQISignHD), and evaluate $\kappa = \sigma \circ \varphi$ on the 2^α-torsion points (this is illustrated in Fig. 2). The evaluation of $\kappa = \sigma \circ \varphi$ is then returned to the verifier as the response. The verifier checks that the data they received represents an isogeny $\kappa : E_A \rightarrow E_1$ of degree qd whose kernel contains $C = P + [a]Q$ and, q and d are co-prime. This proves that κ factors through the challenge $\varphi : E_A \rightarrow E_2$ whose kernel was sampled by the verifier. The verification is performed using dimension 4 isogenies. In SQIPrime2D, we implement further adjustments in order to use only dimension 2 isogenies.

The main obstacle when representing isogenies in dimension 2 is the need of an auxiliary isogeny. To represent the isogeny $\kappa := \sigma \circ \varphi : E_A \rightarrow E_1$ of degree qd returned in SQIPrime4D in dimension 2, we need an auxiliary isogeny $\delta : E_A \rightarrow E_\delta$ of degree $2^\alpha - qd$. Hence, the goal of all the changes we will operate from now on will be to enable an efficient computation of such an auxiliary isogeny. The main change consists in fixing the degree of the secret isogeny τ to q, the same degree as that of the challenge isogeny φ, and making sure that this degree is prime. Once this is done, we sample an endomorphism $\gamma \in \text{End}(E_0)$ of degree $d(2^\alpha - dq)$, and compose it with the secret isogeny $\tau : E_0 \rightarrow E_A$ to obtain an isogeny $\tau \circ \gamma : E_0 \rightarrow E_A$ of degree $dq(2^\alpha - dq)$. This isogeny can be seen as the composition of two isogenies of degree dq and $2^\alpha - dq$ respectively. We then use Kani's Lemma to recover the pushforward of the isogeny of degree $2^\alpha - dq$ in such a way that its domain is E_A, and its codomain is some curve E_δ which is computed at the same time. This pushforward is used as the sought auxiliary isogeny, allowing us to have a variant SQIPrime2D which only uses dimension 2 isogenies. The SQIPrime2D identification scheme is illustrated in Fig. 3.

The key generation in SQIPrime2D requires two dimension 2 isogeny computation and evaluation. The signing process requires two dimension 2 isogeny computations and evaluations, one for the commitment isogeny and another for generating the auxiliary isogeny. The verification requires one dimension 2 isogeny computation and evaluation, bringing it up to a total of three dimension 2 isogeny computations and evaluations for the signature and verification. Given the current efficiency gap between dimension 2 and dimension 4 isogenies, we expect SQIprime2D to be more efficient compared to SQISignHD. This is to be confirmed with a more advanced implementation of SQIprime2D, task that we leave as future work.

In order to prove the security of SQIPrime4D and SQIPrime2D, we assume that the codomain of an isogeny computed using the **KaniDoublePath** algorithm is computationally indistinguishable from a random supersingular curve. Once this assumption is made, we reduce the security of SQIPrime4D and SQIPrime2D to the Supersingular Endomorphism problem in the RUCGDIO or RUCODIO+AIO models respectively, models that we introduce and which are translations of the RUDGIO model (introduced in the context of SQISignHD) into the context of SQIPrime4D and SQIPrime2D respectively.

Related Work. While this work was under finalisation, we became aware of two other concurrent but independent projects that were trying to answer the same open question we answer in the paper. The first project is from Nakagawa and Onuki, named SQISign2D-East [37] and the second one is from Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert and Wesolowski, named SQISign2D-West [2]. Interestingly, all three papers adopt substantially different approaches to solve this problem.

- Our mechanism mainly relies on the primality of the challenge isogeny φ and on the fact that it has the same degree as our secret isogeny τ.

- The SQISign2D-East [37] mechanism uses Eichler modules [32, Definition 1.2.7] to sample endomorphisms over E_0 that can also be interpreted as endomorphisms over E_A. The auxiliary isogeny $\delta : E_A \to E_\delta$ is then generated using such endomorphisms on E_A.
- Finally, the SQISign2D-West [2] mechanism merges **RandIsogImages** with Clapoti [40] to design a new efficient algorithm to evaluate random ideals. This algorithm is then used to compute the auxiliary isogeny by sampling its ideal, composing it with the commitment and challenge ideals, evaluating the composition. Using the knowledge of the commitment and challenge isogenies, the auxiliary isogeny is retrieved.

We wholeheartedly recommend the reader to delve into these two papers (after completing ours, naturally).

Outline. The remainder of this paper is organised as follows. In Sect. 2, we give a quick recall on the architecture of both SQISign and SQISignHD, together with a reminder of the standard algorithms in Isogeny Based Cryptography that we use to define SQIPrime. In Sect. 3, we will introduce special tools that we will need to construct both SQIPrime4D and SQIPrime2D. In Sect. 4, we give the detailed construction of SQIPrime4D, together with an analysis of its security in Sect. 5. Similarly, we give the detailed specification of SQIPrime2D in Sect. 6, with its security analysis in Sect. 7. Finally, we discuss in Sect. 8 how to find adequate parameters for both SQIPrime4D and SQIPrime2D and have a word about their foreseen efficiency.

2 Background

We assume some familiarity with Isogeny Based Cryptography. We provide in [21, Appendix A] a concise overview of isogenies, Deuring correspondence, and Kani's Lemma. For a more comprehensive exploration, we recommend referring to De Feo's notes [13] and Silverman's book [47] for a general understanding of elliptic curves and isogenies. For insights into the Deuring Correspondence, Leroux's thesis [32] is an excellent resource, while Robert's attack on SIDH [43,44] provides valuable details on Kani's Lemma.

Throughout this paper, we denote by λ the security parameter. Let p be a prime, $\mathbb{F}_p$ is the finite field of cardinality p. We denote as E_0 the curve with j-invariant 1728 given by $y^2 = x^3 + x$. If $p = 3 \mod 4$, then it is supersingular and its endomorphism ring correspond to the maximal order $\mathcal{O}_0 = \mathbb{Z} + \mathbf{i}\mathbb{Z} + \frac{\mathbf{i}+\mathbf{j}}{2}\mathbb{Z} + \frac{1+\mathbf{ij}}{2}\mathbb{Z}$ with $\mathbf{i} : (x, y) \to (-x, \sqrt{-1}y)$ and $\mathbf{j} = \pi$ the Frobenius endomorphism. This is an evaluation basis[1] denoted $\mathfrak{O}_0$.

[1] An evaluation basis [11, Definition A.4.1] consists in an isomorphism between the endomorphism ring and a maximal order such that every element of the basis is efficiently computable.

2.1 Standard Algorithms

SQIPrime, even more profoundly than SQISign and SQISignHD, heavily relies on the different efficient representations [11, Definition 1] of isogenies and more specifically the kernel, ideal and high dimensional representations. To do so, it uses the following standard algorithms in Isogeny Based Cryptography:

- **KernelToIsogeny**: Takes as input E a supersingular curve and $K \in E[d]$ and returns ϕ the isogeny of degree d whose kernel is generated by K together with E', its codomain. To do so, it uses Vélu's Formulas [48] and factorises ϕ as a composition of prime degree isogenies. To be efficient, d needs to be smooth.[2]
- **CanonicalTorsionBasis**: Takes as input E a supersingular curve and N an integer such that $N|(p^2-1)$ and returns $\langle P, Q\rangle = E[N]$. To do so, it simply samples points at random in $E(\mathbb{F}_{p^2})$ or its quadratic twist and multiplies it by the right cofactor. To ensure that this method is deterministic, the sampling is performed deterministically using the Elligator algorithm [5].
- **KernelToIdeal** [11, Algorithm 9]: Takes as input $\mathfrak{O}_E$ an evaluation basis of $\mathrm{End}(E)$ and K a generator of the kernel of an isogeny ϕ of smooth degree d and returns I_ϕ.
- **FullRepresentInteger** [32, Algorithm 4]: Takes as input a number $N > 4p$ and returns $\gamma \in \mathcal{O}_0$ an endomorphism of E_0 of norm N. Note that the successful termination of this algorithm relies on plausible heuristics. We refer to [32, Section 3.1] for further details.
- **EvalTorsion** [11, Algorithm 11]: It takes as input $\mathfrak{O}_F$ an evaluation basis of $\mathrm{End}(F)$, $\rho_1 : F \to E$ of degree $d_1, \rho_2 : F \to E'$ of degree d_2, both efficiently computable isogenies together with their respective ideals I_1 and I_2. It also takes as input J an $(\mathcal{O}_E, \mathcal{O}_{E'})$-ideal of norm N co-prime to d_1 and d_2. It outputs $\phi_J(P)$, with P any point whose order is co-prime to $d_1 d_2$.
- **RandomEquivalentIdeal** [32, Algorithm 6]: It takes as input a $(\mathcal{O}_E, \mathcal{O}_F)$-ideal I and returns J another $(\mathcal{O}_E, \mathcal{O}_F)$-ideal such that $n(J)$ is a "small" prime, meaning that $n(J) \simeq \sqrt{p}$ with extremely high probability, as shown in [32, Lemma 3.2.3 & Lemma 3.2.4].
- **HDKernelToIsogeny**: This is an high dimensional equivalent to **KernelToIsogeny**. Depending on the dimension, it can be based upon theta structures [11,12,42], or over Kummer surfaces [45].

2.2 SQISign and SQISignHD

The SQISign and SQISignHD signature algorithms are in fact Σ-protocols that are transformed into digital signature schemes using the Fiat-Shamir transform [23], rendering them Universally Unforgeable under Chosen Message Attacks (UU-CMA) secure in the Random Oracle Model (ROM). The underlying Σ-protocols are built upon the Deuring correspondence, hence the acronym SQIS

[2] Note that this algorithm, as presented here, is not optimal. Among the important improvements on those computations, see [17] and [4].

for Short Quaternion Identification Scheme. The security of both protocols relies on the hardness of the *one endomorphism problem* (Problem 1). The one endomorphism problem was recently [41] shown to be equivalent to the endomorphism ring problem (Problem 2), a central problem in isogeny based cryptography, which is believed to be hard for both classical and quantum adversaries.

Problem 1. Let E be a random supersingular curve defined over $\mathbb{F}_{p^2}$, find a nontrivial (not in $\mathbb{Z}$) endomorphism of E.

Problem 2. Let E be a random supersingular curve defined over $\mathbb{F}_{p^2}$, compute the endomorphism ring $\mathrm{End}(E)$ of E.

The main idea behind SQISign and SQISignHD is to prove the knowledge of the endomorphism ring $\mathrm{End}(E_A)$ of E_A, a supersingular curve. In SQISign, the fact that the prover knows $\mathrm{End}(E_A)$ enables them to find a connecting isogeny between E_A and any other curve E_2, provided that they also know $\mathrm{End}(E_2)$. The idea is then to let E_2 be chosen as the challenge by the verifier, by computing a random isogeny $\varphi : E_1 \to E_2$ where E_1 was generated by the prover (who hence knows its endomorphism ring $\mathrm{End}(E_1)$). Using φ, the prover can retrieve $\mathrm{End}(E_2)$ and respond with an isogeny $\sigma : E_A \to E_2$ that can be easily verified. This is illustrated in Fig. 1. The high level picture in SQISignHD is similar with a minor exception that the domain of φ and σ are interchanged for efficiency reasons. The main difference between SQISign and SQISignHD consists in how the response isogeny σ is computed and represented. The first returns a very long smooth isogeny through a sequence of kernels, while the second uses high dimension isogenies to represent a relatively short but non-smooth isogeny.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\psi"] \arrow[d, dashed, "\tau"'] & E_1 \arrow[d, "\varphi"] \\
E_A \arrow[r, "\sigma"'] & E_2
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
E_0 \arrow[r, dashed, "\psi"] \arrow[d, dashed, bend right, "\tau'"'] \arrow[d, dashed, bend left, "\tau"] & E_1 \arrow[d, "\sigma"] \\
E_A \arrow[r, "\varphi"'] & E_2
\end{tikzcd}$$

Fig. 1. Diagrams of SQISign (left) and SQISignHD (right). The prover is in blue and the verifier is in red. Dashed isogenies are secrets. (Color figure online)

SQISign: To construct σ the connecting isogeny, SQISign uses a variant of the **KLPT** [30] named the **SigningKLPT** [18, Algorithm 5]. The ideal I_σ it retrieves is smooth, as its norm is a large power of 2 of size $O(p^{15/4})$. To be efficiently computed, σ is represented as a composition of isogenies with rational kernel generator. Transcribing I_σ to these kernels is done using **IdealToIsogeny**

[19, Algo. 7]. This **IdealToIsogeny** step requires a lot of smooth torsion, reason why the prime p is such that $2^\ell T | p^2 - 1$ with $T \simeq p^{5/4}$ and T smooth. Finding such primes is *difficult* and T often has prime factors in the order of 10^3. Those big factors significantly slow down the signing procedure, as several T isogenies have to be computed throughout **IdealToIsogeny**. On the other hand, the verification of SQISign is very efficient, as it essentially consists in computing a sequence of isogenies of degree 2^ℓ from their kernels.

SQISignHD: On the other hand, SQISignHD uses the **RandomEquivalentIdeal** to compute σ. The response isogeny is therefore short $O(\sqrt{p})$ but not smooth. It is given to the verifier using high dimension representation [43]. This shift to high dimension isogenies considerably speeds up the signature part of SQISignHD but shifts most of the expensive computation to the verification that has to use Kani's Lemma in dimension 4. To be efficient, SQISignHD uses "SIDH-like" prime, that are easy to find. We refer to [11] for further details.

3 Introduced Techniques

Before jumping into SQIPrime, we detail two new techniques that we will use to construct our variant of SQISignHD.

1. The first tool is called **KaniDoublePath**, a variant of **DoublePath** [11, Section 3.3] that uses Kani's Lemma to sample two (possibly non-smooth) isogenies between E_0 and E_A of co-prime degrees. This algorithm is a modification of the **RandIsogImages** [36, Algorithm 2], as it additionally computes the corresponding ideals of these isogenies. We also describe a variant **ExtKaniDoublePath** that relies on endomorphisms of greater norm.
2. The second is a method to compute, given K a generator of the kernel of an isogeny, the corresponding ideal even when the degree of this isogeny is non-smooth. This method is an adaptation of the work of Leroux on DeuringVRF [33] and allows us to use large non-smooth degree isogenies as challenge isogeny in SQIPrime.

3.1 KaniDoublePath

The main idea behind **KaniDoublePath** is, similarly to the **DoublePath** algorithm, to construct two isogenies of co-prime degree between E_0 and another supersingular curve E. The main interest of **KaniDoublePath** lies in the fact that those isogenies are not necessary smooth.

To perform the **KaniDoublePath**, we first use **FullRepresentInteger** to find an endomorphism $\gamma \in \mathrm{End}(E_0)$ with $\deg(\gamma) = \ell(N - \ell)$ with ℓ, N co-prime and N smooth. We can decompose γ as $\gamma = \rho \circ \tau$ with $\deg \tau = \ell$ and $\deg \rho = N - \ell$. Using Kani's Lemma, we compute the dimension 2 isogeny $F : E_0 \times E_0 \to E \times E'$ given by the following diagram and kernel:

$$\begin{tikzcd}
E \arrow[r, "\widehat{\tau}"] \arrow[d, "\rho"'] & E_0 \arrow[d, "\widehat{\tau}_*\rho"] \arrow[dl, "\gamma"] \\
E_0 \arrow[r, "\rho_*\widehat{\tau}"'] & E'
\end{tikzcd}$$

$$\ker(F) = \left\{ \big([-\ell](P), \gamma(P)\big) \,\middle|\, P \in E_0[N] \right\} \text{ with } F := \begin{pmatrix} \tau & -\widehat{\rho} \\ \widehat{\tau}_*\rho & \rho_*\widehat{\tau} \end{pmatrix}$$

We can therefore efficiently evaluate both τ and $\widehat{\rho}$ at any points of E_0 by writing $\tau(-) = F(-,0)_1$ and $\widehat{\rho}(-) = -F(0,-)_1$. Additionally, we also retrieve I_τ and I_ρ the ideal corresponding to τ and ρ as $I_\tau = \mathcal{O}_0\gamma + \mathcal{O}_0\ell$ and $I_\rho = \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0(N-\ell)$. The full process is summarized in Algorithm 1.

One may ask if a curve generated using **KaniDoublePath** has the same distribution as a curve generated by sampling a random cyclic kernel of size ℓ and computing the corresponding isogeny. In practice, if the degree $N-\ell$ of the byproduct isogeny ρ is not way larger than p, it may happen that for some curve E which is ℓ-isogenous to E_0, there exists no isogeny of degree $N-\ell$ between E_0 and E, meaning that E will never be returned by **KaniDoublePath**. We describe **ExtKaniDoublePath**, a variation of **KaniDoublePath** in which the degree of the byproduct isogeny ρ is larger, hence increasing the chances that there exists such an isogeny between E_0 and any curve which is ℓ-isogenous to E_0, hence reducing the gap between the two distributions.

Algorithm 1. KaniDoublePath

```
Input: $\mathfrak{O}_0$ the evaluation basis of $\mathrm{End}(E_0)$ with $\langle P, Q\rangle$ a basis of $E_0[N]$ and $\ell$ such that $\gcd(\ell, N) = 1$ and $\ell(N-\ell) > p$ with $N$ smooth.
Output: $\tau$,$\widehat{\rho} : E_0 \to E$ isogenies of respective degree $\ell$ and $N-\ell$, together with $I_\tau$ and $I_{\widehat{\rho}}$ their ideals.
$\gamma \leftarrow$ FullRepresentInteger($\mathfrak{O}_0, \ell(N-\ell)$)
$\mathsf{B} \leftarrow \{([-\ell]P, \gamma(P)), ([-\ell]Q, \gamma(Q))\}$
$F \leftarrow$ HDKernelToIsogeny($E_0^2, \mathsf{B}$)
$I_\tau \leftarrow \mathcal{O}_0\gamma + \mathcal{O}_0\ell$
$I_{\widehat{\rho}} \leftarrow \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0(N-\ell)$
return $\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}}$        ▷ $\tau(-) = F(-,0)_1$ and $\widehat{\rho}(-) = -F(0,-)_1$
```

The concept behind **ExtKaniDoublePath** closely resembles that of **KaniDoublePath**, albeit with a slight variation. Instead of operating with $\gamma \in \mathrm{End}(E_0)$ of norm $\ell(N-\ell)$, **ExtKaniDoublePath** involves working with $\gamma \in \mathrm{End}(E_0)$ of norm $\ell(N'-\ell)\big(N - \ell(N'-\ell)\big)$, where N and N' are smooth. Consequently, we have $\deg(\rho) = (N'-\ell)\big(N-\ell(N'-\ell)\big)$. Both τ and $\widehat{\rho}$ are computed by applying Kani's Lemma twice:

1. Initially, we decompose γ into $\gamma = \rho_1 \circ \rho_2 \circ \tau$ where ρ_1 has degree $N - \ell(N' - \ell)$ and $\rho_2 \circ \tau$ has degree $\ell(N' - \ell)$, and we assess $\rho_2 \circ \tau$ over $E_0[N']$.
2. Subsequently, we further break down $\rho_2 \circ \tau$ of degree $\ell(N' - \ell)$ into τ and $\widehat{\rho_2}$ of degree ℓ and $N' - \ell$ respectively, allowing for the computation of $\widehat{\rho}$ as a composition of $\widehat{\rho_1}$ and $\widehat{\rho_2}$.

You may find below the commutative diagram of the **ExtKaniDoublePath**. The first use of Kani's Lemma is in blue and the second is in red.

$$\begin{tikzcd}
 & E_0 \arrow[rr, "{[\widehat{\rho_2 \circ \tau}]_* \rho_1}"] \arrow[dd, "\rho_2 \circ \tau"] \arrow[dr, "\widehat{\tau}_* \rho_2"] \arrow[ddrr, bend left, "\gamma"] & & E'_1 \\
E \arrow[ur, "\widehat{\tau}"] \arrow[dr, "\rho_2"'] & & E' & \\
 & E_1 \arrow[ur, "{[\rho_2]_* \widehat{\tau}}"] \arrow[rr, "\rho_1"'] & & E_0 \arrow[uu, "{[\rho_1]_* \widehat{\rho_2 \circ \tau}}"']
\end{tikzcd}$$

Algorithm 2. ExtKaniDoublePath

Input: $\mathfrak{O}_0$ an evaluation basis of $\text{End}(E_0)$ with $\langle P, Q \rangle$ a basis of $E_0[N]$, $\langle P', Q' \rangle$ a basis of $E_0[N']$ and ℓ such that $\gcd(\ell, N) = \gcd(\ell, N') = 1$ and $\ell(N' - \ell)(N - \ell(N' - \ell)) > p$ with N, N' smooth.

Output: τ,$\widehat{\rho} : E_0 \to E$ isogenies of respective degree ℓ and $(N' - \ell)(N - \ell(N' - \ell))$, together with I_τ and $I_{\widehat{\rho}}$ their ideals.

```
$\gamma \leftarrow$ **FullRepresentInteger**$(\mathfrak{O}_0, \ell(N' - \ell)(N - \ell(N' - \ell))$
$\mathsf{B}_1 \leftarrow \{([-\ell(N' - \ell)]P, \gamma(P)), ([-\ell(N' - \ell)]Q, \gamma(Q))\}$
$F_1 \leftarrow$ **HDKernelToIsogeny**$(E_0^2, \mathsf{B}_1)$        ▷ $\tau \circ \rho_2(-) = F_1(-, 0)_1$
Find $E_1$ the codomain of $(\widehat{\rho_1})$
$\mathsf{B}_2 \leftarrow \{([N' - \ell]P', \tau \circ \rho_2(P')), ([N' - \ell]Q', \tau \circ \rho_2(Q'))\}$
$F_2 \leftarrow$ **HDKernelToIsogeny**$(E_0 \times E_1, \mathsf{B}_2)$
$I_\tau \leftarrow \mathcal{O}_0\gamma + \mathcal{O}_0\ell$
$I_{\widehat{\rho}} \leftarrow \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0(N' - \ell)(N - \ell(N' - \ell))$
**return** $\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}}$        ▷ $\tau(-) = -F_2(-, 0)_1$ and $\widehat{\rho}(-) = F_2(0, -)_1 \circ -F_1(0, -)_1$
```

Remark 1. In **KaniDoublePath** and **ExtKaniDoublePath**, and in other algorithms throughout this paper, we return isogenies and their ideal representations. In practice, during implementation, instead of returning an isogeny, one usually returns its evaluation on some relevant torsion point basis. These torsion point images are used later on to evaluate the isogeny on points lying in the same torsion group.

We will rely on the following assumptions when discussing the security of SQIPrime.

Assumption 1. *The distribution of E the codomain of τ and $\widehat{\rho}$, returned by **KaniDoublePath** (N, P, Q, ℓ) with ℓ a random prime smaller than $\sqrt{p}$ is computationally indistinguishable from the distribution of E sampled randomly among all supersingular curves.*

Assumption 2. *The distribution of an isogeny $\tau : E_0 \rightarrow E$ returned by **ExtKaniDoublePath** $(N, P, Q, N', P', Q', \ell)$ with $\ell < \sqrt{p}$ a random prime is computationally indistinguishable from the distribution of $\tau : E_0 \rightarrow E$ sampled randomly among isogenies of degree ℓ and of domain E_0.*

3.2 KernelToIdeal for Generic Degree Isogenies

Looking at the details of **KernelToIdeal** [11, Algorithm 9], we see that it makes extensive use of discrete logarithms over $E[d]$, with d being the degree of the isogeny for which the representing ideal is being computed. To be efficient via standard methods (i.e. Pohlig-Hellman), this method requires d being smooth. We therefore need another method for isogenies of generic degree. The idea proposed by Leroux in [33] is to use the knowledge of the endomorphism ring of E to construct a *precomputed basis* of $E[d]$.

Definition 1. *Let E be any supersingular curve. The tuple (P, Q, ι, I_P) is a **precomputed basis** of $E[d]$ if the following conditions are satisfied:*

- *$P, Q \in E$ form a basis of $E[d]$.*
- *$\iota \in End(E)$ and $\iota(P) = Q$.*
- *I_P is the ideal corresponding to the isogeny of kernel $\langle P \rangle$.*

Knowledge of an evaluation basis $\mathfrak{O}_E$ of $\mathrm{End}(E)$ enables us to construct a precomputed basis using the **FindPrecomputedBasis** algorithm (Algorithm 3), proposed in [33]. In our case, we apply it to the curve E_0, where we can use the (heuristic) **FullRepresentInteger** algorithm to efficiently sample endomorphisms in $\mathcal{O}_0$ with the desired norm dN where N is co-prime to d and $p \ll dN$.

Using a precomputed basis, we can compute ideals from a kernel generator $K \in E[d]$ by applying the following lemma.

Lemma 1. *Let (P, Q, ι, I_P) be a precomputed basis of $E[d]$ and let $K = [a]P + [b]Q$ be a point in $E[d]$. Then the representing ideal of the isogeny $\phi_K : E \rightarrow E/\langle K \rangle$ is given by $I_K = [a + b\epsilon(\iota)]_* I_P$ where $\epsilon : \mathcal{O}_E \leftrightarrow End(E)$.*

Proof. This comes from the fact that $\langle K \rangle = \langle [a]P + [b]Q \rangle = \langle [a]P + [b]\iota(P) \rangle = [a + b\iota]\langle P \rangle$, meaning that $\phi_K = [a + b\epsilon(\iota)]_* \phi_P$. We then get the desired result through the Deuring correspondence. □

Algorithm 3. FindPrecomputedBasis

```
Input: $\mathfrak{O}_E = \big(\{b_i\}_{i=1}^4, \epsilon\big)$ an evaluation basis of End$(E)$ with $d$ prime.
Output: $(P, Q, \iota, I_P)$ a precomputed basis of $E[d]$.
1: Sample a random $R \in E[d]$
2: Sample $\alpha \in \mathcal{O}_E$ such that gcd $\big(n(\alpha), d^2\big) = d$
3: if $\epsilon^{-1}(\alpha)(R) = 0$ then go to step 1
4: $P \leftarrow \epsilon^{-1}(\alpha)(R)$
5: $I_P \leftarrow \mathcal{O}_E\overline{\alpha} + \mathcal{O}_E d$
6: Sample $\gamma \in \mathcal{O}_E$ such that gcd$(n(\gamma), d) = 1$
7: if $P$ and $\epsilon^{-1}(\gamma)(P)$ are linearly dependent, then go to step 6
8: return $P, \epsilon^{-1}(\gamma)(P), \epsilon^{-1}(\gamma), I_P$
```

We can thus compute the ideals corresponding to a kernel of generic order. Nevertheless, the method that we presented here requires knowing $\mathfrak{O}_E$. Most of the time, the curve E is obtained by computing an isogeny $\phi : E_0 \to E$. With the knowledge of $\mathfrak{O}_0$ and $\phi : E_0 \to E$, one can recover $\mathfrak{O}_E$, and hence determine a precomputed basis of $E[d]$ using the **FindPrecomputedBasis** algorithm. Even though this is already efficient, in Corollary 1, we describe a faster and more convenient method to translate a kernel generator $K \in E[d]$ into an ideal knowing a precomputed basis of $E_0[d]$, $\phi : E_0 \to E$ of degree co-prime to d and its corresponding ideal I_ϕ.

Corollary 1. *Let (P, Q, ι, I_P) be a precomputed basis of $E_0[d]$ and let $\phi : E_0 \to E$ be an isogeny of degree q with corresponding ideal I_ϕ such that d and q are co-prime. Let $S, T \in E$ be the respective images of P and Q by ϕ and let $K = [a]S + [b]T$ be a point in $E[d]$. Then $I_K = \big[(a + b\epsilon(\iota))I_\phi\big]_* I_P$.*

Proof. Similarly to Lemma 1, we have that

$$\langle K\rangle = [q]\langle K\rangle = \phi\widehat{\phi}\,\langle [a]S + [b]T\rangle = \phi\langle [a]\widehat{\phi}(S) + [b]\widehat{\phi}(T)\rangle = \phi\langle [aq]P + [bq]Q\rangle$$
$$= \phi\langle [a]P + [b]Q\rangle = \phi\langle [a]P + [b]\iota(P)\rangle = \phi \circ [a + b\iota]\langle P\rangle,$$

i.e. $\phi_K = [\phi \circ (a + b\iota)]_* \phi_P$ and thus $I_K = [(a + b\epsilon(\iota))I_\phi]_* I_P$. □

It's worth noting that [33] proposes using ϕ to directly generate a precomputed basis over E. Specifically, if (P, Q, ι, I_P) represents a precomputed basis over $E_0[d]$, then $\big(\phi(P), [\deg(\phi)]\phi(Q), \theta, [I_\phi]_* I_P\big)$ constitutes a precomputed basis of $E[d]$ with $\theta = \phi \circ \iota \circ \widehat{\phi}$. The significant advantage of Corollary 1 lies in its exclusive use of endomorphisms over E_0 rather than over E. This characteristic aligns more closely with our requirements in SQIPrime, making it better suited for our purposes.

4 SQIPrime4D: SQIPrime in Dimension 4

As previously stated in the introduction, SQIPrime4D further expands the use of Kani's Lemma to both KeyGen and Commit. Moreover, the challenge isogeny has non-smooth degree. Only the kernel of the challenge isogeny is sampled by the verifier. The challenge isogeny $\varphi : E_A \to E_2$ is computed by the prover, who then appends the usual response isogeny $\sigma : E_2 \to E_1$ to it to get $\kappa := \sigma \circ \varphi : E_A \to E_1$. The high dimensional representation of κ is returned to the verifier. Figure 2 illustrates the architecture of SQIPrime4D.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\psi"] \arrow[d, dashed, "\tau"'] & E_1 \\
E_A \arrow[ur, "\kappa"] \arrow[r, "\varphi"'] & E_2 \arrow[u, dashed, "\sigma"']
\end{tikzcd}$$

Fig. 2. Diagram of SQIPrime4D, prover in blue and verifier in red. Dashed isogenies are not shared. (Color figure online)

The public parameters of SQIPrime4D are defined as:

- p a prime number of the form $p = 2^\alpha f - 1 \simeq 2^{2\lambda}$ and such that $p = 2Nq + 1$, with $q \simeq 2^\lambda$. We discuss in Sect. 8 how to efficiently compute such primes.
- P_0, Q_0 a basis of $E_0[2^\alpha]$.
- $(P, Q, \iota, I_{[N]P})$ which is almost a precomputed basis over $E_0[Nq]$. (It is if we use I_P instead of $I_{[N]P}$ but this ideal is more adapted to SQIPrime4D.)
- β an integer of the form $\beta = 2\lambda + c\log(\lambda)$ with c a small constant. (See Sect. 4.2 for more details.)

They are constructed using the Setup algorithm described in Algorithm 4.

Algorithm 4. SQIPrime4D.Setup

```
Input: $1^\lambda$.
Output: $\mathsf{pp} = \big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$.
Take $p$ a prime of the form $p = 2^\alpha f - 1 \simeq 2^{2\lambda}$ such that $p - 1 = 2Nq$ with $q \simeq 2^\lambda$
  prime and $N$ co-prime to $q$
$P_0, Q_0 \leftarrow$ CanonicalTorsionBasis$(E_0, 2^\alpha)$
$(P, Q, \iota, I_P) \leftarrow$ FindprecomputedBasis$(\mathfrak{O}_0, qN)$
Compute $I_{[N]P} = I_P + \mathcal{O}_0 q$
$\beta \leftarrow \lceil 2\lambda + c\log_2(\lambda) \rceil$
$\mathsf{pp} \leftarrow \big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$
return pp
```

At a high level, the subroutines of SQIPrime4D are as follows.

- KeyGen: Compute $\tau : E_0 \to E_A$ together with its corresponding ideal I_τ using **KaniDoublePath**. Additionally, compute a matrix $\mathbf{M}$ and use it to mask the image through τ of a precomputed basis of degree qN, with $q \simeq 2^\lambda$. The curve E_A and the masked basis form the public key, while τ, I_τ and the matrix $\mathbf{M}$ form the secret key.
- Commit: The prover computes an isogeny $\psi : E_0 \to E_1$ with **KaniDoublePath** together with its ideal I_ψ and shares E_1.
- Challenge: The verifier samples a random scalar $a \in \mathbb{Z}_q$ and returns it to the prover. This scalar defines a point $C_a = P + [a]Q$ where P, Q is a specified basis of $E_A[q]$.
- Response: Using the precomputed basis over E_0 and its knowledge of I_τ, the prover retrieves I_φ, the ideal corresponding to the challenge isogeny $\varphi : E_A \to E_2$ whose kernel is given by $\ker(\varphi) = \langle C_a \rangle$. Using **RandomEquivalentIdeal**, they compute a short $(\mathcal{O}_2, \mathcal{O}_1)$-ideal I_σ corresponding to an isogeny $\sigma : E_2 \to E_1$, and construct $\kappa = \sigma \circ \varphi$, evaluate it using **EvalTorsion** and send this evaluation of κ as the response to the verifier.
- Verify: The verifier receives κ and checks using Kani's Lemma that it is valid by verifying that it is an isogeny from E_A to E_1 and that $\kappa(C_a) = 0$.

4.1 Key Generation and Commitment

Both key generation and commitment consist essentially in using **KaniDoublePath**. We take a random prime ℓ smaller than $\sqrt{p}$ and use the **KaniDoublePath** with an endomorphism of norm $\ell(2^\alpha - \ell)$ to retrieve τ in the case of **SQIPrime.KeyGen** (Algorithm 5) and ψ in **SQIPrime.Commit**. (Algorithm 6). The only significant differences between the key and commitment generation is that during the key generation, we additionally compute a masked basis of $E_A[Nq]$. To do so, we compute the image of (P, Q) through the isogeny τ and use a random matrix $\mathbf{M} \in \mathrm{GL}_2(Nq)$ to mask the torsion points. Note that this masking makes of R, S a random basis of $E_A[Nq]$.

Algorithm 5. SQIPrime4D.KeyGen

```
Input: pp = $\big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$.
Output: sk = $\big(\tau, I_\tau, \mathbf{M}\big)$, pk = $\big(E_A, (R, S)\big)$.
Sample $\ell_A \neq 2$ a random prime smaller than $\sqrt{p}$ such that $\ell_A$ co-prime with $q$
$\tau, *, I_\tau, * \leftarrow$ KaniDoublePath$(2^\alpha, P_0, Q_0, \ell_A)$
Compute $E_A = \mathrm{Im}(\tau)$
Sample a random matrix $\mathbf{M} \in \mathrm{GL}_2(Nq)$
$\binom{R}{S} \leftarrow \mathbf{M}\binom{\tau(P)}{\tau(Q)}$
return $(\tau, I_\tau, \mathbf{M})$, $\big(E_A, (R, S)\big)$
```

Algorithm 6. SQIPrime4D.Commit

```
Input: pp = $\big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$.
Output: sec = $\big(\psi, I_\psi\big)$, com = $E_1$.
Take $\ell_1 \neq 2$ a random prime smaller than $\sqrt{p}$ such that $\ell_1$ co-prime with $q$
$\psi, *, I_\psi, * \leftarrow$ KaniDoublePath$(2^\alpha, P_0, Q_0, \ell_1)$
Compute $E_1 = \mathrm{Im}(\psi)$
return $\big(\psi, I_\psi\big)$, $\big(E_1\big)$
```

4.2 Challenge and Response

Challenge. As touched on earlier, our challenge is significantly different from the challenge of SQISign and SQISignHD, as the evaluation of the challenge isogeny has been moved from the verifier to the prover. This adjustment is necessary since the verifier lacks an efficient means to evaluate this isogeny, as it only has access to the kernel representation of φ, whose degree is not smooth. The prover uses the ideal representation to construct a high dimension representation of φ that is then sent to the verifier together with the high dimension representation of the answer isogeny σ. Thus, instead of providing an isogeny of smooth degree, the challenger simply sends a challenge point $C_a \in E_A[q]$. This point is given as $a \in \mathbb{Z}_q$ such that $C_a = [N](R + [a]S)$ where R, S is the basis of $E_A[Nq]$ included in the public key. This point is the generator of the kernel of $\varphi : E_A \to E/\langle C_a \rangle = E_2$. We have $q \simeq 2^\lambda$ possible challenge isogenies.

Response. In line with SQISignHD, our objective is to compute an isogeny $\sigma : E_2 \to E_1$. However, the verifier lacks knowledge of E_2. An initial idea might be to provide the verifier with an HD representation of φ, allowing him to check that the kernels match. However, this approach requires knowledge of a map between E_0 and E_2 (or E_A and E_2), which is challenging to construct.[3] Instead of sending σ and φ separately, the idea is to send $\kappa = \sigma \circ \varphi$ and use Kani's Lemma over κ to prove that κ factors through φ, utilising the fact that $\ker(\kappa) \cap E_A[q] = \ker(\varphi)$.

First, one adapts Corollary 1 to compute $I_{C_a} = I_\varphi$. Upon receiving the challenge $\mathsf{Chal} = a$, the prover finds $b, c \in \mathbb{Z}_q$ such that $C_a = [N]\big([b]\tau(P) + [c]\tau(Q)\big)$. These scalars are given by $\binom{b}{c} = \mathbf{M}^\top \binom{1}{a}$. One then recovers I_{C_a} as

$$I_{C_a} = \big[\big(b + c\epsilon(\iota)\big) I_\tau\big]_* I_{[N]P}$$

One then computes the $(\mathcal{O}_2, \mathcal{O}_1)$-ideal $\overline{I_{C_a} I_\tau} I_\varphi$ and finds an equivalent short $(\mathcal{O}_2, \mathcal{O}_1)$-ideal J using **RandomEquivalentIdeal**. The ideal J corresponds to an isogeny $\sigma : E_2 \to E_1$ of degree d as shown in Fig. 2, with d such that $2^\beta - qd$ can be written as the sum of two squares. One sufficient condition

[3] We could use the KLPT algorithm followed by the **IdealToKernel** algorithm, but avoiding this algorithm was a primary motivation behind the development of SQISignHD.

is to ask for $2^\beta - qd = 1 \mod 4$ and to be prime. Following the discussion in [11, Section 4.2] and by using the sampling method proposed in [11, Section E.2], we expect to find a valid J after sampling $O(\lambda)$ times. Moreover, we require that d is co-prime to q. This is to prevent backtracking when composing σ and φ. Since q has very few prime factors in our case, then a few supplementary samples will allow to ensure that d and q are co-prime. For the suggested parameters (Sect. 8), the worst case is when $\lambda = 192$ where $q = 3 \cdot 7 \cdot 4803463386334137403 \cdot 116682096886878909945888202135243873061$ and that the probability that a random number shares a prime factor with q is at most 0.47. Note that β can be as large as $2\alpha \approx 2 \log p$, which means there is more than enough room to sample J with the requirements above. In practice, $\beta = 2\lambda + c \log(\lambda) \ll 2^{3\lambda}$ where c is a small constant is sufficient.

The final response is composed of the evaluation of the isogeny $\kappa = \sigma \circ \varphi$ on $E_A[2^\alpha]$ and on the point $C_2 = [a]R - S$, together with the degree d of σ. To do so, one generates a basis of $E_A[2^\alpha]$ using **CanonicalTorsionBasis**, one uses **EvalTorsion** to evaluate κ on the generated basis and C_2. The point $\kappa(C_2)$ is used to ensure the soundness of our verification. It is important to note that C_2 satisfies $\langle C_a, [N]C_2 \rangle = E_A[q]$.

Algorithm 7. SQIPrime4D.Response

```
Input: pp = $\big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$, sk = $(\tau, I_\tau, \mathbf{M})$, pk = $(E_A, (R, S))$, sec = $(\psi, I_\psi)$, com = $E_1$, chal = $a$.
Output: res = $(T, U, V, d)$ with $T, U \in E_1[2^\alpha]$, $V \in E_1[Nq]$ and $d$ the degree of $\sigma$.
1: $\binom{b}{c} \leftarrow \mathbf{M}^\top \binom{1}{a}$
2: $I_{C_a} \leftarrow [(b + c\iota)I_\tau]_* I_{[N]P}$
3: $J \leftarrow$ RandomEquivalentIdeal$(\overline{I_{C_a} I_\tau} I_\psi)$        $d \leftarrow n(J)$
4: if $gcd(d, q) \neq 1$ or $2^\beta - dq \neq 1 \mod 4$ or $2^\beta - dq$ is composite, go back to Step 3
5: $X, Y \leftarrow$ CanonicalTorsionBasis$(E_A, 2^\alpha)$
6: $C_2 \leftarrow [a]R - S$
7: $T, U, V \leftarrow$ EvalTorsion$\Big(\mathfrak{O}_0, \tau, I_\tau, \psi, I_\psi, I_{C_a} J, qd, \{X, Y, C_2\}\Big)$
8: return res = $(T, U, V, d)$                    ▷ $T = \kappa(X), U = \kappa(Y), V = \kappa(C_2)$
```

4.3 Verification

Upon receiving T, U, V, d, we want to verify that the following statement holds: *the torsion points we received define a high dimensional representation of an isogeny $\kappa : E_A \to E_1$ of degree dq such that d and q are co-prime and the isogeny κ factors through φ, meaning that* $\ker(\kappa)[q] = \langle C_a \rangle$.

To perform this verification efficiently, we use Kani's Lemma to construct the isogeny $F : E_1^2 \times E_A^2 \to E_A^2 \times E_1^2$ given by the following diagram and matrices:

$$\begin{tikzcd} E_A^2 \arrow[r, "\Sigma"] \arrow[d, "\eta"'] & E_1^2 \arrow[d, "\eta"] \\ E_A^2 \arrow[r, "\Sigma"] & E_1^2 \end{tikzcd} \qquad F := \begin{pmatrix} \tilde{\Sigma} & -\tilde{\eta} \\ \eta & \Sigma \end{pmatrix} = \begin{pmatrix} \widehat{\kappa} & 0 & -a_1 & -a_2 \\ 0 & \widehat{\kappa} & a_2 & -a_1 \\ a_1 & -a_2 & \kappa & 0 \\ a_2 & a_1 & 0 & \kappa \end{pmatrix}$$

where $\eta := \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix}$ such that $\deg(\eta) = a_1^2 + a_2^2$; $\Sigma := \mathrm{diag}(\kappa, \kappa)$. If the parameters allow us to always have enough torsion, that is we always have $dq < 2^\alpha$ or equivalently $\beta = \alpha$, then F can be computed on one go and its kernel is given by $\ker(F) = \left\{ \left(\Sigma(P), -\eta(P)\right) \middle| \; P \in E_A^2[2^\beta] \right\}$. If the parameters do not allow this, then we split the isogeny $F : E_1^2 \times E_A^2 \to E_A^2 \times E_1^2$ into two isogenies $F_1 : E_1^2 \times E_A^2 \to \Delta$ and $F_2 : \Delta \to E_A^2 \times E_1^2$ where Δ is an abelian surface, $F = F_2 \circ F_1$ with $\deg(F_i) = 2^{\beta_i}$ ($\beta_1 + \beta_2 = \beta$), $\ker(F_1) = \left\{ \left(\Sigma(P), -\eta(P)\right) \middle| \; P \in E_A^2[2^{\beta_1}] \right\}$ and $\ker(\widetilde{F_2}) = \left\{ \left(\Sigma(P), \tilde{\eta}(P)\right) \middle| \; P \in E_A^2[2^{\beta_2}] \right\}$, similarily to SQISignHD[4]. We then use the following property: let $X \in E_A$ be a point of odd order, then

$$F \begin{pmatrix} 0 \\ 0 \\ X \\ 0 \end{pmatrix} = \begin{pmatrix} [-a_1]X \\ [a_2]X \\ Y \\ 0 \end{pmatrix} \iff [2^{\beta_2}]F_1 \begin{pmatrix} 0 \\ 0 \\ X \\ 0 \end{pmatrix} = \widetilde{F_2} \begin{pmatrix} [-a_1]X \\ [a_2]X \\ Y \\ 0 \end{pmatrix}.$$

We use this equivalence on the points C_a and C_2 of respective order q and Nq.

Proposition 1. *Let* $\mathsf{pp}, \mathsf{pk}, \mathsf{com}, \mathsf{chal}$ *be a valid public key, commitment, and challenge of SQIPrime4D and let* P, Q *be the canonical basis of* $E_A[2^\alpha]$*. Let* $\overline{\mathsf{Res}}$ *be a potential response.* ***SQIPrime4D. Verify***$(\mathsf{pp}, \mathsf{pk}, \mathsf{com}, \mathsf{chal}, \overline{\mathsf{Res}}) = 1$ *implies that* $\overline{\mathsf{Res}} = (\overline{T}, \overline{U}, \overline{V}, \overline{d})$ *is such that:*

- $(P, Q, \overline{T}, \overline{U})$ *is a high dimension representation of an isogeny* $\kappa : E_A \to E_1$ *of degree* $q\overline{d}$.
- $\ker(\kappa) \cap E_A[q] = \langle C_a \rangle$.

Proof. Our proof takes inspiration from [11, Section E.5]. In fact if we assume that **SQIPrime.Verify**$(\mathsf{pp}, \mathsf{pk}, \mathsf{pub}, \mathsf{chal}, \overline{\mathsf{Res}}) = 1$, then $\overline{T}, \overline{U}, \overline{V}$ are in E_1, $\overline{F_1}$ and $\overline{F_2}$ are well-defined and have the same codomain, and the following holds:

$$[2^{\beta_2}]\overline{F_1}(0,0,C_a,0) = \widetilde{\overline{F_2}}([-a_1]C_a, [a_2]C_2, 0, 0) \implies \overline{F}(0,0,C_a,0) = ([-a_1]C_a, [a_2]C_2, 0, 0)$$
$$[2^{\beta_2}]\overline{F_1}(0,0,C_2,0) = \widetilde{\overline{F_2}}([-a_1]C_a, [a_2]C_2, \overline{V}, 0) \implies \overline{F}(0,0,C_2,0) = ([-a_1]C_a, [a_2]C_2, \overline{V}, 0).$$

From the isogeny $\overline{F}$, using ι_i and ρ_j the standard injections/restrictions of product spaces, we can construct 16 elliptic curve isogenies $\overline{F}_{i,j} = \rho_i \circ \overline{F} \circ \iota_j$ with $1 \le i, j \le 4$ such that for all $j = 1, \cdots, 4$:

[4] A slight change in the prime used in SQISignHD was suggested in [24] in order to avoid splitting the high dimensional isogeny, in the hope for a better efficiency, but we are not aware of any implementation of this variant.

Algorithm 8. SQIPrime4D.Verify

```
Input: pp = $\big(p, (P_0, Q_0), (P, Q, \iota, I_{[N]P}), \beta\big)$, pk = $(E_A, R, S)$, com = $E_1$, chal = $a$, res = $(T, U, V, d)$.
Output: 0 or 1.
if one of the points $T, U, V$ is not in $E_1$ or $\gcd(d, q) \neq 1$, return 0
$\beta_1 \leftarrow \lfloor \frac{\beta}{2} \rfloor, \beta_2 \leftarrow \lceil \frac{\beta}{2} \rceil$, $k_1 \leftarrow 2^{\alpha - \beta_1}$, $k_2 \leftarrow 2^{\alpha - \beta_2}$
$(a_1, a_2) \leftarrow$ Cornacchia$(2^\beta - qd)$
Compute $\eta$ and $\tilde{\eta}$
Compute $\{P_i\}_{0 \leqslant i \leqslant 4}$ a basis of $E_A^2[2^\alpha]$      ▷ Using CanonicalTorsionBasis
$\mathsf{B}_1 \leftarrow \big\{\big([k_1]\Sigma(P_i), [-k_1]\eta(P_i)\big)\big\}_{0 \leqslant i \leqslant 4}$      ▷ $\Sigma(P_i)$ computed using $T, U$
$\mathsf{B}_2 \leftarrow \big\{\big([k_2]\Sigma(P_i), [k_2]\tilde{\eta}(P_i)\big)\big\}_{0 \leqslant i \leqslant 4}$
$F_1 \leftarrow$ HDKernelToIsogeny$(\mathsf{B}_1)$
$\widetilde{F_2} \leftarrow$ HDKernelToIsogeny$(\mathsf{B}_2)$
if codomain$(F_1) \neq$ codomain$(\widetilde{F_2})$ do return 0      ▷ Do as [11, Section F.3]
$C_a \leftarrow [N](R + [a]S)$, $C_2 \leftarrow ([a]R - S)$
$b_1 \leftarrow [2^{\beta_2}]F_1(0, 0, C_a, 0) \overset{?}{=} \widetilde{F_2}([-a_1]C_a, [a_2]C_a, 0, 0)$
$b_2 \leftarrow [2^{\beta_2}]F_1(0, 0, C_2, 0) \overset{?}{=} \widetilde{F_2}([-a_1]C_2, [a_2]C_2, V, 0)$
return $b_1 \wedge b_2$
```

$$\sum_{i=1}^{4} \deg(\overline{F}_{i,j}) = \deg(\overline{F}) = 2^\beta$$

We focus on the case when $j = 3$. We want to demonstrate that for $i = 1, 2$, and 4, $F_{i,3} = [b_i]$, with b_i being $-a_1$, a_2, and 0, respectively. To achieve this, we utilize the Cauchy interpolation theorem. By applying the triangular inequality, we have:

$$\text{for } i = 1, 2, 4,\ \deg(\overline{F}_{i,3} - [b_i]) \leq 4 \cdot 2^\beta \approx 2^{2\lambda + c\log(\lambda) + 2} \lll 2^{3\lambda}.$$

We know that $\overline{F}_{i,3} = [b_i]$ for all points generated by $\langle C_a, C_2 \rangle$, i.e., for $Nq^2 \approx 2^{3\lambda}$ points. Thus, $\overline{F}_{1,3} = [a_1]$, $\overline{F}_{2,3} = [-a_2]$, and $\overline{F}_{4,3} = 0$. Since $\overline{F}(0, 0, C_a, 0) = ([-a_1]C_a, [a_2]C_2, 0, 0)$, we deduce that $\overline{F}_{3,3}$ is an isogeny of degree qd between E_A and E_1 such that $\overline{F}_{3,3}(C_a) = 0$. Since d and q are co-prime, then $\ker(\overline{F}_{3,3}) \cap E_A[q] = \langle C_a \rangle$. □

5 Security Analysis of SQIPrime4D

We now prove that the SQIPrime4D identification protocol described in the Sect. 4 is a Σ-protocol. To do so, we have to show that SQIPrime4D has special soundness and is Honest Verifier Zero Knowledge (HVZK). Once both are proven, applying the Fiat-Shamir transform [23] over SQIPrime4D will result in a digital signature scheme that is UU-CMA in the ROM. The extractor is constructed as follows.

Proposition 2. *Let* $(E_1, \mathsf{chal}_1, T_1, U_1, V_1, d_1)$ *and* $(E_1, \mathsf{chal}_2, T_2, U_2, V_2, d_2)$ *be 2 transcripts with identical commitment* E_1 *and* $\mathsf{chal}_1 \neq \mathsf{chal}_2$*. There exists an extractor* $\mathcal{E}$ *that, given both transcripts, can efficiently solve the one endomorphism problem (Problem 1) over* E_A*, i.e. find* $\theta_A \in End(E_A)$ *a non-trivial endomorphism.*

Proof. Our proof is very similar to [11, Proposition 17]. We can use T_1, U_1 to compute a high dimension representation of $\kappa_1 = \sigma_1 \circ \varphi_1$ and T_2, U_2 to compute a high dimension representation of $\widehat{\kappa_2} = \widehat{\sigma_2 \circ \varphi_2}$. Then, $\theta_A = \widehat{\kappa_2} \circ \kappa_1 \in \mathrm{End}(E_A)$ is non-scalar. In fact, let us assume for a moment that θ_A is a scalar. Since $\ker(\kappa_1) \cap E_A[q] = \langle C_{\mathsf{chal}_1} \rangle$ is cyclic, $\ker(\kappa_2) \cap E_A[q] = \langle C_{\mathsf{chal}_2} \rangle$ is cyclic, d_1 and d_2 are co-prime to q, then $\ker(\kappa_1) \cap E_A[q] = \ker(\kappa_2) \cap E_A[q]$, which implies that $\langle C_{\mathsf{chal}_1} \rangle = \langle C_{\mathsf{chal}_2} \rangle$. Hence $\mathsf{chal}_1 = \mathsf{chal}_2$, which is a contradiction. □

The extractor ensures us that SQIPrime4D has special soundness. Similarly to [11, Section 5.2], we construct the simulator under the assumption that we have access to the following oracle.

Definition 2. *The* Random Uniformly Constrained Good Degree Isogeny Oracle *(RUCGDIO) is an oracle that takes as input a supersingular curve* E *together with* $P \in E[q]$ *and that returns an efficient representation of* $\kappa : E \to E'$ *of degree* qd *with* d *co-prime with* q *and such that:*

- E' *is uniformly distributed over all supersingular curves.*
- κ *is uniformly distributed among all isogenies between* E *and* E' *such that* $P \in \ker(\kappa)$ *and such that* $2^\beta - qd$ *is a prime congruent to* 1 *modulo* 4 *with* d *co-prime to* q*.*

Proposition 3. *Given* pp, pk *and* chal*, there exists a simulator* $\mathcal{S}$ *with access to a RUCGDIO that simulates transcripts with a distribution that is computationally indistinguishable from the distribution of transcripts of SQIPrime4D, conditioned to* chal*.*

Proof. Given $a \in \mathbb{Z}_q$, we compute $C_a = [N](R + [a]S)$. Calling RUCGDIO over E_A and C_a, we retrieve an efficient representation of $\kappa : E_A \to E_1$ and use this representation to compute the points $A = \kappa(X), B = \kappa(Y)$, and $Z = \kappa([b]R - [a]S)$ with X, Y the canonical basis over $E_A[2^\alpha]$.

We then simply return the following transcript $(E_1, a, A, B, Z, \deg(\kappa)/q)$.

This transcript is computationally indistinguishable from a genuine transcript, as:

- Following Assumption 1, we have that a genuine E_1 or one given by RUDGIO are computationally indistinguishable.
- Following [32, Lemma 3.2.4], a genuine κ or one given by RUDGIO are computationally indistinguishable, and so does $A, B, Z, \deg(\kappa)/q$.

□

We now make the following assumption.

Assumption 3. *The one endomorphism problem (Problem 1) remains hard even when given access to RUCGDIO.*

Indeed, by definition, RUCGDIO, when given an input P, generates a random isogeny that factors ϕ_P and that is of good degree. If P is of smooth order, then RUCGDIO is in fact equivalent to the RUGDIO oracle [11, Definition 5.2.1]. Thus, the arguments of [11, Section 5.3] also applies to RUCGDIO. It is therefore reasonable to assume that RUCGDIO does not help to break the one endomorphism problem.

6 SQIPrime2D: SQIPrime in Dimension 2

In this section, we describe a version of SQIPrime which uses only dimension 2 isogenies. As touched on earlier, moving from dimension 4 isogenies to dimension 2 isogenies allows to obtain a more efficient scheme. This time, SQIPrime2D is expected to be more efficient compared to SQISignHD.

6.1 High Level Description

Recall the diagram for SQIPrime4D in Fig. 2. In order to represent $\kappa = \sigma \circ \varphi$ using Kani's Lemma in dimension 2, we need to compute and evaluate an auxiliary isogeny $\delta : E_A \to E_\delta$ of degree $2^\alpha - dq$. Since the prover knows the endomorphism ring of E_A, they could in fact compute such an isogeny by using the KLPT algorithm, but this is not an admissible way as we want to avoid using the costly KLPT algorithm.

Instead, we will use Kani's Lemma, **KaniDoublePath** and **ExtKaniDoublePath**, together with several other techniques to generate the auxiliary isogeny of degree $2^\alpha - dq$. To achieve this goal, we will operate the following change to SQIPrime4D:

the secret isogeny τ will now be of fixed[5] *degree q, which is also the degree of the challenge isogeny φ.*

With that change in mind, we now sketch how one generates an auxiliary isogeny $\delta : E_A \to E_\delta$ of degree $2^\alpha - dq$. Firstly, one samples an endomorphism $\gamma \in \mathrm{End}(E_0)$ of degree $d(2^\alpha - dq)$, and one evaluates it on the 2^α-torsion. Next, one evaluates $\tau \circ \widehat{\gamma}$ on the 2^α-torsion basis $\{P_0, Q_0\}$ of E_0. Write $\gamma = \gamma_2 \circ \gamma_1$ where γ_1 and γ_2 have degree d and $2^\alpha - dq$ respectively, and let E_0' be the codomain of γ_1. Let $\delta : E_A \to E_\delta$ be the pushforward of γ_2 through $\tau \circ \widehat{\gamma_1}$. Then E_0, E_0', E_A and E_δ are the vertices of an SIDH square where the degrees are dq and $2^\alpha - dq$. One can hence apply Kani's Lemma to compute the isogeny $\delta : E_A \to E_\delta$ and evaluate it on the 2^α-torsion points. This is illustrated in Fig. 3.

For SQIPrime2D, the public parameters are defined as follows:

[5] This already implies that the key recovery problem in SQIPrime4D and SQIPrime2D are different, since the degree of the secret isogeny in SQIPrime4D is random and is not public.

$$\begin{tikzcd}
E_0 \arrow[dd, dashed, "\eta"'] & E_0' \arrow[l, dashed, "\gamma_2"'] & \\
& E_0 \arrow[ul, dashed, "\gamma"] \arrow[u, dashed, "\gamma_1"'] \arrow[r, dashed, "\psi"] \arrow[d, dashed, bend right, "\widehat{\rho}"'] \arrow[d, dashed, bend left, "\tau"] & E_1 \\
E_\delta & E_A \arrow[l, "\delta"'] \arrow[ur, "\kappa"] \arrow[r, "\varphi"] & E_2 \arrow[u, dashed, "\sigma"']
\end{tikzcd}$$

Fig. 3. Diagram of SQIPrime2D, prover in blue and verifier in red. Dashed isogenies are not shared. (Color figure online)

- The base prime p is of the form $p = 2^\alpha f - 1 = 2Nq + 1 \simeq 2^{2\lambda}$, with $q \simeq 2^\lambda$ prime, such that: $\alpha \geq \lceil \frac{\log_2(p)}{2} + \log_2(q) \rceil + 1$.
- P_0, Q_0 is a basis of $E_0[2^\alpha]$.
- (P, Q, ι, I_P) is a precomputed basis of $E_0[q]$.

The computation of the commitment isogeny in SQIPrime2D is identical to that of the secret isogeny in SQIPrime4D, but the key generation, the response and the verification algorithms are modified.

6.2 SQIPrime2D Key Generation Algorithm

For the computation of the secret isogeny τ, whose degree is q and is public, we use **ExtKaniDoublePath**. In SQIPrime2D, the points R and S are no longer the masked images of P and Q by τ (as in SQIPrime4D). Instead, they are the masked images by $\widehat{\rho}$ of the points P and Q, where $\widehat{\rho}$ is the second isogeny computed using **ExtKaniDoublePath**. This change is necessary since $\deg(\tau) = q$, which is also the order of the points P and Q. We thus have that $\binom{R}{S} = \mathbf{M}\widehat{\rho}\binom{P}{Q}$. This time, one also includes $I_{\widehat{\rho}}$ in the secret key since it is needed when translating the kernel of the non-smooth challenge isogeny into an ideal.

Remark 2. With respect to the current state of the art [8,15,34,44] when it comes to the supersingular isogeny problem with torsion point information, there is no known algorithm that exploits the images of torsion points of non-smooth order to weaken the supersingular isogeny problem. All known attacks require the torsion point images to have smooth order. This means that the masking matrix $\mathbf{M}$ is not really necessary since q is prime. We nevertheless keep it in order to avoid having to explicitly assume that revealing the non-smooth order torsion point images in clear does not affect the security of the protocol.

6.3 SQIPrime2D Response Algorithm

Upon receiving $\mathsf{Chal} = a \in \mathbb{Z}_q$ from the verifier, the prover computes $C_a = R + [a]S = [b]\widehat{\rho}(P) + [c]\widehat{\rho}(Q)$. The prover then calculates I_{C_a} defined as $I_{C_a} = \left[\left(b + c\epsilon(\iota)\right) I_{\widehat{\rho}}\right]_* I_P$.

Algorithm 9. SQIPrime2D.KeyGen

```
Input: pp = $\big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_P)\big)$.
Output: sk = $\big(\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}}\big)$, pk = $\big(E_A, (R, S)\big)$.
$\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}} \leftarrow$ ExtKaniDoublePath$(2^\alpha, P_0, Q_0, q)$
Compute $E_A = \mathrm{Im}(\tau)$
Sample a random matrix $\mathbf{M} \in \mathrm{GL}_2(q)$
$\binom{R}{S} \leftarrow \mathbf{M}\widehat{\rho}\binom{P}{Q}$
return $(\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}}, \mathbf{M})$, $\big(E_A, (R, S)\big)$
```

Next, the prover computes the $(\mathcal{O}_2, \mathcal{O}_1)$-ideal $\overline{I_{C_a} I_\tau} I_\psi$ and locates another small $(\mathcal{O}_2, \mathcal{O}_1)$-ideal J using the **RandomEquivalentIdeal** algorithm. Following [11, Lemma 12], we are assured of the existence of such an ideal with a norm smaller than $\sqrt{p}$. Additionally, we require that $n(J)$ is odd. Notably, this condition is considerably less restrictive than that of SQIPrime4D, as approximately half of all potential isogenies remain valid, compared to only $1/\log(p)$ in the case of SQIPrime4D. Therefore, we have a high heuristic probability of finding our desired J with an odd norm d smaller than $2\sqrt{p}$, thereby yielding the corresponding isogeny $\sigma : E_2 \to E_1$. In [21, Appendix B], we provide details on how our method can be adapted to function with even d as well. The other requirement is that $I_{C_a} J$ should not be divisible by q. This is to avoid that the final response $\kappa = \sigma \circ \varphi$ is divisible by q, which would imply that κ is independent of the challenge C_a. In practice, when E_1 is sampled honestly, the probability that $I_{C_a} J$ is divisible by q is at about $q^{-2} \approx 2^{-2\lambda}$. Hence an ideal J that satisfies the previous requirements will satisfy this one as well.

With knowledge of d, the objective now shifts to constructing an auxiliary isogeny $\delta : E_A \to E_\delta$ of degree $2^\alpha - qd$. This specific mechanism lies at the heart of SQIPrime2D and underscores the necessity for the secret isogeny τ to be of degree q. The approach involves sampling $\gamma \in \mathrm{End}(E_0)$, an endomorphism of degree $d(2^\alpha - qd)$. This is done using **FullRepresentInteger**. Next, we compute $\binom{V}{W} = \tau \circ \widehat{\gamma}\binom{P_0}{Q_0}$. Given that $\deg(\tau \circ \widehat{\gamma}) = dq(2^\alpha - qd)$, we find ourselves in the following scenario:

$$\begin{tikzcd}
E_0 \arrow[dr, "\widehat{\gamma}"'] \arrow[dd, "\eta"'] & E_0' \arrow[l, "\gamma_2"'] \arrow[d, "\widehat{\gamma}_1"] \\
& E_0 \arrow[d, "\tau"] \\
E_\delta & E_A \arrow[l, "\delta"']
\end{tikzcd}$$

where $\gamma = \gamma_2 \circ \gamma_1$, $\deg(\gamma_1) = d$ and $\deg(\gamma_2) = (2^\alpha - qd)$. By applying Kani's Lemma, we construct the dimension 2 isogeny $F : E_0 \times E_A \to E_0' \times E_\delta$ of kernel $\ker(F) = \big\{\big([-qd]P, \tau \circ \widehat{\gamma}(P)\big) \big|\ P \in E_0[2^\alpha]\big\}$ and given by

$$F := \begin{pmatrix} \widehat{\gamma_2} & -\gamma_1 \circ \widehat{\tau} \\ [\gamma_2]_*(\tau \circ \widehat{\gamma_1}) & [\tau \circ \widehat{\gamma_1}]_* \gamma_2 \end{pmatrix}.$$

We thus have an efficient representation of our desired $\delta = [\tau \circ \widehat{\gamma_1}]_* \gamma_2$.

The response to our challenge is to give the evaluation T, U of $\delta \circ \widehat{\kappa} = \delta \circ \widehat{\varphi} \circ \widehat{\sigma}$ over a basis of $E_1[2^\alpha]$ to the verifier. Additionally, we share the image $V = \delta(C_a)$ of C_a through δ. To do the evaluation, we call **CanonicalTorsionBasis** over E_1 to deterministically find a basis X, Y of $E_1[2^\alpha]$, evaluate $\widehat{\kappa}$ on X and Y using the **EvalTorsion** and compute δ on these images using the dimension two isogeny F. Finally, we multiply the final points by $(-qd)^{-1} \mod 2^\alpha$. The prover then sends these three points together with the curve E_δ.

Algorithm 10. SQIPrime2D.Response

```
Input: pp = $\big(p, \alpha, q, N, (P_0, Q_0), (P, Q, \iota, I_P)\big)$, sk = $(\tau, \widehat{\rho}, I_\tau, I_{\widehat{\rho}}, \mathbf{M})$, pk = $\big(E_A, (R, S)\big)$, sec = $(\psi, I_\psi)$, com = $E_1$, chal = $a$.
Output: res = $(E_\delta, T, U, V)$ with $T, U \in E_\delta[2^\alpha]$.
 1: $\binom{b}{c} \leftarrow \mathbf{M}^\top \binom{1}{a}$
 2: $I_{C_a} \leftarrow [(b + c\iota) I_\tau]_* I_P$
 3: $J \leftarrow$ RandomEquivalentIdeal$(\overline{I_{C_a} I_\tau} I_\psi)$          $d \leftarrow n(J)$
 4: If $2|d$ or $I_{C_a} J$ is divisible by $q$, go back to step 3.
 5: $X, Y \leftarrow$ CanonicalTorsionBasis$(E_1, 2^\alpha)$
 6: $\gamma \leftarrow$ FullRepresentInteger$(\mathfrak{O}_0, d(2^\alpha - dq))$
 7: $\binom{V}{W} = \tau \circ \widehat{\gamma} \binom{P_0}{Q_0}$
 8: $\mathsf{B} \leftarrow \{([-dq]P_0, V), ([-dq]Q_0, W)\}$
 9: $F \leftarrow$ HDKernelToIsogeny$(E_0 \times E_1, \mathsf{B})$
10: Define $\tau = \mathsf{F}_\mathsf{A}(-, 0)_1$ and $\psi = \mathsf{F}_1(-, 0)_1$
11: $T_1, U_1 \leftarrow$ EvalTorsion$\Big(\mathfrak{O}_0, \tau, I_\tau, \psi, I_\psi, \overline{I_{C_1} J}, qd, \{X, Y\}\Big)$  $\triangleright T_1 = \widehat{\kappa}(X), U_1 = \widehat{\kappa}(Y)$
12: $\binom{T}{U} = [(-qd)^{-1}] \delta \binom{T_1}{U_1}$                                  $\triangleright\ \delta(-) = F(0, -)_2$
13: $V = \delta(R + [a]S)$
14: Recover $E_\delta$, the codomain of $\delta$
15: return res = $(E_\delta, T, U, V)$
```

6.4 SQIPrime2D Verification Algorithm

Note that in SQIPrime2D, the verifier receives a dimension 2 representation of $\widehat{\kappa}$ rather than that of κ. We describe how to use this representation of $\widehat{\kappa}$ to effectively check that κ is an isogeny from E_A to E_1 such that $\ker(\kappa)[q] = \langle C_a \rangle$.

Upon receipt of T, U and V, the verifier deterministically computes the basis $\langle X, Y \rangle = E_1[2^\alpha]$. Following that, the verifier uses X, Y, T and U to compute a basis for the kernel of the isogeny F, as derived from Kani's Lemma over the following diagram.

$$\begin{tikzcd}
E_A \arrow[r, "\kappa"] \arrow[d, "\delta"'] & E_1 \arrow[d, "\kappa_*\delta"] \arrow[dl, "\delta\circ\widehat{\kappa}" description] \\
E_\delta \arrow[r, "\delta_*\kappa"'] & E_\bullet
\end{tikzcd}$$

$$F : E_1 \times E_\delta \to E_A \times E_\bullet \text{ is defined as } \begin{pmatrix} \widehat{\kappa} & -\widehat{\delta} \\ \kappa_*\delta & \delta_*\kappa \end{pmatrix}$$

$$\ker(F) = \big\langle \big([-qd]X, \delta\circ\widehat{\kappa}(X)\big), \big([-qd]Y, \delta\circ\widehat{\kappa}(Y)\big) \big\rangle = \big\langle \big(X, T\big), \big(Y, U\big) \big\rangle$$

Using F, they compute the point $F\binom{0}{V} = \binom{-\widehat{\delta}(V)}{\delta_*\kappa(V)}$ and check that:

1. $\delta_*\kappa(V) = 0$.
2. $\widehat{\delta}(V) = [2^\alpha - qd](R + [a]S) = [2^\alpha](R + [a]S)$.

Additionally, we check that for $W \in E_\delta[q]$ linearly independent with V, $\delta_*\kappa(W) \neq 0$. This ensures that $\ker(\kappa)[q] = \langle C_a \rangle$.

Algorithm 11. SQIPrime2D.Verify

Input: $\mathsf{pp} = \big(p, (P_0, Q_0), (P, Q, \iota, I_P)\big), \mathsf{pk} = (E_A, R, S), \mathsf{com} = (E_1), \mathsf{chal} = a, \mathsf{res} = (E_\delta, T, U, V)$.
Output: 0 or 1.

```
Check $T, U, V \in E_\delta$
$X, Y \leftarrow$ **CanonicalTorsionBasis**$(E_1, 2^\alpha)$
$\mathsf{B} \leftarrow \{(X, T), (Y, U)\}$
$F \leftarrow$ **HDKernelToIsogeny**$(E_1 \times E_\delta, \mathsf{B})$        ▷ If not well defined, return 0
**if** codomain $\widehat{\kappa} \neq E_A$ **do return** 0
Sample $W \in E_\delta[q]$ such that $V$ and $W$ are linearly independent
$\binom{Z_1}{Z_2} \leftarrow F\binom{0}{V} = \binom{-\widehat{\delta}(V)}{\delta_*\kappa(V)}$
$b_1 \leftarrow Z_1 \overset{?}{=} [2^\alpha](R + [a]S)$
$b_2 \leftarrow Z_2 \overset{?}{=} 0$
$b_3 \leftarrow \delta_*\kappa(W) \overset{?}{\neq} 0$
**return** $b_1 \wedge b_2 \wedge b_3$
```

The following proposition shows us that our verification is correct.

Proposition 4. *Let* $\mathsf{pp}, \mathsf{pk}, \mathsf{com}, \mathsf{chal}$ *be the public parameters, a valid public key, a commitment, and a challenge in SQIPrime2D and let* X, Y *be the canonical basis of* $E_1[2^\alpha]$*. Let* $\overline{\mathsf{Res}} = \overline{\mathsf{Res}} = (\overline{E_\delta}, \overline{T}, \overline{U}, \overline{V})$ *be any possible output of Algorithm 10.*
If ***SQIPrime2D.Verify***$(\mathsf{pp}, \mathsf{pk}, \mathsf{com}, \mathsf{chal}, \overline{\mathsf{Res}}) = 1$, *then* $(X, Y, \overline{T}, \overline{U})$ *is a dim 2 representation of an isogeny* $\widehat{\overline{\kappa}} : E_1 \to E_A$ *of degree* $q\overline{d} < 2^\alpha$ *and such that* $\overline{\kappa}$ *factors through* φ*, the isogeny corresponding to the challenge* chal*, but is not divisible by* q*; in other words,* $\ker(\overline{\kappa})[q] = \langle C_a \rangle$*.*

Proof. Let $\overline{E_\delta}, \overline{T}, \overline{U}, \overline{V}$ be an accepting response. Since the $(2^\alpha, 2^\alpha)$ isogeny $\overline{F}$ whose kernel is generated by $\{(X, \overline{T}), (Y, \overline{U})\}$ is well-defined, then $\deg(\overline{F}_{1,1}) = \deg(\overline{F}_{2,2})$, $\deg(\overline{F}_{1,2}) = \deg(\overline{F}_{2,1})$ and $\deg(\overline{F}_{1,1}) + \deg(\overline{F}_{1,2}) = 2^\alpha$.

Thus, as $\overline{F}_{2,2}(\overline{V}) = 0$, we know that q divides $\deg(\overline{F}_{2,2})$, meaning that it cannot divide $\deg(\overline{F}_{1,2})$. Since $\overline{F}_{1,2}(\overline{V}) = [2^\alpha](R + [a]S)$, then $[2^\alpha]\widehat{\overline{F}_{1,2}}(R + [a]S) = [\deg(\overline{F}_{1,2})]\overline{V}$. As q and $2^\alpha \deg(\overline{F}_{1,2})$ are co-prime, we have that $\overline{F}_{2,2} \circ \widehat{\overline{F}_{1,2}}(R + [a]S) = 0 = \overline{F}_{2,1} \circ \widehat{\overline{F}_{1,1}}(R + [a]S)$. As $\deg(\overline{F}_{1,2}) = \deg(\overline{F}_{2,1})$ is not divisible by q, then $\widehat{\overline{F}_{1,1}}(R + [a]S) = 0$. We therefore have that $\widehat{\overline{F}_{1,1}} : E_A \to E_1$ is of degree $q\overline{d} < 2^\alpha$ and it factors through the isogeny φ corresponding to the challenge chal. Since $W \in E_\delta[q]$ is such that $\overline{V}$ and W are linearly independent, then $\overline{F}_{2,2}(W) \neq 0$ induces that $\overline{F}_{2,2}$ and $\widehat{\overline{F}_{1,1}} : E_A \to E_1$ are not divisible by q. □

7 Security Analysis of SQIPrime2D

Similarly to SQIPrime4D, we have to show that SQIPrime2D defines a Σ protocol. We thus have to prove that we have special soundness and are Honest Verifier Zero Knowledge. Our proof of special soundness differs slightly from Proposition 2, as it leverages the primality of q to address the case where $\deg(\sigma)$ is not necessarily co-prime to q.

Proposition 5. *Let $(E_1, \mathsf{chal}_1, T_1, U_1, V_1)$ and $(E_1, \mathsf{chal}_2, T_2, U_2, V_2)$ be 2 transcripts of SQIPrime2D with identical commitment E_1 and $\mathsf{chal}_1 \neq \mathsf{chal}_2$. There exists an extractor $\mathcal{E}$ that, given both transcripts, can efficiently solve the one endomorphism problem (Problem 1) over E_A, i.e. find $\theta_A \in End(E_A)$ a non-trivial endomorphism.*

Proof. Similarly to Proposition 2, we construct $\theta_A = \widehat{\kappa_2} \circ \kappa_1 \in \mathrm{End}(E_A)$. We now show that θ_A is non-scalar.

Let $d_1 = \deg(\sigma_1)$ and $d_2 = \deg(\sigma_2)$. Recall that $d_1 q < 2^\alpha < p$, $d_2 q < 2^\alpha < p$ and q is prime. If both are co-prime to q, then one follows the same reasoning as in the proof of Proposition 2. Let us assume that q divides d_1 and let $d_1 = d_1' q$. Then d_1' is co-prime to q, as otherwise, we would have $d_1 q = d_1'' q^3 \geq q^3 > 2^\alpha \geq d_1 q$ where $d_1' = d_1'' q$, leading to a contradiction.

Now, suppose $\theta_A = [\chi]$. Since $\chi^2 = \deg[\chi] = \deg \theta_A = q^2 d_1 d_2 = q^3 d_1' d_2$, then $d_2 = d_2' q$ with d_2' co-prime to q. Hence $\deg \kappa_1 = q^2 d_1'$ and $\deg \kappa_2 = q^2 d_2'$ where d_1' and d_2' are co-prime with q. Write $\kappa_1 = \phi_1 \circ \kappa_1'$ and $\kappa_2 = \phi_2 \circ \kappa_2'$ where the isogenies κ_1', κ_2', ϕ_1 and ϕ_2 have degree q^2, q^2, d_1' and d_2' respectively. Since $\theta_A = \widehat{\kappa_2} \circ \kappa_1$ is a scalar endomorphism and, κ_1 and κ_2 are not divisible by q (which is prime), then $\kappa_1' = \kappa_2'$. This implies that $\ker(\varphi_1) := \ker(\kappa_1) \cap E_A[q] = \ker(\kappa_1') \cap E_A[q] = \ker(\kappa_2') \cap E_A[q] = \ker(\kappa_2) \cap E_A[q] =: \ker(\varphi_2)$, i.e. $\mathsf{chal}_1 = \mathsf{chal}_2$, which is a contradiction. □

Regarding HVZK, there are several differences between SQIPrime4D and SQIPrime2D:

1. We have access to an auxiliary isogeny $\delta : E_A \to E_\delta$.
2. Our isogeny κ is of degree qd where the requirements that d is co-prime to q and $2^\beta - qd$ is prime congruent to 1 modulo 4 are relaxed.

We therefore need to define our HVZK under new oracles, defined as such.

Definition 3. *The* Random Uniform Constrained Odd Degree Isogeny Oracle *(RUCODIO) is an oracle that takes as input a supersingular curve E together with $P \in E[q]$ and returns an efficient representation of an isogeny $\kappa : E \to E'$ of degree $q\ell$ such that:*

- *E' is uniformly distributed.*
- *κ is uniformly distributed among all isogenies between E and E' such that:*
    - *ℓ is odd with $q\ell \leq 2^\alpha$.*
    - *κ is such that $\kappa(P) = 0$.*

Definition 4. *The* Auxiliary Isogeny Oracle *(AIO) is an oracle that takes as input a supersingular curve E together with an odd integer $\ell < 2^\alpha/q$ and returns an efficient representation of an isogeny $\delta : E \to E''$ of degree $2^\alpha - q\ell$ such that it has the same distribution as the auxiliary isogeny computed in Algorithm 10.*

Using RUCODIO and AIO, we can now prove our HVZK.

Proposition 6. *Given* pp, pk *and* chal*, then there exists a simulator $\mathcal{S}$ with access to a RUCODIO and AIO that simulates transcripts with a distribution that is computationally indistinguishable from the distribution of transcripts of SQIPrime2D, conditioned to* chal.

Proof. Given E_A, we sample $a \in \mathbb{Z}_q$ and construct $C = R + [a]S$ call RUCODIO over E_A and C, we retrieve an efficient representation of $\kappa : E_A \to E_1$. We compute $\ell = \deg(\kappa)/q$ and call AIO over E_A and d to retrieve $\delta : E_A \to E_\delta$. We use this representation to compute the points $T = \delta \circ \widehat{\kappa}(X), U = \delta \circ \widehat{\kappa}(Y)$ and $V = \delta(C)$ with X, Y the canonical basis over $E_1[2^\alpha]$. We then simply return the following transcript $(E_1, a, E_\delta, T, U, V)$.

This transcript is computationally indistinguishable from a genuine transcript, as:

- A genuine E_1 or one given by RUCODIO are computationally indistinguishable, following Assumption 1.
- Due to Definition 4, E_δ has the same distribution as the isogeny computed during SQIPrime2D response. This also applies to the point V.
- Following [32, Lemma 3.2.4], a genuine κ or one given by RUCODIO are computationally indistinguishable, and so does T, U. □

We now make the following assumption.

Assumption 4. *The one endomorphism problem (Problem 1) remains hard even when given access to RUCODIO and AIO.*

Thus, we have that, under our assumptions, SQIPrime2D is a Σ-protocol.

8 Parameters and Efficiency

As discussed in Sect. 4 and Sect. 6, the public parameters in both versions of SQIPrime differ significantly from those used in SQISign [18,19] and SQISignHD [11], particularly concerning their base prime numbers. This section provides a detailed explanation on how to compute suitable baseline "SQIPrime-friendly" primes.

8.1 Finding "SQIPrime4D-Friendly" Primes

We can view "SQIPrime4D-friendly" primes as a combination of the "SIDH primes" used in SQISignHD and the stringent requirements on both $p + 1$ and $p - 1$ seen in SQISign primes. However, in SQIPrime4D, the only condition is that $p - 1$ needs to have a factor of size $O(2^\lambda)$. Finding "SQIPrime4D-friendly" primes is actually easier than finding "SQISign-friendly" primes. These primes can in fact be found by a brute-force search over the cofactor f. Her are some good candidates.

$$\begin{aligned}\lambda = 128: \; p + 1 &= 2^{241} \cdot 33967 \simeq 2^{256} \\ q &= 64713388935233039174428822937611397577 7 \simeq 2^{128}\end{aligned}$$

$$\begin{aligned}\lambda = 192: \; p + 1 &= 2^{368} \cdot 239 \cdot 277 \simeq 2^{384} \\ q &= 3 \cdot 7 \cdot 4803463386334137403 \cdot \\ &\quad 11668209688687890994588820213524387306 1 \simeq 2^{193}\end{aligned}$$

$$\begin{aligned}\lambda = 256: \; p + 1 &= 2^{497} \cdot 5^2 \cdot 479 \simeq 2^{512} \\ q &= 97 \cdot 147869462015622684206054234380684709202350 \\ &\quad 141554573643051528098693560900067 7 \simeq 2^{256}\end{aligned}$$

8.2 Finding "SQIPrime2D-Friendly" Primes

Finding "SQIPrime2D-friendly" primes through a brute-force search over the cofactor f as we did in the case of SQIPrime4D is computationally involved. This essentially comes from the fact that we want $q \approx 2^\lambda$ to be prime this time and if we take $p = 2^\alpha f - 1$ to be a prime, then the probability that a random prime q divides $p - 1$ is roughly $1/q$. Given that there are approximately $2^\lambda(2^t - 1)/\lambda$ distinct primes in the interval $[2^\lambda, 2^{\lambda+t}]$, the probability that there exists a prime q in $[2^\lambda, 2^{\lambda+t}]$ that divides $p - 1$ is heuristically given by:

$$\mathbb{P}\left[\exists q \in \left[2^\lambda, 2^{\lambda+t}\right] \text{ such that } q \middle| (p-1)\right] \geq \sum_{q \geq 2^\lambda}^{2^{\lambda+t}} \mathbb{P}\left[q | (p-1)\right] \simeq \sum_{q \geq 2^\lambda}^{2^{\lambda+t}} \frac{1}{q}$$

$$\geq \sum_{i=1}^{t} \sum_{q \geq 2^{\lambda+i-1}}^{2^{\lambda+i}} \frac{1}{2^{\lambda+i}} \simeq \sum_{i=1}^{t} \frac{2^{\lambda+i}}{(\lambda+i)} \frac{1}{2^{\lambda+i}} \simeq \sum_{i=1}^{t} \frac{1}{\lambda+i} \geq \frac{t}{\lambda+t}$$

Following this computation, the probability that, for a given f, $p = 2^{\alpha} f - 1$ is prime *and* $p-1$ has a factor close to λ-bit long is about $O(1/\lambda^2)$. This induces that the expected size of f is around $2\log_2(\lambda)$, meaning that a “SQIPrime2D-friendly” prime for the security level λ is of expected size $2\lambda + 4\log_2(\lambda)$ bits, or a little bit larger. These additional $4\log_2(\lambda)$ bits present a challenge. For $\lambda = 128$, this results in an overhead of approximately 28 bits, which translates to an 11% increase in the size of the base prime p.[6]

For $\lambda = 128$, the first “SQIPrime2D-friendly” prime we identified is denoted as p_{130}.

$$\begin{aligned} p_{130} &= 2^{273} \cdot 19^2 - 1 \simeq 2^{281.50} \\ q_{130} &= 173312401330203632071817182256347704766 7 \simeq 2^{130.35} \end{aligned}$$

To find a smaller p, it is tempting to ask for q to be non-prime, as we did for SQIPrime4D, but this is not possible as our security would be downgraded by a non-smooth generalisation of the Galbraith meet-in-the-middle attack [26] and our proof of special soundness would be affected. However, to maintain efficiency, we may tolerate a slight reduction in the bit length of q. We suggest the following prime.

$$\begin{aligned} p_{117} &= 2^{247} \cdot 79 - 1 \simeq 2^{253.34} \\ q_{117} &= 168118140144706967996895604212334429 \simeq 2^{117.01} \end{aligned}$$

Searching for “SQIPrime2D-friendly” primes corresponding to security levels $\lambda = 192, 256$, by brute-force search over the cofactor f, is practically out of reach since it requires factoring several numbers of about 384 and 512 bits. We therefore use a more advanced method which consists of sample integers $p = 2x^2 - 1$ where $x = 2^r f_0$ with $r > \lambda$ and f_0 being a small integer. When p is prime, then since $p - 1 = 2x^2 - 2 = 2(x-1)(x+1)$, we only need to check whether $x+1$ or $x-1$ has a prime $q \approx 2^{\lambda}$. Interestingly, this essentially reduces to checking whether $x+1$ or $x-1$ has a small smooth factor s in the order of $x/2^{\lambda}$ such that $(x-1)/s$ or $(x+1)/s$ is prime. This hence leads to a quite efficient method to generate “SQIPrime2D-friendly” primes. A similar technique [7] was also used in the context of SQISign for the parameter generation. We obtained the following primes for the security levels $\lambda = 192, 256$ respectively.

$$\begin{array}{l|l} p_{186} = 2^{397} \cdot 3^2 \cdot 7^2 \cdot 11^2 - 1 \simeq 2^{413} & p_{240} = 2^{499} \cdot 3^2 \cdot 7^2 - 1 \simeq 2^{508} \\ q_{186} = (2^{198} \cdot 3 \cdot 7 \cdot 11 - 1)/664723 \simeq 2^{187} & q_{240} = (2^{249} \cdot 3 \cdot 7 - 1)/7709 \simeq 2^{241} \end{array}$$

[6] It is important to note that this overhead scales logarithmically with λ. As λ doubles, the overhead only increases by 4 bits, meaning that its relative cost decreases at higher security levels.

8.3 Compactness of SQIPrime

Similarly to SQISign and SQISignHD, both version of SQIPrime are made into digital signature schemes via the Fiat-Shamir transform [23]. Thoses digital schemes are universally unforgeable under chosen message attacks (UU-CMA) in the random oracle and RUCGDIO or RUCODIO+AIO model, assuming the hardness of the one endomorphism problem.

Signature Size. In the case of SQIPrime2D, the signature takes the form $\mathsf{sign} = \big(E_1, E_\delta, T, U, V\big)$. This signature can be slightly compressed using methods akin to those outlined in [11, Section 6.1]. The crux of this compression lies in representing T and U by $a_1, a_2, a_3, \in \mathbb{Z}_{2^\alpha}$ corresponding to their coordinates in a deterministic basis of $E_\delta[2^\alpha]$, with the final coordinate a_4 derived using pairings and discrete logs and using d an integer of λ bits. Employing this compression method, each component of a SQIPrime2D signature exhibits the following sizes:

- E_1 and E_δ are represented by their j-invariant in $\mathbb{F}_{p^2}$, hence of size $8\lambda + O(\log \lambda)$.
- T and U are each represented by three integers of size α plus d of size $\log(p)/2$, totaling $7\lambda + O(\log \lambda)$ bits.
- Finally, because q is non-smooth, we can not compress V, meaning that they are represented as a point in $\mathbb{F}_{p^2}$, hence of size $4\lambda + O(\log \lambda)$.

Summing these sizes, a SQIPrime2D signature is $19\lambda + O(\log \lambda)$ bits long. Consequently, it is larger than the signature of SQISignHD, which was $13/2\lambda + O(\log \lambda)$ bits, and also larger than SQISign, which is at least $17/2\lambda + O(\log \lambda)$ bits. Nevertheless, it remains a highly compact post-quantum signature scheme.

It is noteworthy that similar compression techniques can be applied to the SQIPrime4D signature, which is of the form (E_1, T, U, V, d), resulting in a signature size of $12\lambda + O(\log \lambda)$ bits. This difference of 7λ bits comes from the fact that in SQIPrime2D, we have to share E_δ and because in SQIPrime4D, we split the verification in 2 dimension 4 isogenies, therefore only requiring 2^β-torsion points, as opposed to 2^α in SQIprime2D.

8.4 SQIPrime Efficiency

The next phase for SQIPrime involves developing an efficient implementation of SQIPrime2D. Building upon the advancements made in [12], as well as leveraging the efficient implementations of SQISign [18,19] and SQISignHD [11], we anticipate that SQIPrime2D will demonstrate very competitive performance. As for now, we have a proof of concept code written in SageMath, with which the key generation, signature and verification take about 720ms, 1750ms and 200ms respectively when using p_{117}. We will provide a link to the code in the full version [21] of the paper once the code is ready to share, as there is still room for improvement through basic optimisations.

Table 1. Size (in bytes) comparison between the different SQI-protocols for public keys and signatures in both normal and compressed form.

Scheme	λ	pk	signature	signature (compressed)
SQISign	128	64	322	177
	192	92	-	267
	256	128	-	335
SQISignHD	128	64	208	109
	192	92	312	156
	256	128	416	208
SQIPrime4D	128	192	272	240
	192	288	408	288
	256	384	544	384
SQIPrime2D	128	191	320	299
	192	288	517	484
	256	384	635	600

With and advanced implementation, we expect signature time in SQIPrime2D to be relatively slower compared to that of SQISignHD (this is due to the extra cost of generating the auxiliary isogeny in SQIPrime2D when computing the signature), while having a faster verification. This intuition follows from the number of $(2, 2)$ isogenies required to perform SQIPrime2D, as detailed in [21, Appendix C].

9 Conclusion

In this paper, we have designed SQIPrime, an elegant variant of SQISignHD that uses non-smooth degree challenge isogenies. Moreover, we have described a variant, SQIPrime2D, that uses only dimension two isogenies.

We provide a theoretical performance analysis of our schemes and anticipate SQIPrime2D being more efficient compared to SQISignHD. An effective implementation of SQIPrime2D, which should be expected in the near future, will allow us to have a more practical comparison between SQIPrime4D and SQISignHD on one hand, and, SQIPrime2D, SQISign2D-West [2] and SQISign2D-East [37] on the other hand.

Acknowledgments. We are grateful to the anonymous ASIACRYPT 2024 reviewers for their useful comments that helped improve this paper. We would like to thank the authors of SQISign2D-West and the authors of SQISign2D-East for useful discussions regarding this topic. We would also like to thank Michael Meyer, Lorenz Panny and Bruno Sterner for describing a fast method to generate the primes p used in SQIPrime2D.

References

1. Basso, A.: POKE: A framework for efficient PKEs, split KEMs, and OPRFs from higher-dimensional isogenies. Cryptology ePrint Archive, Paper 2024/624 (2024), https://eprint.iacr.org/2024/624
2. 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
3. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 98–126. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8739-9_4
4. Bernstein, D.J., De Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. Open Book Series **4**(1), 39–55 (2020)
5. Bernstein, D.J., Hamburg, M., Krasnova, A., Lange, T.: Elligator: elliptic-curve points indistinguishable from uniform random strings. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013. pp. 967–980. ACM Press (Nov 2013).https://doi.org/10.1145/2508859.2516734
6. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 227–247. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_9
7. Bruno, G., Santos, M.C.R., Costello, C., Eriksen, J.K., Meyer, M., Naehrig, M., Sterner, B.: Cryptographic smooth neighbors. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 190–221. Springer, Singapore (Dec 2023).https://doi.org/10.1007/978-981-99-8739-9_7
8. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Cham (Apr 2023).https://doi.org/10.1007/978-3-031-30589-4_15
9. 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, Cham (Apr 2024).https://doi.org/10.1007/978-3-031-57725-3_7
10. Costello, C.: B-SIDH: Supersingular isogeny Diffie-Hellman using twisted torsion. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 440–463. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64834-3_15
11. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: New dimensions in cryptography. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 3–32. Springer, Cham (May 2024).https://doi.org/10.1007/978-3-031-58716-0_1
12. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to $(2,2)$-isogenies in the theta model and applications to isogeny-based cryptography. Cryptology ePrint Archive, Paper 2023/1747 (2023), https://eprint.iacr.org/2023/1747
13. De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017)
14. De Feo, L., Delpech de Saint Guilhem, C., Fouotsa, T.B., Kutas, P., Leroux, A., Petit, C., Silva, J., Wesolowski, B.: Séta: Supersingular encryption from torsion attacks. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 249–278. Springer, Cham (Dec 2021).https://doi.org/10.1007/978-3-030-92068-5_9

15. De Feo, L., Fouotsa, T.B., Panny, L.: Isogeny problems with level structure. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VII. LNCS, vol. 14657, pp. 181–204. Springer, Cham (May 2024).https://doi.org/10.1007/978-3-031-58754-2_7
16. De Feo, L., Galbraith, S.D.: SeaSign: Compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 759–789. Springer, Cham (May 2019). https://doi.org/10.1007/978-3-030-17659-4_26
17. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology **8**(3), 209–247 (2014)
18. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_3
19. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 659–690. Springer, Cham (Apr 2023).https://doi.org/10.1007/978-3-031-30589-4_23
20. Deuring, M.: Die typen der multiplikatorenringe elliptischer funktionenkörper. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg. vol. 14, pp. 197–272. Springer Berlin/Heidelberg (1941)
21. Duparc, M., Fouotsa, T.B.: SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. Cryptology ePrint Archive, Paper 2024/773 (2024), https://eprint.iacr.org/2024/773
22. Duparc, M., Fouotsa, T.B., Vaudenay, S.: SILBE: an Updatable Public Key Encryption Scheme from Lollipop Attacks. Cryptology ePrint Archive, Paper 2024/400 (2024), https://eprint.iacr.org/2024/400
23. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Berlin, Heidelberg (Aug 1987).https://doi.org/10.1007/3-540-47721-7_12
24. Fouotsa, T.B.: A note on the prime in SQISignHD. Online (2024), https://github.com/BorisFouotsa/BorisFouotsa.github.io/blob/main/files/A_note_on_the_prime_in_SQISignHD.pdf
25. Fouotsa, T.B., Petit, C.: SHealS and HealS: Isogeny-based PKEs from a key validation method for SIDH. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 279–307. Springer, Cham (Dec 2021).https://doi.org/10.1007/978-3-030-92068-5_10
26. Galbraith, S.D.: Constructing isogenies between elliptic curves over finite fields. LMS Journal of Computation and Mathematics **2**, 118–138 (1999)
27. 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, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_1
28. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4. pp. 19–34. Springer (2011). https://doi.org/10.1007/978-3-642-25405-5

29. Kani, E.: The number of curves of genus two with elliptic differentials. Journal für die reine und angewandte Mathematik **1997**(485), 93–122 (1997).https://doi.org/10.1515/1997.485.932
30. 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)
31. Kunzweiler, S.: Efficient computation of $(2^n, 2^n)$-isogenies. Designs, Codes and Cryptography **92**(6), 1761–1802 (2024)
32. Leroux, A.: Quaternion Algebra and Isogeny-Based Cryptography. Ph.D. thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
33. 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
34. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 448–471. Springer, Cham (Apr 2023).https://doi.org/10.1007/978-3-031-30589-4_16
35. Moriya, T.: IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram. Cryptology ePrint Archive, Paper 2023/1506 (2023), https://eprint.iacr.org/2023/1506
36. Nakagawa, K., Onuki, H.: QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 75–106. Springer, Cham (Aug 2024).https://doi.org/10.1007/978-3-031-68388-6_4
37. 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
38. NIST: Post-Quantum Cryptography: Digital Signature Schemes, https://csrc.nist.gov/projects/pqc-dig-sig/standardization
39. Onuki, H., Nakagawa, K.: Ideal-to-isogeny algorithm using 2-dimensional isogenies and its application to SQIsign. Cryptology ePrint Archive, Paper 2024/778 (2024), https://eprint.iacr.org/2024/778
40. Page, A., Robert, D.: Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Paper 2023/1766 (2023), https://eprint.iacr.org/2023/1766
41. Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VI. LNCS, vol. 14656, pp. 388–417. Springer, Cham (May 2024).https://doi.org/10.1007/978-3-031-58751-1_14
42. Robert, D.: Fonctions thêta et applications à la cryptographie. Ph.D. thesis, Université Henri Poincaré-Nancy I (2010)
43. Robert, D.: Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive, Paper 2022/1068 (2022), https://eprint.iacr.org/2022/1068
44. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Cham (Apr 2023).https://doi.org/10.1007/978-3-031-30589-4_17
45. Santos, M.C.R., Costello, C., Smith, B.: Efficient (3,3)-isogenies on fast Kummer surfaces. Cryptology ePrint Archive, Paper 2024/144 (2024), https://eprint.iacr.org/2024/144
46. Santos, M.C.R., Eriksen, J.K., Meyer, M., Reijnders, K.: AprèsSQI: Extra fast verification for SQIsign using extension-field signing. In: Joye, M., Leander, G.

(eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 63–93. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_3

47. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
48. Vélu, J.: Isogénies entre courbes elliptiques. Comptes-Rendus de l'Académie des Sciences **273**, 238–241 (1971)
49. 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, Cham (Apr 2017). https://doi.org/10.1007/978-3-319-70972-7_9