SIDH Proof of Knowledge

Luca De Feo[1], Samuel Dobson[2], Steven D. Galbraith[2], and Lukas Zobernig[2]

[1]IBM Research Europe. luca@defeo.lu
[2]Mathematics Department, University of Auckland, New Zealand.
samuel.dobson.nz@gmail.com, s.galbraith@auckland.ac.nz,
lukas.zobernig@auckland.ac.nz

June 29, 2022

Abstract

We show that the soundness proof for the De Feo–Jao–Plût identification scheme (the basis for supersingular isogeny Diffie–Hellman (SIDH) signatures) contains an invalid assumption, and we provide a counterexample for this assumption—thus showing the proof of soundness is invalid. As this proof was repeated in a number of works by various authors, multiple pieces of literature are affected by this result. Due to the importance of being able to prove knowledge of an SIDH key (for example, to prevent adaptive attacks), soundness is a vital property.

Surprisingly, the problem of proving correctness of an isogeny turns out to be considerably more difficult than was perhaps anticipated. The main result of this paper is a sigma protocol to prove that an SIDH public key (including the torsion points in the public key) is correctly formed. Our scheme also avoids the SIDH identification scheme soundness issue raised by Ghantous, Pintore and Veroni. In particular, our protocol provides a non-interactive way of verifying that SIDH public keys are well-formed as protection against adaptive attacks, leading to an SIDH-based non-interactive key exchange (NIKE).

1 Introduction

While Supersingular Isogeny Diffie-Hellman (SIDH) [JD11, DJP14] is a fast and efficient post-quantum key exchange candidate, it has been hampered by the existence of practical adaptive attacks on the scheme—the first of these given by Galbraith et al. [GPST16] (the GPST attack), followed by other variations [FP21, UXT$^+$22]. These attacks mean it is not safe to re-use a static key across multiple SIDH exchanges without other forms of protection. As such, various countermeasures have been proposed—though each with its unique drawbacks.

The first of these is to require one participant to use a one-time ephemeral key in the exchange, accompanied by a Fujisaki–Okamoto-type transform [HHK17] revealing the corresponding secret to the other party. This allows the recipient to verify the public key is well-formed, ensuring an adaptive attack was not used. This is what was done in SIKE [ACC$^+$17], and converts the scheme to a secure key encapsulation mechanism (KEM). But it is of limited use in cases where both parties wish to use a long-term key.

The second countermeasure is to use many SIDH exchanges in parallel, combining all the resulting secrets into a single value, as proposed by Azarderakhsh, Jao, and Leonardi [AJL17]. This scheme is known as k-SIDH, where k is the number of keys used by each party in the exchange. The authors suggest $k = 92$ is required for a secure key exchange. Dobson et al. [DGL$^+$20] demonstrate how the GPST adaptive attack can be ported to $k = 2$ and above. Note that the number of SIDH instances grows as k^2, so this scheme is very inefficient. Urbanik and Jao's [UJ20] proposal attempted to improve the efficiency of this protocol by making use of the special automorphisms on curves with j-invariant 0 or 1728, but it was shown by Basso et al. [BKM$^+$20] that Urbanik and Jao's proposal is vulnerable to a more efficient adaptive attack and actually scales worse in efficiency than k-SIDH itself (although the public keys are approximately $4/5$ of the size, it requires around twice as many SIDH instances for the same security).

Finally, adaptive attacks can also be prevented by providing a non-interactive proof that a public key is well-formed or honestly generated. Generic NIZK techniques would make this possible, but in a very inefficient manner. Urbanik and Jao [UJ20] claim a method for doing so using a similar idea to their k-SIDH improvement mentioned above. Their scheme is based on the SIDH-based identification scheme by De Feo, Jao, and Plût [DJP14], which is a fairly simple proof with single bit challenges.

We briefly recall the De Feo, Jao, and Plût proof here, for full details see Section 4.1. Let $\phi : E_0 \to E_1$ be the isogeny of degree $\ell_1^{e_1}$ we wish to prove knowledge of. Let P_0, Q_0 be a basis of the torsion subgroup $E_0[\ell_2^{e_2}]$, and let $(P_1, Q_1) = (\phi(P_0), \phi(Q_0))$. The prover chooses a pair of integers (a, b), and sends to the verifier $E_2 = E_0/\langle [a]P_0 + [b]Q_0\rangle$ and $E_3 = E_1/\langle [a]P_1 + [b]Q_1\rangle$. The verifier sends a single bit challenge chall. When $\mathsf{chall} = 0$ the prover responds with (a, b), and when $\mathsf{chall} = 1$ the prover responds with an isogeny $\phi' : E_2 \to E_3$ of degree $\ell_1^{e_1}$. The protocol is repeated until the verifier is satisfied.

We show a counterexample to the soundness of the original De Feo–Jao–Plût scheme. Because this scheme (and proof) has since been used to build an undeniable signature by Jao and Soukharev [JS14], a signature scheme by Yoo, Azarderakhsh, Jalali, Jao, and Soukharev [YAJ$^+$17], and also by Galbraith, Petit, and Silva [GPS20], all of these subsequent papers suffer from the same issue. Our counterexample does not immediately apply to Urbanik and Jao's scheme, but we show other problems with that scheme in Section 4.4.

Ghantous, Pintore, and Veroni [GPV21] have demonstrated that the soundness property for the De Feo–Jao–Plût scheme (and those based on it) fails for a different reason—namely the existence of multiple isogenies of the same degree between some curves. The protocols we propose in this paper are not vulnerable to the same issue, as we briefly discuss in Remark 2.

We stress that the flaw in the De Feo–Jao–Plût soundness argument does not mean that previous isogeny signature schemes [YAJ$^+$17, GPS20] are insecure. Forgery for these schemes still requires an attacker to compute an isogeny between two given elliptic curves, which is a hard problem.

1.1 Our contributions

We present three new sigma protocols for SIDH. They all prove, for a pair (E_0, E_1) of publicly known supersingular curves, knowledge of an isogeny $\phi : E_0 \to E_1$ of the correct degree (the private key or *witness*). But they have some key differences we summarize next.

First, in Section 5.1, we propose a modification to the De Feo–Jao–Plût scheme that ensures that there is an extractor for the witness $\phi : E_0 \to E_1$. The first key idea in this protocol is the provision of bases (P_2, Q_2) for $E_2[\ell_2^{e_2}]$ and (P_3, Q_3) for $E_3[\ell_2^{e_2}]$. This allows the verifier to check that $(P_3, Q_3) = (\phi'(P_2), \phi'(Q_2))$ in the $\mathsf{chall} = 1$ case, and in the $\mathsf{chall} = 0$ case, to check that the isogenies from E_2 to E_0 and E_3 to E_1 are "parallel". The second key idea is, in the 2-special soundness proof, to view the transcript as an SIDH square where E_2 is treated as the "base curve" (instead of E_0), and where E_0 and E_3 play the roles of the participants' two public-key curves in SIDH. It then follows that there is a witness ϕ as required.

This protocol is simple, and sound, but there is a minor problem with zero-knowledge: in the $\mathsf{chall} = 0$ case, contrary to the De Feo–Jao–Plût scheme, the data $(E_2, P_2, Q_2, E_3, P_3, Q_3)$ appears to be difficult to simulate without knowledge of the secret witness. We solve this issue in Section 5.3 by moving from binary to ternary challenges, thus making the protocol 3-special sound: the $\mathsf{chall} = 0$ case is split into two different challenges, so that only one of (E_2, P_2, Q_2) or (E_3, P_3, Q_3) needs to be revealed at a time. Plugging a statistically hiding commitment scheme in, we obtain a zero-knowledge proof of knowledge for what we dub the *weak SIDH relation*, i.e. the existence of $\phi : E_0 \to E_1$ of degree $\ell_1^{e_1}$. To the best of our knowledge, this is the first zero-knowledge and sound protocol for such relation.

Finally, in Section 6, we give a new sigma protocol that convinces a verifier not only that there is an isogeny $\phi : E_0 \to E_1$ of the correct degree, but also that the torsion points provided in an SIDH public key are the correct images of the public parameter points under ϕ. We call this stronger relation the *SIDH relation*. Making this non-interactive using the Fiat-Shamir heuristic gives a secure method for proving well-formedness of SIDH public keys, which is needed if one wants to prevent adaptive attacks. This is the first such protocol in the literature and has important applications in all areas where SIDH key exchange could be used with static keys. Our scheme works with any base elliptic curve, rather than being restricted to the two curves with j-invariant 0 or 1728 as in [UJ20].

The scheme in Section 6 builds on the protocols of Section 5. However, it requires assurance that the ephemeral isogenies used in the commitments by the prover are "independent enough". To achieve this, we "double" the protocol, by essentially running two sessions of the protocol from Section 5.3 for each challenge bit. The prover shows that the two instances are consistent with each other by providing images of a random torsion basis in both squares, which the verifier can check are correct. The verifier also checks that the two instances are independent. This allows us to construct an extractor that outputs a correct witness.

Because the lasts two protocols are 3-special sound, the probability of successful cheating is $2/3$, indeed a forger who does not know the witness can simultaneously construct valid responses to any two challenges. This would have implications on tightness if they were used for signature schemes. We do not recommend our protocols as bases for signatures.

Commitments in the original De Feo–Jao–Plût scheme were just j-invariants of curves, but our new proofs require committing to various points on curves as well. This makes the proofs considerably larger. As with the original De Feo–Jao–Plût scheme, it is non-trivial in the $\mathsf{chall} = 1$ case to simulate valid protocol transcripts without knowing the witness and so we only achieve computational zero-knowledge.

We explain in Section 7 that our scheme gives an asymptotically more efficient non-interactive key exchange (NIKE) than the k-SIDH proposal by Azarderakhsh, Jao and Leonardi [AJL17]. But we stress that NIKE is not the only application of our work.

1.2 Plan of the paper

Section 2 recalls the SIDH protocol and gives some useful lemmas that are used in our soundness proofs. Section 3 presents some isogeny-based hardness assumptions and reductions, including the new decisional assumptions we need for our zero-knowledge proofs. We then recall the De Feo–Jao–Plût identification scheme in Section 4.1 and outline the issue with its proof of soundness in Section 4.2. In Section 5 we present our protocols for the weak SIDH relation: A sound but potentially insecure protocol first, a zero-knowledge modification then. Section 6 presents a protocol to prove correctness of the points in the SIDH public key. In Section 7, we conclude with some standard discussion on how a NIZK scheme which is a Proof of Knowledge (PoK) of an SIDH secret key can be constructed from our last scheme—the first such scheme that is sound and proves correctness of the points in the public key (a protection mechanism against adaptive attacks [GPST16, DGL^{+}20]). Section 8 describes some open problems and future directions.

1.3 Note to reader: Changes from earlier versions of the paper

There are several major changes in this version.

The biggest is changing the main protocols from binary to ternary challenges. This is due to a subtlety in the definition of computational ZK. Earlier versions of our paper required a new computational assumption. It turned out that, with respect to Definition 3, this assumption did not hold. The DSSP and Double-DSSP assumptions do not suffer from this defect. To resolve this we have modified the schemes to use ternary challenges.

On the positive side, some aspects of the paper have become simpler. And we added cryptanalysis of [UJ20].

1.4 Acknowledgements

We thank David Jao, Jason LeGrow, and Yi-Fu Lai for useful discussion about this work. We also thank Paulo Barreto for catching some typos in this paper, and Simon-Philipp Merz for valuable comments. We thank Javad Doliskani for important observations that inspired significant improvements to this work. Finally, we would like to thank those involved with the BIRS Supersingular Isogeny Graphs in Cryptography workshop for great discussion on some questions this work raised—especially Lorenz Panny and his work analyzing SIDH squares in small fields.

2 Preliminaries

Notation. As a convention, we will use K_ϕ to denote a point which generates the kernel of an isogeny ϕ. Let $[t]$ denote the set $\{1, \ldots, t\}$. All isogenies in this paper are assumed to be separable. The notation $\hat{\psi}$ denotes the dual isogeny of ψ.

2.1 SIDH

We now provide a brief refresher on the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange protocol [JD11, DJP14] by De Feo, Jao, and Plût.

As public parameters, we have a prime $p = \ell_1^{e_1} \cdot \ell_2^{e_2} \cdot f \pm 1$, where ℓ_1, ℓ_2 are small primes, f is an integer cofactor, and $\ell_1^{e_1} \approx \ell_2^{e_2}$. We work over the finite field $\mathbb{F}_{p^2}$. Additionally we fix a base supersingular elliptic curve E and bases $\{P_1, Q_1\}$, $\{P_2, Q_2\}$ for both the $\ell_1^{e_1}$ and $\ell_2^{e_2}$-torsion subgroups of $E(\mathbb{F}_{p^2})$ respectively (such that $E[\ell_i^{e_i}] = \langle P_i, Q_i \rangle$). Typically $\ell_1 = 2$ and $\ell_2 = 3$.

It is well known that knowledge of an isogeny (up to isomorphism) and knowledge of its kernel are equivalent, and we can convert between them at will, via Vélu's formulae [Vél71]. In SIDH, the secret keys of Alice and Bob are isogenies $\phi_A : E(\mathbb{F}_{p^2}) \to E_A(\mathbb{F}_{p^2})$, $\phi_B : E(\mathbb{F}_{p^2}) \to E_B(\mathbb{F}_{p^2})$ of degree $\ell_1^{e_1}$ and $\ell_2^{e_2}$, respectively. These isogenies are generated by randomly choosing secret integers $a_i, b_i \in \mathbb{Z}/\ell_i^{e_i}\mathbb{Z}$ (not both divisible by ℓ_i) and computing the isogeny with kernel generated by $K_i = [a_i]P_i + [b_i]Q_i$. We thus unambiguously refer to the isogeny, its kernel, and such integers a, b, as "the secret key."

$$\begin{tikzcd}
E \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E_A \arrow[d, "\phi_{AB}"] \\
E_B \arrow[r, "\phi_{BA}"'] & E_{AB}
\end{tikzcd}$$

Figure 1: Commutative diagram of SIDH, where ker$(\phi_{BA}) = \phi_B($ker$(\phi_A))$ and ker$(\phi_{AB}) = \phi_A($ker$(\phi_B))$.

Figure 1 depicts the commutative diagram making up the key exchange. In order to make the diagram commute, Alice and Bob are required to not only give their image curves E_A and E_B in their respective public keys, but also the images of the basis points of the other participant's kernel on E. That is, Alice provides E_A, $P_2' = \phi_A(P_2), Q_2' = \phi_A(Q_2)$ as her public key. This allows Bob to "transport" his secret isogeny to E_A and compute ϕ_{AB} whose kernel is $\langle [a_2]P_2' + [b_2]Q_2' \rangle$. Both Alice and Bob will arrive along these transported isogenies at isomorphic image curves E_{AB}, E_{BA} (using Vélu's formulae, they will actually arrive at exactly the same curve [Leo20]). Two elliptic curves are isomorphic over $\overline{\mathbb{F}}_p$ if and only if their j-invariants are equal, $j(E_{AB}) = j(E_{BA})$, hence this j-invariant may be used as the shared secret of the SIDH key exchange.

Some cryptographic hardness assumptions related to isogenies and SIDH are discussed in Section 3.

2.2 Isogeny squares

We collect here some basic definitions and lemmas that we will use repeatedly throughout the paper. In the statements below, all elliptic curves are defined over a field of characteristic p.

Definition 1 (Independent points, isogenies). *Let E be an elliptic curve, let $\ell \neq p$ be a prime and e an integer, let (P, Q) be a basis of $E[\ell^e]$. Let $R = [a]P + [b]Q$ and $S = [c]P + [d]Q$. The following conditions are equivalent:*

(a) (R, S) form a basis of $E[\ell^e]$.

(b) ℓ does not divide $ad - bc$, i.e., the matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is invertible modulo ℓ^e.

(c) The value of the ℓ^e-th Weil pairing $w = e(R, S)$ has order ℓ^e, i.e., $w^{\ell^{e-1}} \neq 1$.

When R, S satisfy any of these, we say they are independent *of one another. Similarly, we say that two cyclic groups of order ℓ^e are independent whenever any of their generators are. Finally, we say that two isogenies of degree ℓ^e are independent if their kernels are.*

Proof. $(a) \Rightarrow (b)$: Both P, Q and R, S are bases of the same torsion subgroup $E[\ell^e]$. Hence, $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is a change-of-basis from P, Q to R, S and there must be an inverse change-of-basis A^{-1} from R, S to P, Q. Then A is necessarily invertible, and therefore, so too is its determinant $ad - bc$ modulo ℓ^e.

$(b) \Rightarrow (c)$: We have that

$$w = e(R, S) = e([a]P + [b]Q, [c]P + [d]Q).$$

Then since e is bilinear, $w = e(P, Q)^{ad-bc}$. Now $e(P, Q)$ has order ℓ^e because e is surjective onto the group of ℓ^e-th roots of unity (c.f. [Sil09, Corollary III.8.1.1]), and since $\ell \nmid ad - bc$, then w must also have order ℓ^e.

$(c) \Rightarrow (a)$: Recall that $E[\ell^e] \simeq \mathbb{Z}/\ell^e\mathbb{Z} \times \mathbb{Z}/\ell^e\mathbb{Z}$ [Sil09, Corollary III.6.4b]. Thus, in order for R, S to form a basis, we must show $\langle R \rangle \cap \langle S \rangle = \{\mathcal{O}_E\}$.

Suppose $[w]R = [z]S \neq \mathcal{O}_E$ for some integers w, z. By assumption, it must be that $\ell^e \nmid w$ and $\ell^e \nmid z$. Now consider $e([w]R - [z]S, S) = 1$, since $e(\mathcal{O}_E, T) = 1$ for any T. By the bilinearity of the pairing, this gives

$$e([w]R - [z]S, S) = e(R, S)^w e(S, S)^{-z} = 1.$$

Then, because $e(S, S) = 1$, we arrive at the conclusion $e(R, S)^w = 1$, which is a contradiction since $e(R, S)$ has order ℓ^e and $\ell^e \nmid w$. Thus, there can exist no such integers w, z, and therefore $\langle R \rangle \cap \langle S \rangle = \{\mathcal{O}_E\}$. □

Lemma 1. *Let $\phi : E \to E/\langle R \rangle$ be an isogeny of kernel $\langle R \rangle$ and degree ℓ^e, let S be a point of order ℓ^e independent to R. Then $\phi(S)$ has order ℓ^e and generates* $\ker(\widehat{\phi})$.

Proof. Because R and S are independent (Definition 1), the subgroups generated by R and S intersect trivially. Thus, since ϕ has kernel $\langle R \rangle$, no non-trivial point in $\langle S \rangle$ is in the kernel of ϕ. Furthermore, we know that $\widehat{\phi} \circ \phi = [\ell^e]$ has kernel $E[\ell^e]$, and that $S \in E[\ell^e]$. Thus $\widehat{\phi}(\phi(S)) = \mathcal{O}$, implying $\phi(S)$ is in the kernel of $\widehat{\phi}$. The same holds for all elements $S' = [\lambda]S \in \langle S \rangle$, and since $\phi(S') \neq \mathcal{O}$ for all non-trivial S', $\phi(S)$ has order ℓ^e and generates $\ker(\widehat{\phi})$. □

The following lemma is the main tool we are going to use, repeatedly, to design all proofs of knowledge.

Lemma 2. *Let ℓ_1, ℓ_2 be distinct primes different from p, let e_1, e_2 be integers. Let $\phi_A : E \to E_A$ be an isogeny of degree $\ell_1^{e_1}$. Let $\phi_B : E \to E_B$ and $\phi_{AB} : E_A \to E_{AB}$ be isogenies of degree $\ell_2^{e_2}$ such that* $\ker(\phi_{AB}) = \phi_A(\ker(\phi_B))$. *Then there exists an isogeny $\phi_{BA} : E_B \to E_{AB}$ of degree $\ell_1^{e_1}$.*

Proof. Let K_A be a generator of $\ker(\phi_A)$. Then because the degrees of ϕ_A, ϕ_B are coprime, $\phi_B(K_A)$ also has order $\ell_1^{e_1}$ and generates the kernel of some isogeny

$$\chi : E_B \to E_B/\langle \phi_B(K_A) \rangle.$$

Observe that E_{AB} is defined as the codomain of $\phi_{AB} \circ \phi_A$. We thus have that $E_{AB} \cong E/\langle K_A, K' \rangle$ for a point K' of order $\ell_2^{e_2}$ such that $\langle \phi_A(K') \rangle = \ker(\phi_{AB})$. Because $\ker(\phi_{AB}) = \phi_A(\ker(\phi_B))$, we conclude $\langle K' \rangle = \ker(\phi_B)$. Therefore, $E_B/\langle \phi_B(K_A) \rangle \cong E_{AB}$ as required. □

2.3 Sigma protocols

A sigma protocol Π_Σ for a relation $\mathcal{R} = \{(X, W)\}$ is a public-coin three-move interactive proof system consisting of two parties: A verifier V and a prover P. Recall that public-coin informally means that there are no secret sources of randomness—the verifier's coin tosses are accessible to the prover. In practice this means the challenge sent by the verifier to the prover is uniformly random. For our purposes, a witness W can be thought of as a secret key, while the statement X is the corresponding public key. Thus, proving $(X, W) \in \mathcal{R}$ is equivalent to saying that X is a valid public key for which a corresponding secret key exists. We use the security parameter κ to parametrize the length of the secret keys involved.

Definition 2 (Sigma protocol)**.** *A sigma protocol Π_Σ for a family of relations $\{\mathcal{R}\}_\kappa$ parametrized by security parameter κ consists of PPT algorithms $((P_1, P_2), (V_1, V_2))$ where V_2 is deterministic and we assume P_1, P_2 share states. The protocol proceeds as follows:*

1. *Round 1: The prover, on input $(X, W) \in \mathcal{R}$, returns a commitment* $\mathsf{com} \leftarrow P_1(X, W)$ *which is sent to the verifier.*
2. *Round 2: The verifier, on receipt of* com*, runs* $\mathsf{chall} \leftarrow V_1(1^\kappa)$ *to obtain a random challenge, and sends this to the prover.*
3. *Round 3: The prover then runs* $\mathsf{resp} \leftarrow P_2(X, W, \mathsf{chall})$ *and returns* resp *to the verifier.*
4. *Verification: The verifier runs* $V_2(X, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ *and outputs either* $\top$ (accept) *or* $\bot$ (reject).

A transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$ is said to be valid if $V_2(X, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ outputs $\top$. Let $\langle P, V\rangle$ denote the transcript for an interaction between prover P and verifier V. The main requirements of a sigma protocol are:

Correctness: If the prover P knows $(X, W) \in \mathcal{R}$ and behaves honestly, then the verifier V accepts.

n**-special soundness:** There exists a polynomial-time extraction algorithm that, given a statement X and n valid transcripts

$$(\mathsf{com}, \mathsf{chall}_1, \mathsf{resp}_1), \ldots, (\mathsf{com}, \mathsf{chall}_n, \mathsf{resp}_n)$$

where $\mathsf{chall}_i \neq \mathsf{chall}_j$ for all $1 \leq i < j \leq n$, outputs a witness W such that $(X, W) \in \mathcal{R}$ with probability at least $1 - \varepsilon$ for soundness error ε.

A sound sigma protocol for $\mathcal{R}$ is also called a **Proof of Knowledge (PoK)** for $\mathcal{R}$.

Special Honest Verifier Zero-knowledge (SHVZK): If there exists a polynomial-time simulator that, given a statement X and a challenge chall, outputs a valid transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$ that is indistinguishable from a real transcript.

Definition 3. *A sigma protocol (P, V) is computationally special honest verifier zero-knowledge if there exists a probabilistic polynomial time simulator* Sim *such that for all probabilistic polynomial time stateful adversaries* Adv

$$\begin{aligned} &\Pr\left[(X, W, \mathsf{chall}) \leftarrow \mathsf{Adv}(1^\kappa); \mathsf{com} \leftarrow P_1(X, W); \mathsf{resp} \leftarrow P_2(X, W, \mathsf{chall}) : \mathsf{Adv}(\mathsf{com}, \mathsf{chall}, \mathsf{resp}) = 1\right] \\ &\qquad \approx \Pr\left[(X, W, \mathsf{chall}) \leftarrow \mathsf{Adv}(1^\kappa); (\mathsf{com}, \mathsf{resp}) \leftarrow \mathsf{Sim}(X, \mathsf{chall}) : \mathsf{Adv}(\mathsf{com}, \mathsf{chall}, \mathsf{resp}) = 1\right]. \end{aligned} \quad (1)$$

Although SHVZK is not a particularly strong flavour of zero-knowledge, there exist efficient transformations to full zero-knowledge that incur only a small overhead in communication and computation [DGOW95, Dam00, GMY06]. In particular, it is well known that SHVZK is sufficient to obtain full non-interactive zero-knowledge in the random oracle model [BR93].

An earlier version of our paper proposed schemes with binary challenges whose security required a certain computational assumption. It turned out that with respect to Definition 3 this assumption did not hold. To resolve this we have modified the schemes to use ternary challenges.

3 SIDH problems and assumptions

In this section, we recall some standard isogeny-based hardness assumptions of relevance to this work. We then introduce a new decisional assumption which will be useful for the proof of zero-knowledge in Section 6. The first two are computational isogeny-finding problems.

Definition 4 (General isogeny problem)**.** *Given j-invariants $j, j' \in \mathbb{F}_{p^2}$, find an isogeny $\phi : E \to E'$ if one exists, where $j(E) = j$ and $j(E') = j'$.*

This is the foundational hardness assumption of isogeny-based cryptography, that it is hard to find an isogeny between two given curves. Note the decisional version, determining whether an isogeny exists, is easy—an isogeny exists if and only if $\#E(\mathbb{F}_{p^2}) = \#E'(\mathbb{F}_{p^2})$.

Definition 5 (Computational Supersingular Isogeny (CSSI) problem)**.** *For fixed SIDH prime p, base curve E_0, and $\ell_2^{e_2}$-torsion basis $P_0, Q_0 \in E_0$, let $\phi : E_0 \to E_1$ be an isogeny of degree $\ell_1^{e_1}$. Given an SIDH public key $(E_1, P_1 = \phi(P_0), Q_1 = \phi(Q_0))$, find an isogeny $\phi' : E_0 \to E_1$ of degree $\ell_1^{e_1}$ such that $P_1, Q_1 = \phi'(P_0), \phi'(Q_0)$.*

This is problem 5.2 of [DJP14] and essentially states that it is hard to find the secret key corresponding to a given public key. This problem is also called the SIDH isogeny problem by [GV18, Definition 2].

At the heart of the GPST adaptive attack is the problem that, given a public key (E_1, P_1, Q_1), we cannot validate that P_1, Q_1 are indeed the correct images of basis points P_0, Q_0 under the secret isogeny ϕ. The best we know how to do is to check they are indeed a basis of the correct order, and use the Weil pairing check

$$e_{\ell_2^{e_2}}(P_1, Q_1) = e_{\ell_2^{e_2}}(P_0, Q_0)^{\deg \phi}.$$

Unfortunately this holds for many different choices of basis points. Indeed, if (P_1, Q_1) are the correct images, then any pair $(aP_1 + bQ_1, cP_1 + dQ_1)$ such that $ad - bc = 1 \bmod \ell_2^{e_2}$ also passes the check. So this is not enough to uniquely determine ϕ, and, in particular, is insufficient to protect against the GPST adaptive attack.

The following decisional problem follows Definition 3 of [GV18] and is also very similar to the key validation problem of Urbanik and Jao [UJ18, Problem 3.4] (the key validation problem asks whether a ϕ of degree *dividing* $\ell_1^{e_1}$ exists). However, the previous definitions did not take the Weil pairing check into account, which would serve as a distinguisher.

Definition 6 (Decisional SIDH isogeny (DSIDH) problem)**.** *The decisional SIDH problem is to distinguish between the following two distributions:*

- $\mathcal{D}_0 = \{(E_0, P_0, Q_0, E_1, P_1, Q_1)\}$ *such that E_0 is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, P_0, Q_0 a basis such that $E_0[\ell_2^{e_2}] = \langle P_0, Q_0 \rangle$, $\phi : E_0 \to E_1$ is an isogeny of degree $\ell_1^{e_1}$, and $P_1 = \phi(P_0)$ and $Q_1 = \phi(Q_0)$.*
- $\mathcal{D}_1 = \{(E_0, P_0, Q_0, E_1, P_1, Q_1)\}$ *such that E_0 is a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, P_0, Q_0 a basis such that $E_0[\ell_2^{e_2}] = \langle P_0, Q_0 \rangle$, E_1 is any supersingular elliptic curve over $\mathbb{F}_{p^2}$ with the same cardinality as E_0, and P_1, Q_1 is a basis of $E_1[\ell_2^{e_2}]$ satisfying the Weil pairing check $e_{\ell_2^{e_2}}(P_1, Q_1) = e_{\ell_2^{e_2}}(P_0, Q_0)^{\ell_1^{e_1}}$.*

As shown by Galbraith and Vercauteren [GV18], Thormarker [Tho17], and Urbanik and Jao [UJ18], being able to solve this decisional problem is as hard as solving the computational (CSSI) problem, so key validation is fundamentally difficult. This is done by testing ℓ_1-isogeny neighboring curves of E_1 and learning the correct path one bit at a time.

Definition 7 (Decisional Supersingular Product (DSSP) problem)**.** *Given an isogeny $\phi : E_0 \to E_1$ of degree $\ell_1^{e_1}$, the decisional supersingular product problem is to distinguish between the following two distributions:*

- $\mathcal{D}_0 = \{(E_2, E_3, \phi')\}$ *such that there exists a cyclic subgroup $G \subseteq E_0[\ell_2^{e_2}]$ of order $\ell_2^{e_2}$ and $E_2 \cong E_0/G$ and $E_3 \cong E_1/\phi(G)$, and $\phi' : E_2 \to E_3$ is a degree $\ell_1^{e_1}$ isogeny.*
- $\mathcal{D}_1 = \{(E_2, E_3, \phi')\}$ *such that E_2 is a random supersingular curve with the same cardinality as E_0, and E_3 is the codomain of a random isogeny $\phi' : E_2 \to E_3$ of degree $\ell_1^{e_1}$.*

This is problem 5.5 of [DJP14] and intuitively states that it is hard to determine whether there exist valid "vertical sides" to an SIDH square given the corners and the bottom horizontal side.

3.1 Double variant

In Section 6, we propose a scheme which uses two independent SIDH squares in each round of the sigma protocol. For the zero-knowledge proof in that section, we require a "double" variant of the DSSP problem.

The Double-DSSP problem differs from the "single" version by the introduction of two bases U'_i, V'_i of the $\ell_1^{e_1}$-torsion subgroups on $E_{2,i}$, for $i \in \{0,1\}$. As we shall see in Section 6, these extra points will be used to verify that the two independent SIDH squares in the "double" protocol both use consistent isogenies ϕ'_i. These extra points, plus the requirement that the isogenies ψ_i used in each of the two squares should be independent, mean a reduction from DSSP to the Double-DSSP problem is unlikely. We believe Double-DSSP is a hard problem.

Definition 8 (Double-DSSP Problem)**.** *Given an isogeny $\phi : E_0 \to E_1$ of degree $\ell_1^{e_1}$, let $\mathcal{D}_0$ and $\mathcal{D}_1$ denote the two distributions in the DSSP problem. The double decisional supersingular product problem is to distinguish between the following two distributions:*

- $\mathcal{D}'_0 = \{(\mathsf{inst}_i, U'_i, V'_i)_{i\in\{0,1\}}\}$ *where* $\mathsf{inst}_i = (E_{2,i}, E_{3,i}, \phi'_i) \leftarrow \mathcal{D}_0$*, and additionally, if* $\psi_i : E_0 \to E_{2,i}$ *are the respective isogenies of degree* $\ell_2^{e_2}$*, then* ψ_0 *and* ψ_1 *are independent and* $U'_i, V'_i = \psi_i(U), \psi_i(V)$ *where* $\{U, V\}$ *is a random (secret) basis of* $E_0[\ell_1^{e_1}]$*.*
- $\mathcal{D}_1 = \{(\mathsf{inst}_i, U'_i, V'_i)_{i\in\{0,1\}}\}$ *where* $\mathsf{inst}_i = (E_{2,i}, E_{3,i}, \phi'_i) \leftarrow \mathcal{D}_1$*, and* U'_i, V'_i *is a random basis of the* $\ell_1^{e_1}$ *torsion subgroup on* $E_{2,i}$ *such that* $e_{\ell_1^{e_1}}(U'_0, V'_0) = e_{\ell_1^{e_1}}(U'_1, V'_1)$ *and for any generator* K_i *of* $\ker(\phi'_i)$

$$e_{\ell_1^{e_1}}(U'_0, K_0)e_{\ell_1^{e_1}}(K_1, V_1) = e_{\ell_1^{e_1}}(K_0, V'_0)e_{\ell_1^{e_1}}(U'_1, K_1).$$

4 Previous SIDH identification scheme and soundness issue

4.1 De Feo–Jao–Plût scheme

Let p be a large prime of the form $\ell_1^{e_1} \cdot \ell_2^{e_2} \cdot f \pm 1$, where ℓ_1, ℓ_2 are small primes. We start with a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ with $\#E_0(\mathbb{F}_{p^2}) = (\ell_1^{e_1}\ell_2^{e_2}f)^2$. The private key is a uniformly random point $K_\phi \in E_0(\mathbb{F}_{p^2})$ of exact order $\ell_1^{e_1}$. Define $E_1 = E_0/\langle K_\phi\rangle$ and denote the corresponding $\ell_1^{e_1}$-isogeny by $\phi : E_0 \to E_1$.

Let P_0, Q_0 be a basis of the torsion subgroup $E_0[\ell_2^{e_2}] = \langle P_0, Q_0\rangle$. The fixed public parameters are $pp = (p, E_0, P_0, Q_0)$. The public key is $(E_1, \phi(P_0), \phi(Q_0))$. The private key is the kernel generator K_ϕ (equivalently, the isogeny ϕ). The interaction goes as follows:

1. The prover chooses a random primitive $\ell_2^{e_2}$-torsion point K_ψ as $K_\psi = [a]P_0 + [b]Q_0$ for some integers $0 \le a, b < \ell_2^{e_2}$ not both divisible by ℓ_2. Note that $\phi(K_\psi) = [a]\phi(P_0) + [b]\phi(Q_0)$. The prover defines the curves $E_2 = E_0/\langle K_\psi\rangle$ and $E_3 = E_1/\langle\phi(K_\psi)\rangle = E_0/\langle K_\psi, K_\phi\rangle$, and uses Vélu's formulae to compute the following diagram.

$$\begin{tikzcd} E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\ E_2 \arrow[r, "\phi'"] & E_3 \end{tikzcd}$$

The prover sends commitment $\mathsf{com} = (E_2, E_3)$ to the verifier.

2. The verifier challenges the prover with a uniformly random bit $\mathsf{chall} \leftarrow \{0,1\}$.
3. If $\mathsf{chall} = 0$, the prover reveals $\mathsf{resp} = (a,b)$ from which K_ψ and $\phi(K_\psi) = K_{\psi'}$ can be reconstructed. If $\mathsf{chall} = 1$, the prover reveals $\mathsf{resp} = (\psi(K_\phi) = K_{\phi'})$.

In both cases, the verifier accepts the proof if the points revealed have the correct order and generate kernels of isogenies between the correct curves. We iterate this process t times to reduce the cheating probability (where t is chosen based on the security parameter κ). Note that in an honest execution of the proof, we have

$$\widehat{\psi'} \circ \phi' \circ \psi = [\ell_2^{e_2}]\phi.$$

Note that in this basic scheme (and all protocols known in the literature) honest transcripts involve responses like K_ψ and $\phi(K_\psi)$. Hence it is natural to allow the proof to reveal $\phi(P_0), \phi(Q_0)$ where $\{P_0, Q_0\}$ is a basis for $E_0[\ell_2^{e_2}]$.

4.2 Issue with soundness proofs for the De Feo–Jao–Plût scheme

A core component of the security proof of the De Feo–Jao–Plût identification scheme is the soundness proof. A proof of soundness was given by multiple previous works [DJP14, YAJ+17, GPS20]. A sketch of it is as follows:

Suppose $\mathcal{A}$ is an adversary that takes as input the public key and succeeds in the identification protocol (all t iterations) with noticeable probability ϵ. Given a challenge instance $(E_0, E_1, R_0, S_0, \phi(R_0), \phi(S_0))$ for the CSSI problem, we run $\mathcal{A}$ on the tuple $(E_1, \phi(R_0), \phi(S_0))$ as the public key. In the first round, $\mathcal{A}$ outputs commitments $(E_{i,2}, E_{i,3})$ for $1 \leq i \leq t$. We then send a challenge $b \in \{0,1\}^t$ to $\mathcal{A}$ and, with probability ϵ, $\mathcal{A}$ outputs a response that satisfies the verification algorithm. Now, we use the standard replay technique: Rewind $\mathcal{A}$ to the point where it had output its commitments and then respond with a different challenge $b' \in \{0,1\}^t$. With probability ϵ, $\mathcal{A}$ outputs a valid response. This gives exactly the 2-special soundness requirement of two valid transcripts with the same commitment but different challenges.

Now, choose some index i such that $b_i \neq b'_i$. We now restrict our focus to the components (E_2, E_3) for that index, and the two responses. It means $\mathcal{A}$ sent E_2, E_3 and can answer both challenges $b = 0$ and $b = 1$ successfully. Hence $\mathcal{A}$ has provided the maps ψ, ϕ', ψ' in the following diagram.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\phi"] \arrow[r, bend right, "\tilde{\phi}"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\
E_2 \arrow[r, "\phi'"] & E_3
\end{tikzcd}$$

The argument proceeds as follows: We have an explicit description of an isogeny $\tilde{\phi} = \widehat{\psi'} \circ \phi' \circ \psi$ from E_0 to E_1. The degree of $\tilde{\phi}$ is $\ell_1^{e_1}\ell_2^{2e_2}$. One can determine $\ker(\tilde{\phi}) \cap E_0[\ell_1^{e_1}]$ by iteratively testing points in $E_0[\ell_1^j]$ for $j = 1, 2, \ldots$. Hence, one determines the kernel of ϕ, as desired.

However, the important issue with this argument which has so far gone unnoticed, is that it assumes $\ker(\phi) = \ker(\tilde{\phi}) \cap E_0[\ell_1^{e_1}]$. This assumption has no basis, and we will provide a simple counterexample to this argument in the following section. While we always recover an isogeny, it may not be ϕ at all—it is entirely possible the isogeny we recover does not even have codomain E_1 so this proof of 2-special soundness is not valid.

4.3 Counterexample to soundness

Fix a supersingular curve E_0 as above. Generate a random $\ell_2^{e_2}$-torsion point $K_\psi \in E_0(\mathbb{F}_{p^2})$ as $K_\psi = [a]P_0 + [b]Q_0$ for some integers $0 \leq a, b < \ell_2^{e_2}$ not both divisible by ℓ_2. Let $\psi : E_0 \to E_2$ have kernel generated by K_ψ. Then choose a random isogeny $\phi' : E_2 \to E_3$ of degree $\ell_1^{e_1}$ with kernel generated by $K_{\phi'}$. Then choose a random isogeny $\psi' : E_3 \to E_1$ of degree $\ell_2^{e_2}$. Choose points $P_0', Q_0' \in E_1(\mathbb{F}_{p^2})$ such that $\ker(\widehat{\psi'}) = \langle [a]P_0' + [b]Q_0' \rangle$. Then publish

$$(E_0, E_1, P_0, Q_0, P_0', Q_0')$$

as a public key. In other words, we have

$$E_0 \xrightarrow{\psi} E_2 \xrightarrow{\phi'} E_3 \xrightarrow{\psi'} E_1$$

Now there is no reason to believe that there exists an isogeny from E_0 to E_1 of degree $\ell_1^{e_1}$, yet we can respond to both challenge bits 0 and 1 in a single round of the identification scheme. Pulling back the kernel of ϕ' via ψ to E_0 will result in the kernel of an isogeny which, in general, will not have codomain E_1 (but instead a random other curve). This is because ψ' is entirely unrelated to ψ in this case (they are not "parallel"), so we have no SIDH square.

The key observation is that a verifier could be fooled into accepting this public key by a prover who always uses the same curves (E_2, E_3) instead of randomly chosen ones. When $b = 0$ the prover responds with the pair (a, b) corresponding to the kernel of ψ and $\widehat{\psi'}$, and when $b = 1$ the prover responds with $K_{\phi'}$. The verifier will agree that all responses are correct and will accept the proof.

It is true that the verifier could test whether the commitments (E_2, E_3) are being re-used, but this has never been stated as a requirement in any of the protocol descriptions. To tweak the verification protocol we need to know how "random" the pairs (E_2, E_3) (or, more realistically, the pairs (a, b)) need to be. One may think that the original scheme seems to be secure despite the issue with the proof, as long as the commitment (E_2, E_3) is not reused every time. However, in experiments with small primes, it is entirely possible to construct instances[1] where even with multiple different commitments, a secret isogeny of the correct degree between E_0 and E_1 does not exist. We expect that this extrapolates to large primes too, although one could potentially argue that finding enough such instances is computationally infeasible.

It is also true that repeating (E_2, E_3) means the protocol is no longer zero-knowledge. We emphasize that soundness and zero-knowledge are independent security properties, which are proved separately (and affect different parties: one gives an assurance to the verifier and the other to the prover). The counterexample we have provided is a counterexample to the soundness proof. The fact that the counterexample is not consistent with the proof that the protocol is zero-knowledge is irrelevant.

Finally, one could consider basing security of the protocol on the general isogeny problem (Definition 4) because, even in our counterexample, an isogeny $E_0 \to E_1$ exists and can be extracted—it just doesn't have degree $\ell_1^{e_1}$. We find it interesting that none of the previous authors chose to do it that way. However, some applications may require using the identification/signature protocols to prove that an SIDH public key is well-formed, implying the secret isogeny has the correct degree. For such applications we need soundness to be rigorously proved.

The issue in the security proofs in the literature is not only that it is implicitly assumed that there is an isogeny of degree $\ell_1^{e_1}$ between E_0 and E_1. The key issue is that it is implicitly assumed that the pullback under ψ of $\ker(\phi')$ is the kernel of this isogeny. Our counterexample calls these assumptions into question, and shows that the proofs are incorrect as written.

To make this very clear, consider the soundness proof from De Feo, Jao, and Plût [DJP14]. The following diagram is written within the proof. It implicitly assumes that the horizontal isogeny ϕ' has kernel given by $\psi(S)$, so that the image curve is $E/\langle S, R \rangle$.

[1] Thank you to Lorenz Panny for demonstrating this.

$$\begin{tikzcd}
E \arrow[d, "\psi"'] & E/\langle S \rangle \arrow[d, "\psi'"] \\
E/\langle R \rangle \arrow[r, "\phi'"'] & E/\langle S, R \rangle
\end{tikzcd}$$

This implicit assumption seems to have been repeated in all subsequent works, such as [YAJ$^+$17] and [GPS20].

4.4 Soundness of UJ20

Urbanik and Jao [UJ20] give a variant of SIDH that exploits automorphisms and gets essentially three SIDH keys out of single protocol messages. Section 5 of their paper claims an isogeny-based zero-knowledge identification protocol that validates all elements of an SIDH key.

The statement being proved is $(E, P_B, Q_B, E_A, P'_B, Q'_B)$ and the witness is an isogeny $\phi : E \to E_A = E/A$ with $P'_B = \phi(P_B), Q'_B = \phi(Q_B)$. (Here the symbol A is overloaded to signify "Alice" and also Alice's subgroup that is the kernel of the isogeny.) Here the base curve E has a non-trivial automorphism η of order 6.

The proof works by sending E/B such that there are three SIDH keys that can be computed by Alice and Bob: $E_1 = E/\langle A, B \rangle, E_2 = E/\langle \eta(A), B \rangle, E_3 = E/\langle \eta^2(A), B \rangle$. More precisely, the prover picks $B = \langle [a]P_B + [b]Q_B \rangle$ and commits to the three related squares. The verifier makes a challenge $\mathsf{chall} \in \{0, 1, 2, 3\}$. When $\mathsf{chall} = 0$ the prover reveals (a, b), and the verifier can check all three isogenies $E \to E_B, E_A \to E_i$ for $i \in \{1, 2, 3\}$. When $\mathsf{chall} \geq 1$ the prover reveals the kernel of an isogeny $E_B \to E_{\mathsf{chall}}$.

There is no formal proof of soundness given in [UJ20].

First, it is easy to see that if P'_B and Q'_B are the correct image points, then replacing them with $[z]P'_B$ and $[z]Q'_B$ for any invertible z modulo the order of P'_B is also accepted by the verifier. So it is clear that the protocol is at most giving an assurance of a weaker statement than claimed.

However, the protocol fails more drastically due to a similar issue to the problem discussed in Section 5.2. Briefly, because (a, b) is chosen by the prover, the prover can "hide" their cheating. For example, suppose a dishonest prover sets $P'_B = \phi(P_B), Q'_B = \phi(Q_B) + T$ where T is a point of order ℓ_2 (a divisor of the order of P_B and Q_B). Then as long as b is chosen to be a multiple of ℓ_2 we have

$$[a]P'_B + [b]Q'_B = [a]\phi(P_B) + [b]\phi(Q_B)$$

and so the cheating is not detected by the verifier.

5 Steps towards an SIDH proof – the weak SIDH relation

The purpose of this section is to present a protocol to prove in zero-knowledge a natural but weaker statement than the knowledge of an SIDH secret key. In the next section we will augment this protocol to prove the full SIDH statement.

5.1 A sound but insecure protocol

We start with a simple protocol which follows the blueprint of De Feo–Jao–Plût, but fixes its soundness issue. Unfortunately, the fix breaks zero-knowledge, and we will need to change the protocol again to achieve our goal.

Let public parameters $pp = (p, \ell_1, \ell_2, e_1, e_2, E_0)$ be such that $\#E_0(\mathbb{F}_{p^2}) = \ell_1^{e_1}\ell_2^{e_2}$. As before, suppose a user has a secret isogeny $\phi : E_0 \to E_1$ of degree $\ell_1^{e_1}$ with kernel $\ker(\phi) = \langle K_\phi \rangle$. In this section we are only interested in proving knowledge of ϕ, thus we will not consider the public torsion basis (P_0, Q_0) and its image (P_1, Q_1) by ϕ.

Our simple (but insecure) protocol is presented in Figure 2. It includes some basic functions:

- IsogenyFromKernel is a function taking a point $S \in E$ and outputting a (normalised) isogeny with kernel $\langle S \rangle$ and codomain curve $E/\langle S \rangle$.
- $\mathsf{RandomBasis}_i$ is a function taking a curve and outputting a uniformly random pair of points U, V which generate the $\ell_i^{e_i}$-torsion subgroup on the given curve, for $i = 1, 2$.
- DualKernel is a function taking an isogeny ψ and outputting a generator $K_{\widehat{\psi}}$ of the kernel of the dual isogeny $\widehat{\psi}$.

Intuitively, the sigma protocol follows Section 4.1, with a single bit challenge—if the challenge is 0, we reveal the vertical isogenies ψ, ψ', while if the challenge is 1, we reveal the horizontal ϕ'. The difference is the introduction of additional points on E_3 to the commitment, which force ψ, ψ' to be, in some sense, "compatible" or "parallel". This restriction lets us prove 2-special soundness by extracting the secret ϕ from two accepting transcripts.

round 1 (commitment)

```
Sample uniformly random $\ell_2^{e_2}$-isogeny kernel $\langle K_\psi \rangle \subset E_0$
$\psi, E_2 \leftarrow \mathsf{IsogenyFromKernel}(K_\psi)$
$P_2, Q_2 \leftarrow \mathsf{RandomBasis}_2(E_2)$
$K_{\phi'} \leftarrow \psi(K_\phi) \in E_2$
$\phi', E_3 \leftarrow \mathsf{IsogenyFromKernel}(K_{\phi'})$
$P_3, Q_3 \leftarrow \phi'(P_2), \phi'(Q_2) \in E_3$
Prover sends $\mathsf{com} \leftarrow (E_2, P_2, Q_2, E_3, P_3, Q_3)$ to Verifier.
```

round 2 (challenge)

```
Verifier sends $\mathsf{chall} \leftarrow \{0, 1\}$ to Prover.
```

round 3 (response)

```
if $\mathsf{chall} = 1$ then
    $\mathsf{resp} \leftarrow K_{\phi'}$
else
    $K_{\widehat{\psi}} \leftarrow \mathsf{DualKernel}(\psi)$
    Write $K_{\widehat{\psi}} = [c]P_2 + [d]Q_2$ for $c, d \in \mathbb{Z}/\ell_2^{e_2}\mathbb{Z}$
    $\mathsf{resp} \leftarrow (c, d)$
Prover sends resp to Verifier.
```

Verification

```
$(E_2, P_2, Q_2, E_3, P_3, Q_3) \leftarrow \mathsf{com}$
if $\mathsf{chall} = 1$ then
    $K_{\phi'} \leftarrow \mathsf{resp}$
    Check $K_{\phi'}$ has order $\ell_1^{e_1}$ and lies on $E_2$, otherwise output reject
    $\phi', E_3' \leftarrow \mathsf{IsogenyFromKernel}(K_{\phi'})$
    Verify $E_3 = E_3'$ and $(P_3, Q_3) = (\phi'(P_2), \phi'(Q_2))$, otherwise output reject
else
    $(c, d) \leftarrow \mathsf{resp}$
    $K_{\widehat{\psi}} \leftarrow [c]P_2 + [d]Q_2$
    $K_{\widehat{\psi'}} \leftarrow [c]P_3 + [d]Q_3$
    Check $K_{\widehat{\psi}}$, $K_{\widehat{\psi'}}$ have order $\ell_2^{e_2}$, otherwise output reject
    $\widehat{\psi}, E_0' \leftarrow \mathsf{IsogenyFromKernel}(K_{\widehat{\psi}})$
    $\widehat{\psi}', E_1' \leftarrow \mathsf{IsogenyFromKernel}(K_{\widehat{\psi'}})$
    Check $E_0 = E_0'$ and $E_1 = E_1'$, otherwise output reject
Output accept
```

Figure 2: One iteration of the simple but insecure sigma protocol for SIDH. The public parameters are $pp = (p, \ell_1, \ell_2, e_1, e_2, E_0)$. The public key is E_1, and the corresponding secret isogeny is ϕ.

Theorem 3. *The sigma protocol in Figure 2 for relation*

$$\mathcal{R}_{\mathsf{weakSIDH}} = \{(E_1, \phi) \mid \phi : E_0 \to E_1, \deg \phi = \ell_1^{e_1}\}$$

is correct and 2-special sound. Repeated with κ iterations, it is thus a Proof of Knowledge for $\mathcal{R}_{\mathsf{weakSIDH}}$ with knowledge error $2^{-\kappa}$.

Proof. We prove the properties of Theorem 3 separately below.

Correctness: Following the protocol honestly will result in an accepting transcript. This is clear for the $\mathsf{chall} = 1$ case. For the $\mathsf{chall} = 0$ case, observe that

$$\phi'(K_{\widehat{\psi}}) = \phi'([c]P_2 + [d]Q_2) = [c]P_3 + [d]Q_3 = K_{\widehat{\psi'}},$$

thus $K_{\widehat{\psi'}}$ generates the kernel of $\widehat{\psi}'$.

2-special soundness: Without loss of generality, suppose we obtain two transcripts $(\mathsf{com}, 0, \mathsf{resp})$, $(\mathsf{com}, 1, \mathsf{resp}')$. Then recover $(c, d) \leftarrow \mathsf{resp}$ and $K_{\phi'} \leftarrow \mathsf{resp}'$, and let ϕ' be an isogeny whose kernel is generated by $K_{\phi'}$. Applying Lemma 2, with $(\phi_A, \phi_B, \phi_{AB}) = (\phi', \widehat{\psi}, \widehat{\psi}')$, we obtain an isogeny $\chi : E_0 \to E_1$ of degree $\ell_1^{e_1}$. The conditions of the lemma on the kernels of $\widehat{\psi}$ and $\widehat{\psi}'$ are satisfied because $\phi'(K_{\widehat{\psi}}) = K_{\widehat{\psi'}}$, as above. This shows the protocol is 2-special sound, and that it is a Proof of Knowledge of an isogeny corresponding to the given public key curve. □

5.2 Why this protocol does not prove correctness of the points (P_1, Q_1)

We briefly explain why the protocol in this section does not convince a verifier that $(P_1, Q_1) = (\phi(P_0), \phi(Q_0))$. The first observation is that Figure 2 does not actually use P_1 or Q_1 anywhere, so of course, nothing is proved. But one could tweak the protocol in the $\mathsf{chall} = 0$ case to use the isogenies $\widehat{\psi} : E_2 \to E_0$ and $\widehat{\psi}' : E_3 \to E_1$ to test the points. For example, using the duals of these isogenies, one could compute integers (a, b) such that $\ker(\psi) = \langle [a]P_0 + [b]Q_0 \rangle$ and then test whether or not $\ker(\psi') = \langle [a]P_1 + [b]Q_1 \rangle$.

The problem for the verifier is that this is not enough to deduce that $(P_1, Q_1) = (\phi(P_0), \phi(Q_0))$. For example, a dishonest prover who wants to perform an attack might set $(P_1, Q_1) = (\phi(P_0), \phi(Q_0) + T)$ where T is a point of order ℓ_2. If the prover always uses integers b that are multiples of ℓ_2 (and remember, the prover does choose (a, b)) then this cheating will not be detected by the verifier. Hence, the protocol needs to be changed so that the verifier can tell that the kernels of the isogenies $\widehat{\psi}$ are sufficiently independent across the executions of the protocol. This is the fundamental problem that we solve in Section 6.

5.3 Making the proof zero-knowledge

There is an obvious reason why the protocol is not zero-knowledge: we already noted that it is not sufficient to prove that $P_1 = \phi(P_0)$ and $Q_1 = \phi(Q_0)$, even if we try some minor tweaks. However, a honest prover leaks a random pair $(K_\psi, \phi(K_\psi))$ every time it is challenged with $\mathsf{chall} = 0$. Thus, after less than three iterations on average, it leaks the action of ϕ on the full $E_0[\ell_2^{e_2}]$, and in particular it leaks P_1 and Q_1. This fact was already observed by De Feo, Jao and Plût, who instead sketched a proof of how their protocol is zero-knowledge with respect to the stronger SIDH relation, which includes (P_1, Q_1) in the language (see definition in Section 6).

But there is a second reason why our protocol fails to be zero-knowledge, even with respect to the SIDH relation. When challenged with $\mathsf{chall} = 0$ a simulator can perfectly simulate the isogenies ψ and ψ', however it will not be able to compute the associated ϕ', and thus the correct points (P_3, Q_3). On the other hand, the adversary of Definition 3 knows ϕ, and after seeing ψ and ψ' it can easily compute ϕ' and then P_3 and Q_3, thus unmasking the simulator. We stress this is not an issue limited to SHVZK: any other definition of computational zero-knowledge we are aware of has the protocol fall, in a way or another, in the same trap.

We solve both issues at once by moving to ternary challenges $\{-1, 0, 1\}$, splitting the $\mathsf{chall} = 0$ case into two separate flows: $\mathsf{chall} = -1$ corresponding to revealing ψ, and $\mathsf{chall} = 0$ corresponding to revealing ψ'. However, now the information on E_2, E_3 and the respective torsion bases may not be fully revealed when $\mathsf{chall} \in \{-1, 0\}$: to hide it but still commit to it, we introduce a binding and hiding commitment scheme that we call $\mathsf{C}(x;y)$. We need statistical hiding, so that $\mathsf{C}(\mathsf{com};r)$, where r is a sufficiently long random string, can in principle be a commitment of any of the possible values for com. We also need it to be (computationally) hard for a malicious prover to open $\mathsf{C}(\mathsf{com};r)$ to a different value $(\mathsf{com}';r')$. As an example, we can take $\mathsf{C}(x;y) = H(x\|y)$ where H is a cryptographic hash function and y is considerably longer than the output length of H (e.g., H hashes to n bits and r is $2n$ bits, chosen uniformly at random at the time of the commitment). The resulting scheme is presented in Figure 3.

```
round 1 (commitment)
Run commitment from Figure 2, giving commitment $\mathsf{com}_0 = (E_2, P_2, Q_2, E_3, P_3, Q_3)$
Let $\psi$ be the isogeny from Line 2 of Figure 2
$K_{\widehat{\psi}} \leftarrow \mathsf{DualKernel}(\psi)$
Compute $c, d \in \mathbb{Z}/\ell_2^{e_2}\mathbb{Z}$ such that $K_{\widehat{\psi}} = [c]P_2 + [d]Q_2$ (and $K_{\widehat{\psi'}} = [c]P_3 + [d]Q_3$)
Set $\mathsf{com}_L = (E_2, P_2, Q_2)$ and $\mathsf{com}_R = (E_3, P_3, Q_3)$
Choose random nonces $r_L, r_R, r$
Output $\mathsf{com} \leftarrow (\mathsf{C}_L = \mathsf{C}(\mathsf{com}_L; r_L), \mathsf{C}_R = \mathsf{C}(\mathsf{com}_R; r_R), \mathsf{C} = \mathsf{C}(c, d; r))$.

round 3 (response)
if $\mathsf{chall} = 1$ then
    Let $K_{\phi'}$ be the kernel generator computed at Line 4 of Figure 2
    Output $\mathsf{resp} \leftarrow (\mathsf{com}_L, r_L, K_{\phi'}, \mathsf{com}_R, r_R)$
else
    if $\mathsf{chall} = 0$ then
        Output $\mathsf{resp} \leftarrow (\mathsf{com}_R, r_R, c, d, r)$
    else
        Output $\mathsf{resp} \leftarrow (\mathsf{com}_L, r_L, c, d, r)$

Verification
$(\mathsf{C}_L, \mathsf{C}_R, \mathsf{C}) \leftarrow \mathsf{com}$
if $\mathsf{chall} = 1$ then
    $(\mathsf{com}_L, r_L, K_{\phi'}, \mathsf{com}_R, r_R) \leftarrow \mathsf{resp}$
    Check that the commitments $\mathsf{C}_L$ and $\mathsf{C}_R$ are well-formed, if not output reject
    $\mathsf{com}' \leftarrow (E_2, P_2, Q_2, E_3, P_3, Q_3)$
    Verify $(\mathsf{com}', \mathsf{chall}, K_{\phi'})$ as in Figure 2 verification
    If verification fails, output reject.
else
    $(\mathsf{com}_X, r_X, c, d, r) \leftarrow \mathsf{resp}$
    Check that the commitments $\mathsf{C}$ and $\mathsf{C}_X$ are well-formed, if not output reject
    if $\mathsf{chall} = -1$ then
        $K_{\widehat{\psi}} \leftarrow [c]P_2 + [d]Q_2$
        Check $K_{\widehat{\psi}}$ has order $\ell_2^{e_2}$, otherwise output reject
        $\widehat{\psi}, E_0' \leftarrow \mathsf{IsogenyFromKernel}(K_{\widehat{\psi}})$
        Check $E_0 = E_0'$, otherwise output reject
    else
        $K_{\widehat{\psi'}} \leftarrow [c]P_3 + [d]Q_3$
        Check $K_{\widehat{\psi'}}$ has order $\ell_2^{e_2}$, otherwise output reject
        $\widehat{\psi'}, E_1' \leftarrow \mathsf{IsogenyFromKernel}(K_{\widehat{\psi'}})$
        Check $E_1 = E_1'$, otherwise output reject
Output accept if all the above conditions hold.
```

Figure 3: Sigma protocol to prove the weak SIDH relation $\mathcal{R}_{\mathsf{weakSIDH}}$.

Theorem 4. *For a fixed security parameter κ, a proof consisting of κ iterations of the sigma protocol in Figure 3 is a computationally SHVZK Proof of Knowledge for $\mathcal{R}_{\mathsf{weakSIDH}}$ with knowledge error $(2/3)^\kappa$, assuming the DSSP problem is hard and the commitment scheme $\mathsf{C}()$ is computationally binding and statistically hiding.*

Proof. Because the protocol only adds a few commitments to the protocol in Figure 2, correctness follows immediately from Theorem 3.

Soundness: We prove 3-special soundness by reducing to the 2-special soundness of the simplified protocol. From three transcripts $(\mathsf{com}, -1, \mathsf{resp}_{-1})$, $(\mathsf{com}, 0, \mathsf{resp}_0)$ and $(\mathsf{com}, 1, \mathsf{resp}_1)$, we recover $\mathsf{com}_0 = (E_2, P_2, Q_2, E_3, P_3, Q_3)$, $K_{\phi'}$ and (c, d), like in the simplified protocol. Because C is binding, these values are (computationally) uniquely determined by com, so they must be consistent across the three transcripts. Joining together the verifications of cases $\mathsf{chall} = -1, 0$, we see that the verifier does the exact same computations as in the simplified protocol. Hence, Theorem 3 shows that there exists an isogeny $\chi : E_0 \to E_1$ of degree $\ell_1^{e_1}$, and thus the protocol is sound.

A cheating prover has $1/3$ chances of being caught, as it may prepare commitments in a way that lets it answer any two out of the three challenges. We conclude that the protocol has knowledge error $(2/3)^\kappa$.

Zero-knowledge: We only need to prove that a single execution of the protocol is SHVZK, then SHVZK of κ repetition follows by the the hybrid technique of Goldreich et al. [GMW91]. We define the simulator Sim as follows.

Case $\mathsf{chall} = -1$: Sim follows the honest protocol by choosing a random generator $K_\psi \in E_0[\ell_2^{e_2}]$, then picking $P_2, Q_2 \leftarrow \mathsf{RandomBasis}_2(E_2)$ and computing c, d such that $\ker(\widehat{\psi}) = \langle [c]P_2 + [d]Q_2 \rangle$. It finally commits to $\mathsf{C}_L = \mathsf{C}(E_2, P_2, Q_2; r_L)$ and $\mathsf{C} = \mathsf{C}(c, d; r)$, while taking a uniformly random value for C_R. The responses are the openings to C_L and C, it is clear that this transcript is valid.

Observe that the commitments C_L and C are identical to the honest commitments, thus the only way for Adv to distinguish Sim from a real transcript is to distinguish C_R from a commitment to (E_3, P_3, Q_3), but this is impossible since we assumed that $\mathsf{C}()$ is statistically hiding.

Case $\mathsf{chall} = 0$: This is nearly identical to the previous case. Sim chooses a random kernel generator $K_{\psi'} \in E_1[\ell_2^{e_2}]$, picks a random basis (P_3, Q_3) of $E_3[\ell_2^{e_2}]$, and computes c, d such that $\ker(\widehat{\psi}) = \langle [c]P_3 + [d]Q_3 \rangle$. It then computes the commitments C_R and C like in the honest protocol, and takes a random value for C_L.

We only need to observe that in the honest protocol both $K_{\psi'}$ and (P_3, Q_3) are uniformly random, thus C_R and C are distributed identically to the honest protocol. We conclude again using the fact that $\mathsf{C}()$ is statistically hiding.

Case $\mathsf{chall} = 1$: Sim chooses a random supersingular elliptic curve[2] E_2. It then chooses uniformly a random kernel generator $K_{\phi'} \in E_2$ of order $\ell_1^{e_1}$ and computes the isogeny $\phi' : E_2 \to E_3$. Next, Sim generates a basis $P_2, Q_2 \leftarrow \mathsf{RandomBasis}_2(E_2)$ and computes $P_3, Q_3 \leftarrow \phi'(P_2), \phi'_i(Q_2)$. Finally, it commits to $\mathsf{C}_L = \mathsf{C}(E_2, P_2, Q_2; r_L)$ and $\mathsf{C}_R = \mathsf{C}(E_3, P_3, Q_3; r_R)$, while taking a uniformly random value for C. The responses are the openings to C_L and C_R, it is clear that this transcript is valid.

Like before, because $\mathsf{C}()$ is statistically hiding the adversary cannot use C to gain an advantage in distinguishing Sim. But now the curves E_2 and E_3 and the isogeny ϕ' are not distributed identically to the honest protocol, but rather like in distribution $\mathcal{D}_1$ of the DSSP problem (Definition 7). It is then clear that an adversary that has a non-negligible advantage in distinguishing Sim from the real protocol can be used as a distinguisher for DSSP. □

Remark 1. There are certainly improvements that can be made to improve efficiency and compress the size of signatures, but these are standard and we will not explore them here. For example, in practice the information (E_2, P_2, Q_2) would be replaced with a triplet of x-coordinates, as in SIKE [ACC+17].

6 Correctness of the points in an SIDH public key

Section 5 gave a simple protocol, which can be shown to be a proof of knowledge of a degree $\ell_1^{e_1}$ isogeny from E_0 to E_1. However, an SIDH public key (E_1, P_1, Q_1) also consists of the two torsion points, and these points are the cause of issues such as the adaptive attack [GPST16], as discussed in Section 3. In this section, we show that the choice of points P_1, Q_1 by a malicious prover is severely restricted if they must keep them consistent with "random enough" values of a, b (i.e., random choices of ψ)—preventing adaptive attacks entirely.

[2] One way to do so is to take a random ℓ_2-isogeny walk from E_0. To ensure a distribution close to uniform, we take a walk of length $\gtrsim \log(p) \approx 2e_2$. However a walk of length e_2 is sufficient to get a variant of DSSP that is also believed to be hard.

Fix E_0 and a basis $\{P_0, Q_0\}$ for $E_0[\ell_2^{e_2}]$. We define the strong[3] SIDH relation to be

$$\mathcal{R}_{\mathsf{SIDH}} = \left\{ ((E_1, P_1, Q_1), \phi) \,\middle|\, \begin{array}{c} \phi : E_0 \to E_1,\, \deg \phi = \ell_1^{e_1}, \\ P_1 = \phi(P_0),\, Q_1 = \phi(Q_0) \end{array} \right\}.$$

Figure 4 presents our protocol for proving this strong relation.

This protocol is reminiscent of the one in Section 5 in that it "flips the SIDH square upside down": We view E_2 as the "starting curve" in SIDH, and use the fact that the verifier can check $\widehat{\psi} : E_2 \to E_0$ and $\phi' : E_2 \to E_3$. The verifier also checks that $\ker(\widehat{\psi'}) = \phi'(\ker(\widehat{\psi}))$, and from this the curve E_1 is well-defined and the existence of an isogeny $\phi : E_0 \to E_1$ with $\ker(\phi) = \widehat{\psi}(\ker(\phi'))$ follows.

But this is not enough, since there might be multiple isogenies from E_0 to E_1. The key idea we introduce here is to require pairs of points $R_{1,0}, R_{1,1} = \phi(R_{0,0}), \phi(R_{0,1})$ that are "independent" (in the sense that they generate the full torsion). Hence the action of ϕ on the whole $\ell_2^{e_2}$ torsion is determined. This is why we "double" the protocol. So in each round of our new sigma protocol, we commit to two SIDH squares rather than just one, and require that the kernel generators of ψ in these two squares are independent from each other. We add this independence as an extra check during verification. We also require an assurance that both squares use consistent isogenies ϕ'. For this purpose we use a uniformly random $\ell_1^{e_1}$-torsion basis (U, V) on E_0 and compute the image of this basis on both curves $E_{2,i}$—if both ϕ'_i are the images of ϕ under the vertical isogenies ψ_i, then both should be representable in terms of $(\psi_i(U), \psi_i(V))$ using the same coefficients. These extra checks achieve a 3-special sound protocol for the strong SIDH relation above.

We stress that (U, V) are not made public in the commitment. In the protocol the function $\mathsf{RandomBasis}_1$ is called many times on the same curve E_0 during t rounds of the protocol and it is important that the outputs are independent and not known to the verifier in the $\mathsf{chall} = 1$ case.

Theorem 5. *For a fixed security parameter κ, a proof consisting of κ iterations of the sigma protocol in Figure 4 is a computationally SHVZK Proof of Knowledge for $\mathcal{R}_{\mathsf{SIDH}}$ with knowledge error $(2/3)^\kappa$, assuming the Double-DSSP problem is hard and the commitment scheme $\mathsf{C}()$ is computationally binding and statistically hiding.*

Proof. We prove correctness, soundness, and zero-knowledge individually.

Correctness: The point $R_{0,i}$ will always be an invertible scalar multiple of the point K_ψ used by the prover in the commitment round (in the i-th SIDH square) of the protocol because both K_ψ and $R_{0,i}$ are generators of the kernel of ψ in the i-th SIDH square. Hence, because the honest prover will use commitments such that ψ_0 and ψ_1 are independent, then a_i, b_i necessarily exist such that $a_0 b_1 - a_1 b_0$ is invertible in line 8 of commitment. Also note that because $K_{\phi',i} = [e]U'_i + [f]V'_i = [e]\psi_i(U) + [f]\psi_i(V)$ for both $i \in \{0, 1\}$, and U, V have order coprime to the degree of ψ_i, the checks involving U'_i, V'_i, e, and f will also succeed. Correctness of the rest of the protocol can also be verified in a straightforward way.

Zero-knowledge: We start from the simulator described in Theorem 4, and extend it to simulate the parts of the transcript that are specific to Figure 4, namely the bases U'_i, V'_i and the coefficients c'_i, d'_i, a_i, b_i.

Case $\mathsf{chall} = -1$: For $i = 0, 1$, Sim constructs $K_{\psi_i} \in E_0[\ell_2^{e_2}]$, $P_{2,i}, Q_{2,i}$ and c_i, d_i like in Theorem 4, while ensuring that ψ_0 and ψ_1 are independent.

Additionally, Sim samples $U, V \leftarrow \mathsf{RandomBasis}_1(E_0)$ and computes $U'_i = \psi_i(U)$ and $V'_i = \psi_i(V)$. Then it takes c'_i, d'_i such that $c'_i d_i - d'_i c_i$ is invertible and computes $R_{0,i}, a_i, b_i$ like in the honest protocol. Finally it computes all commitments like in the honest protocol, except for C_R^i which are taken at random.

It is clear that the distribution of $U'_i, V'_i, c'_i, d'_i, a_i, b_i$ is identical to the honest protocol, thus this simulation is indistiguishable following the same argument as in Theorem 4.

[3] The word "strong" here indicates that we confirm not only the correctness of the degree of the isogeny, but the correct images of points.

round 1 (commitment)

```
Run **commitment** from Figure 3, giving commitment $\mathsf{com}^0 = (\mathsf{C}^0_L = \mathsf{C}(\mathsf{com}^0_L; r^0_L), \mathsf{C}^0_R = \mathsf{C}(\mathsf{com}^0_R; r^0_R), \mathsf{C}^0 = \mathsf{C}(c_0, d_0; r^0))$.
Let $\psi_0$ be the isogeny from Line 2 of Figure 3
Run **commitment** from Figure 3 again, subject to one extra condition:
    • If $\psi_1$ is the isogeny from Line 2 of Figure 3, then $\psi_0$ and $\psi_1$ must be independent. Otherwise repeat the commitment phase.
Let $\mathsf{com}^1 = (\mathsf{C}^1_L = \mathsf{C}(\mathsf{com}^1_L; r^1_L), \mathsf{C}^1_R = \mathsf{C}(\mathsf{com}^1_R; r^1_R), \mathsf{C}^1 = \mathsf{C}(c_1, d_1; r^1))$ be the commitment returned by this execution.
$U, V \leftarrow \mathsf{RandomBasis}_1(E_0)$
**for** $i \in \{0, 1\}$ **do**
    Choose $c'_i, d'_i \in \mathbb{Z}/\ell_2^{e_2}\mathbb{Z}$ such that $c'_i d_i - d'_i c_i$ is invertible modulo $\ell_2^{e_2}$
    Set $R_{0,i} \leftarrow \widehat{\psi_i}([c'_i]P_{2,i} + [d'_i]Q_{2,i})$ and $R_{1,i} \leftarrow \widehat{\psi'_i}([c'_i]P_{3,i} + [d'_i]Q_{3,i})$
    Compute $a_i, b_i \in \mathbb{Z}/\ell_2^{e_2}\mathbb{Z}$ such that, simultaneously, $R_{0,i} = [a_i]P_0 + [b_i]Q_0$ and $R_{1,i} = [a_i]P_1 + [b_i]Q_1$
    Let $U'_i = \psi_i(U)$ and $V'_i = \psi_i(V)$
Choose random nonces $r^0_m, r^1_m$
Output $\mathsf{com}_i \leftarrow (U'_i, V'_i, \mathsf{C}^i_L, \mathsf{C}^i_R, \mathsf{C}^i, \mathsf{C}^i_m = \mathsf{C}(c'_i, d'_i, a_i, b_i; r^i_m))$ for $i \in \{0, 1\}$.
```

round 3 (response)

```
**if** $\mathsf{chall} = 1$ **then**
    Write $K_\phi = [e]U + [f]V$ for $e, f \in \mathbb{Z}/\ell_1^{e_1}\mathbb{Z}$
    Output $\mathsf{resp} \leftarrow ((e, f), \mathsf{com}^0_L, r^0_L, \mathsf{com}^1_L, r^1_L, \mathsf{com}^0_R, r^0_R, \mathsf{com}^1_R, r^1_R)$
**else**
    **if** $\mathsf{chall} = 0$ **then**
        Output $\mathsf{resp} \leftarrow (\mathsf{com}^0_R, r^0_R, \mathsf{com}^1_R, r^1_R, c_0, d_0, r^0, c_1, d_1, r^1, c'_0, d'_0, a_0, b_0, r^0_m, c'_1, d'_1, a_1, b_1, r^1_m)$
    **else**
        Output $\mathsf{resp} \leftarrow (\mathsf{com}^0_L, r^0_L, \mathsf{com}^1_L, r^1_L, c_0, d_0, r^0, c_1, d_1, r^1, c'_0, d'_0, a_0, b_0, r^0_m, c'_1, d'_1, a_1, b_1, r^1_m)$
```

Verification

```
$(U'_0, V'_0, \mathsf{C}^0_L, \mathsf{C}^0_R, \mathsf{C}^0, \mathsf{C}^0_m), (U'_1, V'_1, \mathsf{C}^1_L, \mathsf{C}^1_R, \mathsf{C}^1, \mathsf{C}^1_m) \leftarrow \mathsf{com}^0, \mathsf{com}^1$
**if** $\mathsf{chall} = 1$ **then**
    $((e, f), \mathsf{com}^0_L, r^0_L, \mathsf{com}^1_L, r^1_L, \mathsf{com}^0_R, r^0_R, \mathsf{com}^1_R, r^1_R) \leftarrow \mathsf{resp}$
    **for** $i \in \{0, 1\}$ **do**
        $\mathsf{com}'_i \leftarrow (\mathsf{C}^i_L, \mathsf{C}^i_R, \mathsf{C}^i)$
        Compute $K_{\phi'_i} = [e]U'_i + [f]V'_i$
        $\mathsf{resp}'_i \leftarrow (\mathsf{com}^i_L, r^i_L, K_{\phi'_i}, \mathsf{com}^i_R, r^i_R)$
        Verify $(\mathsf{com}'_i, \mathsf{chall}, \mathsf{resp}'_i)$ as in Figure 3 **verification**
        If verification fails, output $\mathsf{reject}$.
**else**
    $(\mathsf{com}^0_X, r^0_X, \mathsf{com}^1_X, r^1_X, c_0, d_0, r^0, c_1, d_1, r^1, c'_0, d'_0, a_0, b_0, r^0_m, c'_1, d'_1, a_1, b_1, r^1_m) \leftarrow \mathsf{resp}$
    **for** $i \in \{0, 1\}$ **do**
        $\mathsf{com}'_i \leftarrow (\mathsf{C}^i_L, \mathsf{C}^i_R, \mathsf{C}^i)$
        $\mathsf{resp}'_i \leftarrow (\mathsf{com}^i_X, r^i_X, c_i, d_i, r^i)$
        Verify $(\mathsf{com}'_i, \mathsf{chall}, \mathsf{resp}'_i)$ as in Figure 3 **verification**
        **if** $\mathsf{chall} = -1$ **then**
            $R_{0,i} \leftarrow \widehat{\psi_i}([c'_i]P_{2,i} + [d'_i]Q_{2,i})$
            Check $R_{0,i} = [a_i]P_0 + [b_i]Q_0$, otherwise output $\mathsf{reject}$
        **else**
            $R_{1,i} \leftarrow \widehat{\psi'_i}([c'_i]P_{3,i} + [d'_i]Q_{3,i})$
            Check $R_{1,i} = [a_i]P_1 + [b_i]Q_1$, otherwise output $\mathsf{reject}$
    If $\mathsf{chall} = -1$ check $\widehat{\psi}_0(U'_0) = \widehat{\psi}_1(U'_1)$ and $\widehat{\psi}_0(V'_0) = \widehat{\psi}_1(V'_1)$, otherwise output $\mathsf{reject}$.
    Check that $a_0 b_1 - a_1 b_0$ is invertible modulo $\ell_2^{e_2}$, otherwise output $\mathsf{reject}$.
Output $\mathsf{accept}$ if all the above conditions hold.
```

Figure 4: Sigma protocol to prove the strong SIDH relation $\mathcal{R}_{\mathsf{SIDH}}$.

<u>Case $\mathsf{chall} = 0$:</u> This case is similar to the previous one, however Sim needs to compute both ψ_i and ψ'_i in order to simulate U'_i, V'_i. Because the image points $P_1 = \phi(P_0)$ and $Q_1 = \phi(Q_0)$ are part of the SIDH relation, Sim can choose $a', b' \in \mathbb{Z}/\ell_2^{e_2}\mathbb{Z}$ and compute $K_{\psi_i} = [a']P_0 + [b']Q_0$ and $K_{\psi'_i} = [a']P_1 + [b']Q_1$.

It then proceeds like in Theorem 4, but also computes U, V, U'_i, V'_i as above using the knowledge of ϕ_i. After taking c'_i, d'_i with the usual condition, it computes $R_{1,i}$ and then a_i, b_i. Finally, it computes all commitments honestly, except for C^i_L. Again, the simulation is perfect except for C^i_L, and is thus indistinguishable thanks to the hiding property of $\mathsf{C}()$.

<u>Case $\mathsf{chall} = 1$:</u> The simulator twice chooses a random supersingular elliptic curve $E_{2,i}$ for $i \in \{0, 1\}$.

The simulator then chooses uniformly a random point $K_{\phi'_0} \in E_{2,0}$ of order $\ell_1^{e_1}$ and computes the isogeny $\phi'_0 : E_{2,0} \to E_{3,0}$ with kernel $K_{\phi'_0}$. Sim chooses a random basis $\{U'_0, V'_0\}$ for $E_{2,0}[\ell_1^{e_1}]$, and writes $K_{\phi'_0} = [e]U'_0 + [f]V'_0$ for integers e, f.

Next, Sim will randomly generate a basis $\{U'_1, V'_1\}$ of the $\ell_1^{e_1}$-torsion subgroup on $E_{2,1}$, such that $e_{\ell_1^{e_1}}(U'_1, V'_1) = e_{\ell_1^{e_1}}(U'_0, V'_0)$. It sets $K_{\phi'_1} = [e]U'_1 + [f]V'_1$ and lets $\phi'_1 : E_{2,1} \to E_{3,i}$ be an isogeny with kernel generated by $K_{\phi'_1}$.

Next, the simulator generates basis $P_{2,i}, Q_{2,i} \leftarrow \mathsf{RandomBasis}_2(E_{2,i})$, computes $P_{3,i}, Q_{3,i} \leftarrow \phi'_i(P_{2,i}), \phi'_i(Q_{2,i})$. Finally, Sim chooses random values for the commitments $\mathsf{C}^i, \mathsf{C}^i_m$, which will never be opened when $\mathsf{chall} = 1$.

Like in Theorem 4, this is not a perfect simulation of the honest protocol. However, thanks to the hiding property of $\mathsf{C}()$, the adversary is reduced to solve precisely an instance of the Double-DSSP problem (Definition 8).[4]

3-special soundness: Suppose we obtain three accepting transcripts $(\mathsf{com}, -1, \mathsf{resp}_{-1})$, $(\mathsf{com}, 0, \mathsf{resp}_0)$, and $(\mathsf{com}, 1, \mathsf{resp}_1)$. The secret isogeny corresponding to the public key $X = (E_1, P_1, Q_1)$ can be recovered as follows, hence we can extract a valid witness W for the statement X such that $(X, W) \in \mathcal{R}_{\mathsf{SIDH}}$.

Consider just one of the isogeny squares (e.g., $i = 0$). We have (c, d) which defines a point $K_{\widehat{\psi}} = [c]P_{2,0} + [d]Q_{2,0}$ and hence an isogeny $\widehat{\psi} : E_{2,0} \to E_0$. We also have $K_{\phi'} \in E_{2,0}$ which defines an isogeny $\phi' : E_{2,0} \to E_{3,0}$ whose kernel is generated by $K_{\phi'}$. Applying Lemma 2, with $(\phi_A, \phi_B, \phi_{AB}) = (\phi', \widehat{\psi}, \widehat{\psi}')$, we obtain an isogeny $\phi_0 : E_0 \to E_1$ of degree $\ell_1^{e_1}$. The conditions of the lemma on the kernels of $\widehat{\psi}$ and $\widehat{\psi}'$ are satisfied because $\phi'(K_{\widehat{\psi}}) = K_{\widehat{\psi}'}$, as above. Hence we have extracted an isogeny as required.

Repeating the argument for $i = 1$ provides another isogeny $\phi_1 : E_0 \to E_1$ of degree $\ell_1^{e_1}$. The next step is to prove that these isogenies are equivalent (i.e., have the same kernel). This is where the points U_0, V_0, U_1, V_1 are needed. We have

$$\begin{aligned}\ker(\phi_0) &= \widehat{\psi}_0(\ker(\phi'_0))\\ &= \langle \widehat{\psi}_0([e]U_0 + [f]V_0)\rangle\\ &= \langle \widehat{\psi}_1([e]U_1 + [f]V_1)\rangle\\ &= \widehat{\psi}_1(\ker(\phi'_1)) \quad = \ker(\phi_1).\end{aligned}$$

Therefore, we recover the same[5] isogeny $\phi_0 = \phi_1 = \phi$ from both squares.

It remains to prove that the isogeny ϕ we have extracted does map (P_0, Q_0) to (P_1, Q_1) and so is a correct witness.

Recall we are provided with points $R_{j,i}$ and integers a_i, b_i such that $R_{0,i} = [a_i]P_0 + [b_i]Q_0$. Define

$$B = \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \end{pmatrix}.$$

[4] Note that the second pairing condition in Definition 8 is equivalent to the existence of K_0, K_1 such that $K_0 = [e]U'_0 + [f]V'_0$ and $K_1 = [e]U'_1 + [f]V'_1$.

[5] They could differ by an automorphism, but this does not matter. Fix one of them and call it ϕ.

Since B is invertible, $\langle R_{0,0}, R_{0,1}\rangle$ is another basis for $\langle P_0, Q_0\rangle = E_0[\ell_2^{e_2}]$. Recall that $R_{0,i} = \widehat{\psi}_i([c_i']P_{2,i} + [d_i']Q_{2,i})$, $R_{1,i} = \widehat{\psi}_i'([c_i']P_{3,i} + [d_i']Q_{3,i})$, and $P_{3_i}, Q_{3,i} = \phi'(P_{2,i}), \phi'(Q_{2,i})$. It follows from $\phi \circ \widehat{\psi}_i = \widehat{\psi}_i' \circ \phi'$ that $\phi(R_{0,i}) = R_{1,i}$. Hence we have

$$\begin{pmatrix} R_{0,0} \\ R_{0,1} \end{pmatrix} = B \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix}$$

$$\begin{pmatrix} R_{1,0} \\ R_{1,1} \end{pmatrix} = \begin{pmatrix} \phi(R_{0,0}) \\ \phi(R_{0,1}) \end{pmatrix} = B \begin{pmatrix} \phi(P_0) \\ \phi(Q_0) \end{pmatrix}$$

$$\begin{pmatrix} R_{1,0} \\ R_{1,1} \end{pmatrix} = B \begin{pmatrix} P_1 \\ Q_1 \end{pmatrix},$$

therefore

$$B \begin{pmatrix} \phi(P_0) \\ \phi(Q_0) \end{pmatrix} = B \begin{pmatrix} P_1 \\ Q_1 \end{pmatrix},$$

and since B is invertible, we must have that $P_1 = \phi(P_0)$ and $Q_1 = \phi(Q_0)$, as required. □

Note that the protocol in Figure 4 runs the previous protocol (in Figure 3) twice, hence the transcripts produced by this Proof of Knowledge for $\mathcal{R}_{\mathsf{SIDH}}$ will be (at least) twice the size. We expect that improvements to the efficiency and size of the scheme are possible with more analysis, but leave this for future work.

Remark 2. Ghantous et al. [GPV21] discuss issues with extraction of a witness in two different scenarios. Their first scenario ("single collision") involves two distinct isogenies $\phi' : E_2 \to E_3$ in the SIDH square of the identification scheme. Neither of our new identification schemes are impacted by such collisions because the provision of points $P_3, Q_3 \in E_3$ uniquely determines the isogeny ϕ', as shown by Martindale and Panny [MP19]. Their second scenario ("double collision") involves two distinct (non-equivalent) isogenies $\phi, \tilde{\phi} : E_0 \to E_1$, both of degree $\ell_1^{e_1}$ and a point $R \in E_0$ such that

$$E_1/\langle\phi(R)\rangle \cong E_1/\langle\tilde{\phi}(R)\rangle.$$

Our second protocol, for the relation $\mathcal{R}_{\mathsf{SIDH}}$, ensures that the witness extracted is a valid witness for the public key used (including the torsion points). Hence, this second collision scenario does not have any impact on the soundness of our protocol either.

7 SIDH signatures and Non-Interactive Proof of Knowledge

We conclude with some brief remarks about the use of the new protocol proposed above.

It is standard to construct a non-interactive proof of knowledge from an interactive protocol using the Fiat-Shamir transformation (secure in the random oracle model). This works by making the challenge chall for the t rounds of the ID scheme a random-oracle output from input the commitment com and a message M. That is, for message M,

$$V_1^{\mathcal{O}}(\mathsf{com}) = \mathcal{O}(\mathsf{com} \parallel M).$$

In some situations one should include the instance $(E_0, P_0, Q_0, E_1, P_1, Q_1)$ in the hash too. Thus the prover does not need to interact with a verifier and can compute a non-interactive transcript. Because the sigma protocol described in the preceding sections not only proves knowledge of the secret isogeny between two curves, but also correctness of the torsion points in the public key, we obtain a non-interactive proof of knowledge of the secret key corresponding to a given SIDH public key, which proves that the SIDH public key is well-formed. This provides protection against adaptive attacks.

Such a NIZK of an SIDH secret key can, among other applications, be used to achieve a secure non-interactive key exchange scheme based on SIDH.

Currently the only other method known to get a NIKE from SIDH is the k-SIDH proposal by Azarderakhsh, Jao and Leonardi [AJL17]. This requires both parties to publish k SIDH keys and to compute $O(k^2)$ shared SIDH keys, and so requires k^2 isogeny computations to construct the shared key. It is known [DGL$^+$20, BKM$^+$20] that one can attack the scheme in $\tilde{O}(16^k)$ oracle queries and time. For a given security parameter λ it is therefore natural to suppose k grows linearly in λ, in which case the complexity of the protocol grows quadratically in λ. In contrast, the soundess of our NIZK protocol means the number of rounds grows linearly in λ, and the key exchange protocol itself is a single SIDH exchange. So asymptotically the cost of our scheme will be less than k-SIDH.

8 Conclusions

We have shown a counterexample to the soundness of the De Feo–Jao–Plût sigma protocol. We have described a new sigma protocol that addresses this issue, and also allows to prove that an SIDH key is correctly generated. Our protocol also solves the soundness issue raised by Ghantous, Pintore and Veroni.

The problem of proving correctness of an isogeny turns out to be considerably more difficult than was anticipated (at least, by us!), and there are several open problems for future work. First it would be good to have a protocol with 2-special-soundness for the SIDH relation. The 3-special-soundness and ternary challenges seem to be necessary for the weak SIDH relation, but are relied upon in Section 6 only to bypass the difficulty in simulating the torsion bases (P_2, Q_2) and (P_3, Q_3). Hence, a protocol with statistical zero-knowledge instead of computational zero-knowledge would help with this issue. Second, the protocol seems extremely complex and it would be wonderful to have a simpler and more elegant one.

We have not considered ways to make the protocol more compact. There are some trivial modifications that would reduce the communication (such as replacing pairs (c_i, d_i) with projective points $(c_i : d_i)$) and there is scope for more sophisticated compression of the protocol messages. However, we feel that progress at the conceptual level to reduce the communication cost is more relevant than applying standard implementation tricks.

References

[ACC$^+$17] Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, David Jao, Brian Koziel, Brian LaMacchia, Patrick Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017.

[AJL17] Reza Azarderakhsh, David Jao, and Christopher Leonardi. Post-quantum static-static key agreement using multiple protocol instances. In *International Conference on Selected Areas in Cryptography*, pages 45–63. Springer, 2017.

[BKM$^+$20] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Charlotte Weitkämper. On adaptive attacks against Jao-Urbanik's isogeny-based protocol. In *Progress in Cryptology - AFRICACRYPT 2020*, pages 195–213, Cham, 2020. Springer International Publishing.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[Dam00] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 418–430. Springer, Heidelberg, May 2000.

[DGL$^+$20] Samuel Dobson, Steven D. Galbraith, Jason LeGrow, Yan Bo Ti, and Lukas Zobernig. An adaptive attack on 2-SIDH. *International Journal of Computer Mathematics: Computer Systems Theory*, 5(4):282–299, 2020.

[DGOW95] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs dishonest verifier in public coin zero-knowledge proofs. In Don Coppersmith, editor, *CRYPTO '95*, volume 963 of *LNCS*, pages 325–338. Springer, Heidelberg, August 1995.

[DJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

[FP21] Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on SIDH. Cryptology ePrint Archive, Report 2021/1322, 2021. `https://ia.cr/2021/1322`.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[GMY06] Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. *Journal of Cryptology*, 19(2):169–209, April 2006.

[GPS20] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1):130–175, 2020.

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology – ASIACRYPT 2016*, pages 63–91. Springer Berlin Heidelberg, 2016.

[GPV21] Wissam Ghantous, Federico Pintore, and Mattia Veroni. Collisions in supersingular isogeny graphs and the SIDH-based identification protocol. Cryptology ePrint Archive, Report 2021/1051, 2021. `https://eprint.iacr.org/2021/1051`.

[GV18] Steven D. Galbraith and Frederik Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, 17(10):1–22, 2018.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.

[JD11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography*, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[JS14] David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In *PQCrypto 2014*, volume 8772 of *Lecture Notes in Computer Science*, pages 160–179. Springer, 2014.

[Leo20] Christopher Leonardi. A note on the ending elliptic curve in SIDH. Cryptology ePrint Archive, Report 2020/262, 2020. `https://ia.cr/2020/262`.

[MP19] Chloe Martindale and Lorenz Panny. How to not break SIDH. CFAIL, 2019. `https://ia.cr/2019/558`.

[Sil09] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, Dordrecht, second edition, 2009.

[Tho17] Erik Thormarker. *Post-Quantum Cryptography: Supersingular Isogeny Diffie-Hellman Key Exchange*. Thesis, Stockholm University, 2017.

[UJ18] David Urbanik and David Jao. SoK: The problem landscape of SIDH. In *Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop*, pages 53–60, 2018.

[UJ20] David Urbanik and David Jao. New techniques for SIDH-based NIKE. *Journal of Mathematical Cryptology*, 14(1):120–128, 2020.

[UXT+22] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 296–322, 2022.

[Vél71] Jacques Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.

[YAJ+17] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *International Conference on Financial Cryptography and Data Security*, pages 163–181. Springer, 2017.