Proving knowledge of isogenies – A survey

Ward Beullens[1], Luca De Feo[1], Steven D. Galbraith[2], and Christophe Petit[3]

[1] IBM Research Europe, Zürich, Switzerland. wbe@zurich.ibm.com, des.cod.crypt.2022@defeo.lu
[2] University of Auckland, New Zealand. s.galbraith@auckland.ac.nz
[3] University of Birmingham and Université libre de Bruxelles. christophe.f.petit@gmail.com

Abstract. Isogeny-based cryptography is an active area of research in post-quantum public key cryptography. The problem of proving knowledge of an isogeny is a natural problem that has several applications in isogeny-based cryptography, such as allowing users to demonstrate that they are behaving honestly in a protocol. It is also related to isogeny-based digital signatures. Over the last few years, there have been a number of advances in this area, but there are still many open problems. This paper aims to give an overview of the topic and highlight some open problems and directions for future research.

Keywords: isogeny, post-quantum cryptography, zero-knowledge

1 Introduction

Let E_0 and E_1 be elliptic curves. An isogeny is a mapping $\phi : E_0 \to E_1$ (see Section 2.1 for technical background). In particular, an isogeny is a group homomorphism. If there is an isogeny $\phi : E_0 \to E_1$ then there is also an isogeny $\hat{\phi} : E_1 \to E_0$, called the dual isogeny. Two curves are called isogenous if there is an isogeny between them. Given two isogenous elliptic curves E_0 and E_1 over a finite field $\mathbb{F}_q$ it is believed to be computationally hard (even for quantum computers) to compute an isogeny between them. A natural problem is therefore for an entity to prove that they know an isogeny $\phi : E_0 \to E_1$ between two public elliptic curves E_0 and E_1 without revealing the isogeny.

Tate's isogeny theorem (due to Deuring in the elliptic curve case) states that two elliptic curves E_0 and E_1 over a finite field $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if $\#E_0(\mathbb{F}_q) = \#E_1(\mathbb{F}_q)$. Since there are polynomial time algorithms to compute $\#E_0(\mathbb{F}_q)$ it follows that one can determine in polynomial time whether two curves are isogenous. However, this is not the end of the story, since for cryptographic applications one often wants to prove knowledge of an isogeny between E_0 and E_1, possibly with some additional restrictions, e.g., on its degree.

Indeed, there are (at least) six relations that have been studied in the setting of isogeny-based cryptography. These relations are defined as sets of pairs (x, w) where x is the *statement* and w is a *witness*. Each relation defines a *language* $L = \{x : \exists w, (x, w) \in \mathcal{R}\}$. For simplicity, we define the relations for a fixed field $\mathbb{F}_q$. In each of the relations below

E_0, E_1 denote elliptic curves (usually supersingular in this paper) and ϕ denotes an isogeny (the witness), all defined over $\mathbb{F}_q$. We implicitly assume that witnesses can be represented in polynomial space and evaluated in polynomial time.

$$\begin{aligned}
\mathcal{R}_{\text{triv}} &= \big\{\big((E_0, E_1), \perp\big) \,\big|\, \#E_0(\mathbb{F}_q) = \#E_1(\mathbb{F}_q), \text{ so } E_0, E_1 \text{ isogenous over } \mathbb{F}_q\big\};\\
\mathcal{R}_{\text{isog}} &= \big\{\big((E_0, E_1), \phi\big) \,\big|\, \phi : E_0 \to E_1 \text{ is an arbitrary isogeny}\big\};\\
\mathcal{R}_{\text{deg}} &= \big\{\big((E_0, E_1, d), \phi\big) \,\big|\, \phi : E_0 \to E_1 \text{ is an isogeny of degree } d\big\};\\
\mathcal{R}_{\text{SIDH}} &= \big\{\big((E_0, E_1, d, D, P_0, Q_0, P_1, Q_1), \phi\big)\\
&\qquad \big|\, \phi : E_0 \to E_1 \text{ is an isogeny of degree } d\\
&\qquad \text{ and } P_1 = \phi(P_0), Q_1 = \phi(Q_0)\\
&\qquad \text{ where } E_0[D] = \langle P_0, Q_0\rangle \text{ and } \gcd(D, d) = 1\big\};\\
\mathcal{R}_{\text{M}-\text{SIDH}} &= \big\{\big((E_0, E_1, d, D, P_0, Q_0, P_1, Q_1), \phi\big)\\
&\qquad \big|\, \phi : E_0 \to E_1 \text{ is an isogeny of degree } d\\
&\qquad \text{ and } P_1 = \lambda\phi(P_0), Q_1 = \lambda\phi(Q_0)\\
&\qquad \text{ and } \lambda^2 = 1 \mod D\\
&\qquad \text{ where } E_0[D] = \langle P_0, Q_0\rangle \text{ and } \gcd(D, d) = 1\big\};\\
\mathcal{R}_{\text{CSIDH}} &= \big\{\big((E_0, E_1), \phi\big) \big|\, E_0, E_1 \text{ supersingular},\\
&\qquad j(E_0), j(E_1) \in \mathbb{F}_p,\ \phi \text{ defined over } \mathbb{F}_p\big\}\\
&\qquad \text{See Definition 3 in Section 2.4.}
\end{aligned}$$

As we just said, the language (E_0, E_1) of isogenous curves can be decided in polynomial time, hence the trivial witness $\perp$ in $\mathcal{R}_{\text{triv}}$. The relations $\mathcal{R}_{\text{isog}}$ and $\mathcal{R}_{\text{deg}}$ are the main focus of this survey. These two relations are believed to be hard in the sense that there is no efficient algorithm known to compute a witness when given an arbitrary element of the language. For $\mathcal{R}_{\text{deg}}$ even deciding the language is usually hard (depending on d).

The relation $\mathcal{R}_{\text{SIDH}}$ is no longer relevant due to the devastating attacks discovered in 2022 by Castryck and Decru [CD23], Maino and Martindale [MMP$^+$23] and Robert [Rob23]. However, we briefly mention SIDH in a few places for historical reasons, and since some aspects of it are used to build protocols for the other relations. The relation $\mathcal{R}_{\text{M}-\text{SIDH}}$ is equivalent to $\mathcal{R}_{\text{SIDH}}$ when D is a power of a prime, indeed in this case there are only two or four possibilities for the square root of unity λ. When D contains many prime factors, however $\mathcal{R}_{\text{M}-\text{SIDH}}$ is believed to be harder than $\mathcal{R}_{\text{SIDH}}$, and has been proposed as an alternative foundation for an SIDH-like key exchange [FMP23].

Motivation. The main motivation for zero-knowledge proofs of knowledge of an isogeny is in ensuring that public keys in a system are correctly formed and that the owner of a public key does know the corresponding private key. Some systems require such "proof of possession" checks when a user registers their public key, to prevent malicious behaviour such as a user registering another user's public key as their own. For discussion see Section 4.3 of the X.509 RFC [AFKM05], and [BFPW07].

Zero-knowledge proofs are also useful to defeat active attacks in cryptographic protocols. A passively secure protocol can indeed be turned into an actively secure one by requiring all parties to accompany each message they send with a zero-knowledge proof that the message has been generated correctly as specified by the protocol. A recent example of this is given in [BCC$^+$23]. An earlier example of this (now obsolete) is that the SIDH protocol [DFJP14] is vulnerable to the *GPST attack* [GPST16], in which an attacker deviates from the protocol to progressively learn the secret key of its victim. In the key encapsulation mechanism SIKE based on SIDH [JAC$^+$17], this attack was defeated using a variant of the Fujisaki-Okamoto transform, but this involved generating and communicating ephemeral keys. In contrast, a solution based on a non-interactive proof of knowledge would have allowed static keys and non-interactive key exchange.

Proofs of knowledge have other applications as well, most notably they give digital signature schemes through the Fiat-Shamir transform and variants. The RSA and ECDSA signature schemes currently in use are based on the hardness of integer factoring and the elliptic curve discrete logarithm problem, which will be solved efficiently when large-scale quantum computers are available. It is important to develop new signature schemes with post-quantum security. Isogeny-based cryptography is believed to resist quantum computers, hence it is a natural and important question to build efficient digital signatures from isogeny problems. A natural way to build those signatures is to first develop zero-knowledge proofs of knowledge for isogeny relations.

A simple approach that does not work. To introduce some of the challenges in developing zero-knowledge proofs for the relations above, we first describe a simple idea for the second relation, namely to adapt the classic *Goldreich, Micali, Wigderson (GMW)* zero-knowledge proof of graph isomorphism [GMW91] (see Section 3.3). We then explain why a straightforward adaptation does not work.

Let $\phi : E_0 \to E_1$ be an $\mathbb{F}_q$-isogeny, which is the witness known to the prover. The natural idea is to choose a random isogeny $\psi : E_1 \to E_2$. Due to the expansion properties of isogeny graphs (discussed in Section 2.2), if ψ corresponds to a long enough walk in the graph then E_2 is uniformly distributed in the set of supersingular curves. Set the commitment to be E_2. The challenger sends a challenge $\mathsf{chall} \in \{0, 1\}$. When $\mathsf{chall} = 0$ the prover responds with a description of ψ, and when $\mathsf{chall} = 1$ the prover responds with $\rho := \psi \circ \phi$. The verifier checks that the response is an isogeny from $E_{1-\mathsf{chall}}$ to E_2, and accepts if this is the case.

One might think this could be zero-knowledge since E_2 is distributed uniformly and so can be simulated without knowing the witness. But the problem is that $\psi \circ \phi$ can leak the witness ϕ. The details depend on how the isogeny $\psi \circ \phi$ is represented (see Section 2.1). If it is represented as a sequence of j-invariants, then the receiver will note that $j(E_1)$ appears on the list and so the sequence of j-invariants up to that point represents the isogeny ϕ. If it is represented using a kernel point (or a set of generators for the kernel) then the leakage depends on whether $\deg(\phi)$ is known (or guessable) and how this relates to $\deg(\psi)$. In any case, because the composition factors through E_1 we have $\ker(\phi) \subseteq \ker(\rho)$.

Solutions to the zero-knowledge issue. While the above simple approach fails as such, it still underlies all proofs of knowledge of an isogeny, with the zero-knowledge issue solved in the following ways:

- In CSIDH-based protocols as well as in GPS and SQISign signatures, the response isogeny in the above sketch is replaced by a canonical or randomized isogeny in the same class (i.e. connecting the same two curves). This is done efficiently using the KLPT algorithm in the GPS case [GPS20], after a big precomputation in CSI-FiSh [BKV19], or using blinding and rejection sampling in SeaSign [DFG19]. On the other hand, SQISign [DFKL$^+$20] relies on an ad-hoc computational assumption to guarantee that the isogeny returned does not leak information on the secret.
- In SIDH-like protocols, the triple of isogenies in the above sketch is replaced by a four-tuple corresponding to an SIDH commutative diagram, with the secret isogeny being one of the four edges (see Section 5.1). The response isogeny is made of either the only "parallel" isogeny in this commutative diagram, or one of the two "orthogonal" ones. Zero-knowledge relies on an ad hoc computational assumption, essentially stating that "parallel" pairs of isogenies are indistinguishable from random pairs with the same degrees. However, we briefly mention in Section 5.3 some recent work that allows statistical zero-knowledge.

Note that we will later introduce CSIDH-based protocols separately from GPS and SQISign because the two sets of protocols differ significantly in the mathematical machinery that they use.

Soundness. In addition to the zero-knowledge issue, the simple protocol above only offers limited soundness guarantees, as a cheating prover can correctly predict the challenge bit with a probability of one out of two. A simple solution to this is iterating the protocol; this however brings a large efficiency cost, both in proof/signature sizes and in computation times. One of the biggest open problems in this area is to develop more efficient protocols by lowering the probability of successful cheating (the *soundness error*) by a user who does not know a witness. In SeaSign and CSI-FiSh signatures those costs can be traded for larger key sizes by relying on multiple instances of the basic isogeny problem. Currently, the most efficient scheme is the SQISign protocol. It uses a large challenge space, and may therefore only be run once, resulting in very short proofs/signatures.

Outline. Our paper aims to explore these topics in detail and to list some open problems and directions for future research. Our focus is exactly on the issues raised above: How to achieve (and prove) soundness; what computational assumptions are required for zero-knowledge; how to get more efficient systems.

We provide background on isogeny-based cryptography and zero-knowledge proofs of knowledge in Sections 2 and 3 respectively. We then describe proofs based on group actions (SeaSign, CSI-FiSh) in Section 4, proofs for $\mathcal{R}_{\mathrm{deg}}$ based on ideas from SIDH in Section 5, and finally GPS and SQISign in Section 6. Specifically, we discuss proofs for the relation $\mathcal{R}_{\mathrm{isog}}$ in Section 5.1 and Section 6.1, and for the relation $\mathcal{R}_{\mathrm{deg}}$ in Section 5.2. Proofs for the CSIDH relation $\mathcal{R}_{\mathrm{CSIDH}}$ are discussed in Section 4. We conclude the paper and list open problems in Section 7.

Ackowledgements. Steven Galbraith is funded by NZ Government MBIE Catalyst Fund UOAX1933. Luca De Feo was supported by SNSF grant TMCG-2_213766, CryptonIs. Christophe Petit was supported by EPSRC award EP/V011324/1. Ward Beullens holds a Junior Post-Doctoral fellowship 1S95620N from the Research Foundation Flanders (FWO).

2 Isogeny-based cryptography background

2.1 Elliptic curves and isogenies

An elliptic curve over a field $\mathbb{F}_q$ is a non-singular projective algebraic curve of genus 1 with a designated point that we call O. A typical example is the projective closure of the affine Montgomery model $y^2 = x(x^2 + Ax + 1)$, where the point O is the point at infinity. An elliptic curve is defined up to isomorphism by its j-invariant.

An *isogeny* $\phi : E_1 \to E_2$ (see Chapter 12 of Washington [Was08] or Section III.4 of Silverman [Sil09]) is a morphism and has finite kernel. Given a finite subgroup $G \subseteq E_1(\overline{\mathbb{F}}_q)$ there is a (unique separable) isogeny $\phi_G : E_1 \to E_2$ with kernel G, and it is possible to compute ϕ_G using Vélu formulae [Vél71] in time *linear* in $\#G$ using operations in $\mathbb{F}_{q^t}$, where G is defined over $\mathbb{F}_{q^t}$. For more details see Proposition III.4.12 of Silverman [Sil09], Section 12.3 of Washington [Was08], or Section 25.1 of Galbraith [Gal12]. We sometimes write $E_2 = E_1/G$.

The degree of an isogeny is its degree as a morphism of curves (see Section II.2 of Silverman [Sil09] or Section 12.2 of Washington [Was08]). A separable isogeny with kernel G has degree equal to $\#G$. If $\phi_1 : E_1 \to E_2$ and $\phi_2 : E_2 \to E_3$ are isogenies then $\deg(\phi_2 \circ \phi_1) = \deg(\phi_2)\deg(\phi_1)$.

Theorem 1. *(Corollary III.4.11 of [Sil09]; Theorem 9.6.18 of [Gal12]) Let E_1, E_2, E_3 be elliptic curves over $\mathbb{F}_q$ and $\phi : E_1 \to E_2$, $\psi : E_1 \to E_3$ isogenies over $\mathbb{F}_q$. Suppose*

$\ker(\phi) \subseteq \ker(\psi)$ *and that* ψ *is separable. Then there is a unique isogeny* $\lambda : E_2 \to E_3$ *defined over* $\mathbb{F}_q$ *such that* $\psi = \lambda \circ \phi$.

The above facts show that an isogeny $\phi_1 : E_1 \to E_2$ of composite degree can always be factored as the composition of isogenies of prime degree (see Theorem 25.1.2 of [Gal12]). In many applications, this is the most efficient way to represent and compute an isogeny.

Isogenies are naturally represented as rational maps between concrete models of elliptic curves; however for large-degree isogenies this representation may not be efficient. When the degree is a smooth number one can instead represent the large degree isogeny as a composition of low degree isogenies. Alternatively, such an isogeny can be represented by a sequence of j-invariants that are the codomains of the successive (low-degree) isogeny steps. Any isogeny can also alternatively be represented by its kernel, or more precisely a kernel generator.

While computing a rational map representation of an isogeny of prime degree ℓ requires $O(\ell)$ time just to write the output, evaluating this isogeny on a domain point can be done more efficiently with the so-called "square root Vélu" formulae [BDFLS20].

We denote by $\#E(\mathbb{F}_q)$ the number of points on an elliptic curve E defined over $\mathbb{F}_q$ (including O). The Tate isogeny theorem states that if E_1 and E_2 are elliptic curves over $\mathbb{F}_q$ with $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$ then there is an isogeny $\phi_1 : E_1 \to E_2$ defined over $\mathbb{F}_q$.

If $t = q + 1 - \#E(\mathbb{F}_q)$, then $|t| \leq 2\sqrt{q}$. An elliptic curve is called *supersingular* if $p \mid t$, where q is a power of the prime p, and is called *ordinary* otherwise. It follows that E is supersingular if $\#E(\mathbb{F}_q) \equiv 1 \pmod{p}$, and in fact for supersingular curves one has $\#E(\mathbb{F}_{q^n}) \equiv 1 \pmod{p}$ for all $n \in \mathbb{N}$.

2.2 Endomorphism rings and isogeny graphs

The endomorphism ring of E (see Section III.9 of Silverman [Sil09]) is the set of isogenies from E to itself, together with the zero map $[0] : E \to E$ given by $[0](P) = O$. In other words

$$\mathrm{End}(E) = \{\phi : E \to E\} \cup \{[0]\}.$$

This is a ring where addition of isogenies is defined pointwise using the elliptic curve addition $(\phi_1 + \phi_2)(P) = \phi_1(P) + \phi_2(P)$ and multiplication is composition of isogenies. Note that $\mathbb{Z} \subset \mathrm{End}(E)$ from the map $n \mapsto [n]$.

When E is a supersingular curve then a theorem of Deuring states that $\mathrm{End}(E)$ is a maximal order in the quaternion algebra ramified at p and infinity (see Theorem III.9.3 of [Sil09] or Theorem 42.1.9 of Voight [Voi21]). When E is ordinary the situation is much simpler, as $\mathrm{End}(E)$ is an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{t^2 - 4q})$.

Our main focus in this survey is supersingular curves, as they are the ones currently used in cryptographic applications.

For any supersingular elliptic curve E in characteristic p and any prime $\ell \neq p$, there are exactly $\ell + 1$ isogenies of degree ℓ with E as their domain (up to isomorphisms of the co-domain curves). To any prime numbers p and ℓ, one can associate a *supersingular isogeny graph* where each vertex corresponds to a supersingular curve (up to isomorphism) and each edge to an isogeny of degree ℓ between the corresponding curves. Ignoring at most two exceptional points, this is an undirected $(\ell + 1)$-regular graph. In the supersingular case, the graph is Ramanujan, meaning it has optimal expansion properties (see Section 4 of [CLG09]). The expansion properties imply that any two curves are connected by a path of length $O(\log p)$, and that random walks on the graph quickly converge to the uniform distribution (apart from at most 2 special vertices). All supersingular curves are defined over $\mathbb{F}_{p^2}$.

Given two elliptic curves (E_0, E_1) over $\mathbb{F}_q$ that are isogenous over $\mathbb{F}_q$, there are infinitely many isogenies between them. In the ordinary case, where for simplicity we assume that the endomorphism ring of both curves is the maximal order in the imaginary quadratic field, the set of isogenies from E_0 to E_1 corresponds to an ideal class. In the supersingular case the set of isogenies from E_0 to E_1 corresponds to a rank-4 $\mathbb{Z}$-module in a quaternion algebra.

When the endomorphism ring of an elliptic curve is known, one may instead represent an isogeny from that curve in terms of a module in the endomorphism ring, namely an ideal of a quadratic imaginary number field in the ordinary case, and a (left or right) ideal of a maximal order in a quaternion algebra in the supersingular case. Modulo some restrictions on parameters (such as requiring smooth or power-smooth norms for ideals, and working with special orders), one can transform this representation into the other standard representations (sequences of j-invariants, or kernel generators) in polynomial-time, and vice versa. We give further details in Section 2.5.

2.3 SIDH

Supersingular Isogeny Diffie-Hellman (SIDH) [JDF11,DFJP14] is a key exchange protocol based on isogenies between supersingular curves defined over $\mathbb{F}_{p^2}$. It is the basis for SIKE [JAC$^+$17], an isogeny based KEM that was submitted to the NIST post-quantum standardization process. Of course, due to recent cryptanalysis [CD23,MMP$^+$23,Rob23] SIDH and SIKE are no longer considered secure. Nevertheless, it is necessary for some of the schemes mentioned in our paper to introduce some concepts from SIDH.

SIDH is built around a commutative square constructed as the *push-out* of two isogenies of coprime degrees. We will use the same squares in the proofs of knowledge of supersingular isogenies in Section 5. For efficiency, SIDH restricts to pairs of isogenies of degree 2^n and 3^m, where $2^n \approx 3^m$ and the finite field characteristic is an "SIDH prime" $p = 2^n 3^m f - 1$.

Given such a prime, there exists over $\mathbb{F}_{p^2}$ an isogeny class of supersingular curves with group structure

$$E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p+1)\mathbb{Z})^2 \cong (\mathbb{Z}/2^n\mathbb{Z})^2 \times (\mathbb{Z}/3^m\mathbb{Z})^2 \times (\mathbb{Z}/f\mathbb{Z})^2.$$

Let E be one such curve. For example, choose the curve $y^2 = x^3 + x$ of j-invariant 1728, which is always supersingular modulo $p \equiv 3 \pmod 4$. Furthermore, $\mathrm{End}(E)$ is known for this curve.

Let $A \subset E[2^n]$ and $B \subset E[3^m]$ be cyclic groups of order $2^n, 3^m$, respectively. They define isogenies $\phi_A : E \to E/A$ and $\psi_B : E \to E/B$. The "SIDH square" on (A, B) is the commutative diagram

$$\begin{tikzcd} E \arrow[r, "\phi_A"] \arrow[d, "\psi_B"'] & E/A \arrow[d, "\psi'_B"] \\ E/B \arrow[r, "\phi'_A"] & E/\langle A, B\rangle \end{tikzcd} \tag{1}$$

where $\ker(\psi'_B) = \phi_A(B)$, $\ker(\phi'_A) = \psi_B(A)$, and $\langle A, B\rangle$ denotes the group generated by A and B. We say that (ϕ_A, ϕ'_A) (and (ψ_B, ψ'_B)) are "parallel" isogenies.

Definition 1 (Parallel isogenies). *Let $\phi : E_0 \to E_1$ and $\phi' : E_2 \to E_3$ be separable isogenies of degree d. We say that (ϕ, ϕ') are* parallel with respect to a separable isogeny *$\psi : E_0 \to E_2$ of degree coprime to d if* $\ker(\phi') = \psi(\ker(\phi))$.

Lemma 1. *Let $\phi : E_0 \to E_1$ and $\phi' : E_2 \to E_3$ be parallel with respect to some ψ, then there exists an isogeny $\psi' : E_1 \to E_3$ parallel to ψ with respect to ϕ, defined by* $\ker(\psi') = \phi(\ker(\psi))$.

Proof. By comparing kernels it is clear that $\phi' \circ \psi = \psi' \circ \phi$, up to composing ψ' with an isomorphism, and thus the image curve of ψ' is E_3. Then ψ' is parallel to ψ by definition. □

Given generators for A and B, the curves E/A and E/B can be efficiently computed using Vélu's formulae. The bottom-right curve $E/\langle A, B\rangle$ can then be computed in two ways as

$$(E/A)/\phi_A(B) \;\cong\; E/\langle A, B\rangle \;\cong\; (E/B)/\psi_B(A). \tag{2}$$

This is as much as we need for defining proofs of knowledge of isogenies.

Other base fields. The setup we presented above is the most common and the most efficient one for SIDH-like proofs of knowledge. However, the paper that introduced SIDH also considered primes of the form $p = \ell_A^n \ell_B^m f \pm 1$ for any small primes ℓ_A, ℓ_B. A variant named B-SIDH [Cos20] used primes such that $p^2 - 1 = Nf$, where N is a sufficiently large smooth integer. Finally, variants of the SIDH key exchange designed to resist the recent attacks consider $p = 4N - 1$, where N is a product of sufficiently many distinct small primes [FMP23]. In this work, we will focus on the basic SIDH case, but the ideas are easily generalized to all other settings. Ultimately we will see in Section 5.3 that, with some loss of efficiency, SIDH-like commutative squares can be constructed in any characteristic.

2.4 CSIDH

Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) [CLM+18] was proposed by Castryck, Lange, Martindale, Panny, and Renes. It is an example of a cryptographic group action, and it builds on ideas of Couveignes [Cou06], Rostovtsev and Stolbunov [RS06], and De Feo, Kieffer and Smith [DFKS18]. Let G be a finite abelian group and X a set of size $|X| = |G|$. The group G acts on X if there is a binary operation $G \times X \to X$ which we write as $(g, x) \mapsto g * x$. We require $g * (g' * x) = (gg') * x$ for all $g, g' \in G$ and $x \in X$.

In the case of CSIDH, the group G is the ideal class group of the order $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$ in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. The set X is the set of isomorphism classes of elliptic curves over $\mathbb{F}_p$ with endomorphism ring $\mathrm{End}(E) \cong \mathcal{O}$, which are necessarily supersingular. Alternatively, one could work with supersingular curves whose $\mathbb{F}_p$-endomorphism ring is $\mathbb{Z}[(1 + \sqrt{-p})/2]$, see [CD20].

The use of supersingular curves over $\mathbb{F}_p$ is for efficiency reasons. In fact, we choose the prime p to have the special form $p = 4\ell_1 \cdots \ell_r - 1$, for some integer r, where the ℓ_i are distinct small odd primes. Let $E_0/\mathbb{F}_p$ be the supersingular curve defined by $y^2 = x^3 + x$. The endomorphism ring of E_0 is a maximal order in a quaternion algebra, but the subring of endomorphisms that are defined over $\mathbb{F}_p$ (we call this the $\mathbb{F}_p$-endomorphism ring) is isomorphic to a ring containing $\mathbb{Z}[\sqrt{-p}]$. It is known that the ideal class group of the $\mathbb{F}_p$-endomorphism ring acts on the set of supersingular elliptic curves defined over $\mathbb{F}_p$, where each ideal corresponds to an isogeny [Wat69].

We make the reasonable assumption that the class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ is generated by the r ideals $\mathfrak{l}_i = (\ell_i, 1 + \sqrt{-p})$ for i from 1 to r. The class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ acts freely and transitively on the set $\mathcal{E}\ell\ell$ of $\mathbb{F}_p$-isomorphism classes of elliptic curves with $\mathbb{F}_p$-endomorphism ring $\mathbb{Z}[\sqrt{-p}]$. We can efficiently compute the action of the ideal classes $\bar{\mathfrak{l}}_1, \cdots, \bar{\mathfrak{l}}_r$, and their inverses.

For a vector $\mathbf{x} \in \mathbb{Z}^r$ and $E \in \mathcal{E}\ell\ell$ we define

$$[\mathbf{x}]E := \left(\prod_{i=1}^{r} \bar{\mathfrak{l}}_i^{\,x_i} \right) * E \,,$$

where $*$ is the action of the ideal class group. This is known as the *CSIDH action.*

Technically, we have defined the infinite group $\mathbb{Z}^r$ to act on the finite set $\mathcal{E}\ell\ell$, which does not match our earlier definition of a group action. In fact, one can define a lattice

$$L = \{\mathbf{x} \in \mathbb{Z}^r : [\mathbf{x}]E = E\}$$

such that the class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ is isomorphic to $\mathbb{Z}^r/L$ and we really have an action of $\mathbb{Z}^r/L$ on $\mathcal{E}\ell\ell$.

We can now define the relation $\mathcal{R}$ as

$$\mathcal{R}_{\text{CSIDH}} = \{(E, \mathbf{x}) \in \mathcal{E}\ell\ell \times \mathbb{Z}^r \mid [\mathbf{x}]E_0 = E\} .$$

The CSIDH key exchange protocol works by Alice choosing random $\mathbf{x}_A$ and sending $E_A = [\mathbf{x}_A]E_0$ to Bob. Similarly, Bob chooses random $\mathbf{x}_B$ and sends $E_B = [\mathbf{x}_B]E_0$ to Alice. The shared key that both of them can compute is

$$[\mathbf{x}_A]E_B = [\mathbf{x}_A + \mathbf{x}_B]E_0 = [\mathbf{x}_B]E_A.$$

2.5 Quaternion algorithms and the KLPT algorithm

Kohel, Lauter, Petit, and Tignol (KLPT) [KLPT14] introduced important algorithmic ideas for computing with quaternion algebras and computing isogenies of smooth degree. We will not cover all the mathematics. Instead, we sketch the basic functionalities provided by their work (and extended by other authors).

For this section, we always assume E_0 is a very particular supersingular curve (namely, it is supersingular, defined over $\mathbb{F}_p$, and has a non-scalar endomorphism of very small degree). The canonical example when $p \equiv 3 \pmod 4$ is $y^2 = x^3 + x$, which has the non-trivial automorphism $\iota(x, y) = (-x, iy)$ where $i \in \mathbb{F}_{p^2}$ satisfies $i^2 = -1$. The p-power Frobenius map $\pi(x, y) = (x^p, y^p)$ satisfies $\pi \circ \iota = -\iota \circ \pi$ and $\pi^2 = [-p]$. It follows that $\mathrm{End}(E_0)$ contains a subring isomorphic to $\mathbb{Z} \oplus i\mathbb{Z} \oplus \sqrt{-p}\mathbb{Z} \oplus i\sqrt{-p}\mathbb{Z}$. The norm of an element $w + ix + \sqrt{-p}(y + iz)$ is $w^2 + x^2 + p(y^2 + z^2)$, and the KLPT algorithm heavily relies on norm forms that can be written as $Q(w, x) + pQ(y, z)$ where Q is a binary quadratic form of small discriminant.

The KLPT algorithm and other results we need all rely on the Deuring correspondence, which says that the endomorphism ring of a supersingular elliptic curve is a maximal order $\mathcal{O}$ in a quaternion algebra, and that an isogeny $\phi : E_0 \to E_1$ corresponds to an ideal I such that the left-order of I is $\mathcal{O}_0 = \mathrm{End}(E_0)$ and the right-order of I is $\mathcal{O}_1 = \mathrm{End}(E_1)$. Furthermore, the degree of ϕ is equal to the reduced norm of I. Key points are that maximal orders are defined up to conjugation $\alpha\mathcal{O}\alpha^{-1}$ by a non-zero element α in the quaternion algebra, and that the module I is also defined up to equivalence $I \equiv \alpha I \equiv I\alpha$, since I having left-order $\mathcal{O}$ means αI has left-order $\alpha\mathcal{O}\alpha^{-1}$. Hence the infinitely many choices of isogeny $\phi : E_0 \to E_1$ correspond to the infinitely many equivalent ideals I. General references for this topic include [EHL$^+$18,KLPT14].

The KLPT algorithm allows to replace ideals by power-smooth degree ones in the same class. More precisely, let $\mathcal{O}_0$ be a maximal order in the quaternion algebra $B_{p,\infty}$ whose norm is of the form $Q(w,x)+pQ(y,z)$, and let $\mathcal{O}_1$ be another maximal order. The KLPT algorithm takes as input a small prime ℓ and a sufficiently large integer n (the original KLPT paper requires $n > \frac{7}{2}\log_\ell(p)$), and returns an ideal I of norm ℓ^n that is a left $\mathcal{O}_0$-ideal and whose right order is isomorphic to $\mathcal{O}_1$. The algorithm was adapted in [GPS20] to produce ideals of B-powersmooth norm (meaning the norm is of the form $N = \prod_i \ell_i^{e_i}$ where the ℓ_i are distinct primes and $\ell_i^{e_i} \leq B$). One takes $B = c\log p$ for a suitable constant c. The heuristic analyses in [GPS20] suggest that this can be done for any choice of N of size $\log(N) \approx \frac{7}{2}\log(p)$. This is improved to $\log(N) \approx 3\log(p)$ with another small modification to the KLPT algorithm proposed in [PS18].

Because their norms are powersmooth, ideals output by the KLPT algorithm can be efficiently converted into isogenies using standard techniques based on kernel identification [Wat69]. The KLPT algorithm has this way enabled efficient solutions to several algorithmic problems related to supersingular curves, their endomorphisms and isogenies, including:

- Given a maximal order $\mathcal{O}$, to compute an elliptic curve E_1 with $\mathrm{End}(E_1) \cong \mathcal{O}$ (namely by constructing an isogeny $\phi : E_0 \to E_1$).
- Given E_0 and $\mathrm{End}(E_1)$ to compute a (random or canonical) smooth or semi-smooth degree isogeny from E_0 to E_1.
- Given E_0 and an isogeny $\phi : E_0 \to E_1$, to compute $\mathrm{End}(E_1)$.

We refer to [KLPT14,GPS20,EHL$^+$18,DFKL$^+$20,DFLLW23] for details.

3 Zero-knowledge proofs of knowledge

In this section, we recall standard definitions for sigma protocols (an important class of zero-knowledge protocols), the GMW protocol for graph isomorphism, and the connections between sigma protocols and signatures.

3.1 Sigma protocols

Many zero-knowledge proof protocols (e.g., Schnorr proofs) fall in the so-called framework of *sigma protocols* for *hard relations*. A *hard relation* is one where there is an efficient algorithm to generate pairs (x, w), but it is computationally infeasible to compute w given only x. A sigma protocol is a type of proof of knowledge protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, where the prover wants to convince the verifier that for some statement x known to $\mathcal{P}$ and $\mathcal{V}$, he knows a witness w, such that $(x, w) \in \mathcal{R}$, for some relation $\mathcal{R}$. Informally, we want the sigma protocol to be complete, which means that an honest prover can always convince the verifier that he knows a witness; we want the sigma protocol to be sound, which means that if a prover does not know a witness, he can only convince the verifier with a limited probability, and we want the protocol to be zero-knowledge, which means that the prover does not reveal anything about the witness. More formally, a sigma protocol is defined as follows:

Definition 2. *We say $\Sigma = (P_1, P_2, V_2)$ is a sigma protocol with challenge space $\mathcal{C}$, for a relation $\mathcal{R}$ if the following properties are satisfied:*

- **Three-round public coin.** *The prover $\mathcal{P}$ starts the protocol by generating a first message* $\mathsf{com} \leftarrow P_1(x, w)$ *(often called a commitment), and sending it to the verifier $\mathcal{V}$. Then, the verifier chooses a challenge* $\mathsf{chall} \leftarrow \mathcal{C}$ *uniformly at random from the challenge space $\mathcal{C}$, and sends it to $\mathcal{P}$. Finally, $\mathcal{P}$ computes a response* $\mathsf{resp} \leftarrow P_2(\mathsf{chall})$ *(P_1 and P_2 share a state) and sends it to $\mathcal{V}$, who runs the verification algorithm* $V_2(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$*, which outputs 1 or 0, signalling that $\mathcal{V}$ accepts or rejects the proof respectively. (See Fig. 1)*
- **Completeness/correctness.** *The completeness (sometimes called correctness) property says that if both parties follow the protocol, and if the prover uses a valid witness w such that $(x, w) \in \mathcal{R}$, then the verifier will accept the proof with probability 1. In other words, for all $(x, w) \in \mathcal{R}$, we have*

$$\Pr\left[V_2(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp}) = 1 \,\middle|\, \begin{array}{l} \mathsf{com} \leftarrow P_1(x, w) \\ \mathsf{chall} \leftarrow \mathcal{C} \\ \mathsf{resp} \leftarrow P_2(\mathsf{chall}) \end{array} \right] = 1 \,.$$

- ***n*-Special Soundness.** *A sigma protocol has n-special soundness if there exists an efficient extractor algorithm that given n transcripts of the form* $(x, \mathsf{com}, \mathsf{chall}_i, \mathsf{resp}_i)$ *for $1 \leq i \leq n$, with* $V_2(x, \mathsf{com}, \mathsf{chall}_i, \mathsf{resp}_i) = 1$ *for all i, and* $\mathsf{chall}_i \neq \mathsf{chall}_j$ *for all $1 \leq i < j \leq n$, outputs a witness w such that $(x, w) \in \mathcal{R}$.*
- **Special honest-verifier zero-knowledge.** *A sigma protocol is called special honest-verifier zero-knowledge if there exists an efficient simulator $\mathcal{S}$ that given x and* $\mathsf{chall} \in \mathcal{C}$ *outputs transcripts* $(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ *that are indistinguishable from transcripts of honest executions of the protocol. More precisely, for every $(x, w) \in \mathcal{R}$, and every* $\mathsf{chall} \in \mathcal{C}$ *we require that*

$$\mathcal{S}(x, \mathsf{chall}) \approx \left\{ (x, \mathsf{com}, \mathsf{chall}, \mathsf{resp}) \,\middle|\, \begin{array}{l} \mathsf{com} \leftarrow P_1(x, w) \\ \mathsf{resp} \leftarrow P_2(\mathsf{chall}) \end{array} \right\} .$$

  If the distributions are identical, we say the sigma protocol has perfect *special honest-verifier zero-knowledge (sHVZK), if the distributions are statistically close, we say the protocol has* statistical *sHVZK, and if the distributions are only computationally indistinguishable, then we say the sigma protocol is* computationally *sHVZK.*

Section 4.3 of Goldreich [Gol01] gives a stronger formulation of computational zero knowledge that considers malicious verifiers.

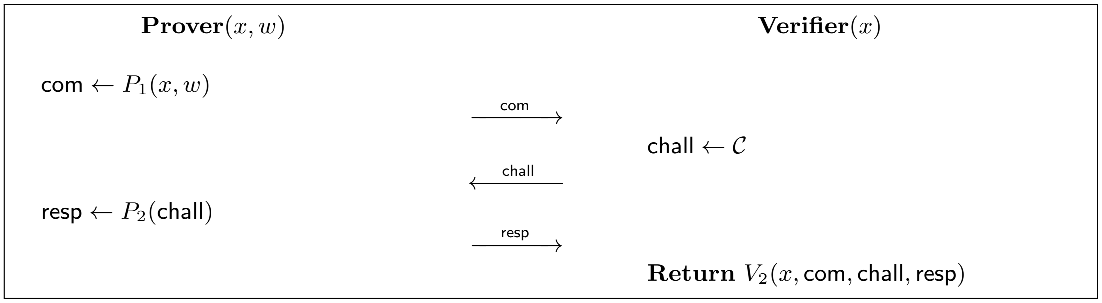
$\textbf{Prover}(x, w)$
$\textbf{Verifier}(x)$
$\mathsf{com} \leftarrow P_1(x, w)$
$\xrightarrow{\mathsf{com}}$
$\mathsf{chall} \leftarrow \mathcal{C}$
$\xleftarrow{\mathsf{chall}}$
$\mathsf{resp} \leftarrow P_2(\mathsf{chall})$
$\xrightarrow{\mathsf{resp}}$
$\textbf{Return } V_2(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$

Fig. 1. The structure of a general sigma protocol.

Remark 1. The existence of the simulator implies that, for any sigma protocol that is special honest-verifier zero-knowledge, there is always a cheating prover who does not know a witness but who guesses the challenge and can respond correctly if their guess is correct. Hence one can always cheat with probability at least $1/|\mathcal{C}|$. We call the successful cheating probability the *soundness error*. A protocol with soundness error ϵ should be computed k times, so that the overall soundness error ϵ^k is negligible (usually $< 2^{-128}$). The ideal case is when the soundness error is equal to $1/|\mathcal{C}|$ and $|\mathcal{C}|$ is large.

Suppose the sigma protocol is n-special sound where $n > 2$. Suppose further that, for a statement x and a commitment com, a cheating prover can respond to n distinct challenges. Then the prover can generate n transcripts, and by the n-special soundness property can compute a witness w such that $(x, w) \in \mathcal{R}$. It follows that the cheating prover must only be able to answer up to $n - 1$ distinct challenges. Hence if a protocol is n-special sound then the soundness error should be at most $(n - 1)/|\mathcal{C}|$.

Remark 2. The special honest-verifier zero-knowledge property implies that an honest execution of the protocol does not reveal any additional information about w, because everything that could be learned by inspecting a transcript $(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ could also be learned directly from x alone, by first running the simulator $\mathcal{S}(x, \mathsf{chall})$ and then inspecting the simulated transcript.

Remark 3. Perfect ZK and statistical ZK provide an information-theoretic guarantee that a verifier learns nothing about the witness from the proof. Hence they are preferable to computational ZK. Computational ZK usually requires introducing new computational assumptions (typically decisional assumptions) beyond what is needed for the hardness of the relation, and this leads to increased complexity and possible weaknesses. The sigma protocols in the case of CSIDH have perfect or statistical ZK. The original protocols for $\mathcal{R}_{\text{deg}}$ that we describe in Section 5 are only computational ZK, but recent work surveyed in Section 5.3 gives a statistical ZK protocol. In Section 6, the sigma protocol underlying GPS has statistical ZK, while SQISign requires a computational assumption for ZK.

3.2 Generic solutions

It is well-known since Goldreich, Micali, and Widgerson [GMW91] that any NP statement can be proved in zero-knowledge, assuming the existence of one-way functions. This is achieved by reducing the NP statement to an instance of 3-coloring and applying a zero-knowledge protocol for the 3-coloring problem.

All the statements listed in the introduction are in NP (at least if we restrict the isogeny degrees d to be smooth), thus we trivially obtain zero-knowledge proofs for each of them. Additionally, we may use isogenies to instantiate the one-way function, e.g., the CGL hash function [CLG09], thus basing the security of the zero-knowledge proof entirely on the computational hardness of computing isogenies between random supersingular elliptic curves.

We stress that while this approach gives a feasibility result in theory, it would not be practical: the 3-coloring problem obtained by translating any one of the relations of the introduction will be huge, and the zero-knowledge protocol for 3-coloring provided in [GMW91] requires to commit to all vertices with a one-way function and repeat this and other operations a number of times quadratic in the number of vertices.

The research on more efficient generic ZK-proof systems has been extremely active lately, producing several systems capable of proving real-world-sized statements. However, these tend to be based on stronger assumptions than the existence of one-way functions, such as the Random Oracle Model.

In the context of proving knowledge of isogenies, generic proof systems were first applied in [CSRHT22], with the goal of defining *Verifiable Delay Functions* (VDF). We don't cover VDFs in this article, but we nevertheless quickly summarize the main techniques of [CSRHT22], as they may be useful in other contexts.

Let ℓ be prime. The *classic modular polynomial* $\varPhi_\ell$ is a bivariate polynomial with integer coefficients such that $\varPhi_\ell(j, j') = 0$ if and only if there exists an isogeny of degree ℓ between curves of j-invariant j and j'. The degree and the size of the coefficients of $\varPhi_\ell$ grow polynomially in ℓ, thus it cannot be used to decide whether two curves are ℓ-isogenous when ℓ grows exponentially large. However, the typical case we encounter in cryptography is when two curves are d-isogenous with d being smooth. Thus, the isogeny of degree d can be written as a composition of isogenies $E_0 \to E_1 \to \cdots \to E_n$ of degrees $\ell_1, \dots, \ell_n$. Since the ℓ_i are polynomially small, from the sequence of j-invariants one can extract the isogenies $E_i \to E_{i+1}$ efficiently. Hence, proving knowledge of an isogeny $E_0 \to E_n$ of degree d is equivalent to proving knowledge of $j_0, j_1, \dots, j_n$ such that

$$\varPhi_{\ell_1}(j_0, j_1) = \varPhi_{\ell_2}(j_1, j_2) = \cdots = \varPhi_{\ell_n}(j_{n-1}, j_n) = 0. \tag{3}$$

The proof becomes even simpler when $d = \ell^n$, then only one modular polynomial is needed to prove the statement.

The proof of knowledge of [CSRHT22] is based on proving the polynomial identities above using a tailored Sumcheck protocol [LFKN92], an interactive protocol that lets a prover convince a verifier that

$$\sum_{\vec{x}\in\{0,1\}^s} P(\vec{x}) = 0, \tag{4}$$

where P is a polynomial in s variables with coefficients in a finite field, and the values 0,1 are understood as finite field elements.

The way Eq. (3) is reduced to a Sumcheck equation is by *arithmetization.* Assume that $n < 2^s$. First, the sequence $j_0, \dots, j_n$ is interpolated to an s-variate polynomial j such that

$$j(\vec{i}) = j_i, \tag{5}$$

where $\vec{i}$ denotes the integer i written in binary. Second, interpolate the $2s$-variate polynomial

$$L(\vec{x},\vec{y}) = \begin{cases} 1 & \text{if } x+1=y, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Finally, Eq. (3) is equivalent to

$$\Phi_\ell\bigl(j(\vec{x}), j(\vec{y})\bigr) \cdot L(\vec{x},\vec{y}) = 0 \qquad \forall \vec{x},\vec{y} \in \{0,1\}^s. \tag{7}$$

To convince the verifier, the prover commits to the polynomial in the left-hand side above. The receiver responds with a random *weighting polynomial* $w(x,y)$, and the prover uses the Sumcheck protocol to prove that

$$\sum_{\vec{x}\in\{0,1\}^s} \sum_{\vec{y}\in\{0,1\}^s} w(\vec{x},\vec{y}) \cdot \Phi_\ell\bigl(j(\vec{x}), j(\vec{y})\bigr) \cdot L(\vec{x},\vec{y}) = 0. \tag{8}$$

The fact that $w(x,y)$ was chosen randomly after the prover had committed to $\Phi_\ell(j(x), j(y)) \cdot L(x,y)$ is enough to convince the verifier that Eq. (7) holds.

The actual protocol in [CSRHT22] is slightly more involved because it also convinces the verifier that the sequence of j-invariants is chosen pseudo-randomly from a seed. Their main motivation is to have a verifier that only takes $O(\log(n))$ time to verify, thus their last step is converting the Sumcheck to a SNARG [Mic00].

So far, we have only achieved soundness: the only difference between this protocol and simply handing the list $j_0, \dots, j_n$ to the verifier is the better asymptotic complexity of verification. Adding zero-knowledge to a Sumcheck protocol can be done in a standard way using *polynomial commitments* [KZG10], but this step is not analyzed in [CSRHT22] because VDFs do not require it.

The concrete cost of the non-ZK protocol above is not analyzed in [CSRHT22]. Recent work by Cong, Lai, and Levin [CLL23] shows how to express the conditions in equation (3) as a rank-1 constraint system (R1CS). They then show how existing proof systems for

R1CS can be applied, and give implementation results. Such systems give statistical zero-knowledge and are post-quantum secure. Very practical results are obtained from using the Aurora system of Ben-Sasson et al [BCR$^+$19].

3.3 GMW protocol for graph isomorphism

Apart from the generic techniques in the previous section, all existing zero-knowledge proofs of knowledge of an isogeny can be seen as variations of the classical [GMW91] sigma protocol that proves knowledge of a graph isomorphism, i.e., a relabeling of the vertices that turns one graph into another one (see Section 3.3). (Do not confuse the graphs in this analogy with isogeny graphs; the graphs are replaced with elliptic curves and the graph isomorphisms are replaced by isogenies.) More precisely, this is a sigma protocol for the relation

$$\mathcal{R} = \{((G,H),\phi) \mid \phi \text{ is a graph isomorphism from } G \text{ to } H\} \ .$$

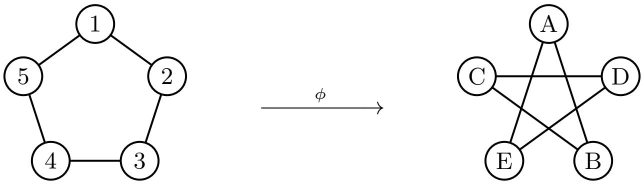

Fig. 2. Example of two isomorphic graphs. An isomorphism is given by the map ϕ that sends 1 to A, 2 to B, and so on.

We denote the vertex sets of G and H by $V(G)$ and $V(H)$, and the set of edges of G and H by $E(G)$ and $E(H)$. The sigma protocol goes as follows:

- **Commitment phase.** The prover picks a random bijection ρ from $V(G)$ to a new vertex set $V' = \{1,\dots,l\}$ of size $l = |V(G)| = |V(H)|$. Then he computes a new set of edges
$$E' = \{(\rho(x),\rho(y)) \,|\, \forall(x,y) \in E(G)\} \ .$$
He sends the new graph $G' = (V',E')$ to the verifier.
- **Challenge phase.** The verifier chooses a random challenge bit $\mathsf{chall} \in \{0,1\}$ and sends it to the prover.
- **Response phase.** If $\mathsf{chall} = 0$ the prover sends ρ, otherwise the prover sends $\rho' = \rho \circ \phi^{-1}$.
- **Verification.** The verifier accepts the proof if $\rho(G_1) = G'$ in case $\mathsf{chall} = 0$, and he accepts if $\rho'(G_2) = G'$ in case $\mathsf{chall} = 1$.

It is relatively straightforward to check that this protocol satisfies the completeness, special soundness, and perfect honest-verifier zero-knowledge properties.

$$\begin{tikzcd}
 & G' & \\
G_1 \arrow[ur, "\rho"] \arrow[rr, "\phi"] & & G_2 \arrow[ul, dashed, "\rho \circ \phi^{-1}"']
\end{tikzcd}$$

Fig. 3. The classic zero-knowledge proof of knowledge of a graph isomorphism.

3.4 Non-interactive ZK and signatures

Sigma protocols are interactive protocols between a prover and a verifier, and an important feature of them is that the challenge is chosen by the verifier after the commitment has been sent. In many applications, it is inconvenient to work with interactive protocols and so we want non-interactive versions of these. One important application of non-interactive sigma protocols for hard relations is the construction of digital signatures that are existentially unforgeable under adaptive chosen-message attacks.

The best-known approach to making sigma protocol non-interactive is the Fiat-Shamir heuristic. The basic idea of the Fiat-Shamir transform is to use a cryptographic hash function H to compute the challenge, as $\mathsf{chall} = H(\mathsf{com})$. This can only be secure when the challenge is a bit string of sufficient length such that the soundness error is negligible. In this case, the protocol should still be sound, as the prover cannot choose chall in advance and compute a commitment that hashes to that value.

In the context of digital signatures, the Fiat-Shamir transform is as follows. The public key of the signature scheme is an instance x for a hard relation, and the secret key is a witness w for the relation. To sign a message m, the prover runs the sigma protocol, except that it replaces the challenge by the hash value $\mathsf{chall} = H(m, x, \mathsf{com})$, where m is the message to sign. A signature then consists of the commitment and response messages. To verify a signature, one simply recomputes the hash value and runs the verification algorithm for the sigma protocol. Intuitively, this is secure because the security properties of the hash function force the challenge to be a randomly distributed value generated after the commitment, as in the normal execution of the sigma protocol. The n-special soundness of the sigma protocol provides an extractor that, by re-winding a forger in the random oracle model, allows to compute a witness from the signatures which are output by the forger. The zero-knowledge property of the sigma protocol provides a simulator that allows to generate signatures in the random oracle model on messages queried by the forger to the signing oracle. Hence we have security against chosen message attacks.

In the context of post-quantum cryptography, the random oracle model should be replaced by the quantum random oracle model. Fiat-Shamir signatures have also been proven secure in this model under certain conditions [KLS18], and otherwise the Unruh transform offers an alternative, albeit less efficient construction [Unr15].

While the Fiat-Shamir heuristic allows to construct a signature from any sigma protocol for a hard relation, not every Fiat-Shamir-style signature corresponds to a special-sound sHVZK sigma protocol. There are two reasons for this. First, the computational assumption(s) required for the hardness of forgery are not necessarily the same as the computational assumption of extracting a witness for a statement x. Second, in the case where the signature scheme requires a computational assumption for zero-knowledge, there is a subtle difference between the definition of computational zero-knowledge for sigma protocols and the requirements for the simulator to generate signatures in the random oracle model to answer signing queries. Precisely, computational ZK requires computational indistinguishability for every fixed x, whereas for signatures the probability space in the security definition includes the random generation of x.

4 The CSIDH setting

In this section, we discuss two sigma protocols for the natural relation coming from group actions in the specific case of CSIDH [CLM$^+$18] (see Definition 3 below).

Recall that the action of a vector $\mathbf{x} \in \mathbb{Z}^r$ on $E \in \mathcal{E}\ell\ell$ is defined by

$$[\mathbf{x}]E := \left(\prod_{i=1}^{r} \bar{\mathfrak{l}}_i^{\,x_i} \right) * E \,,$$

where $*$ is the action of the ideal class group. Note that $*$ is computed using a sequence of isogenies of degree ℓ_i corresponding to the prime ideals $\mathfrak{l}_i$.

Definition 3. *The CSIDH relation is*

$$\mathcal{R}_{\text{CSIDH}} = \{(E, \mathbf{x}) \in \mathcal{E}\ell\ell \times \mathbb{Z}^r \mid [\mathbf{x}]E_0 = E\} \,.$$

We now describe two sigma protocols for the relation $\mathcal{R}_{\text{CSIDH}}$. The first protocol is simpler and more efficient, but it requires knowledge of the structure of the class group $\text{cl}(\mathbb{Z}[\sqrt{-p}])$ and the relations between the ideal classes $\bar{\mathfrak{l}}_i$. This is a big disadvantage because (pre)computing this information is expensive, which means the first protocol can only be used for small parameters, e.g., when the order of the class group is $\approx 2^{256}$, see [BKV19].[4] The second protocol is less efficient, but it does not require knowledge of the class group, and can thus be used for larger class group actions.

[4] The structure of a class group can be computed in quantum polynomial time, so this protocol could be used with large class groups if anyone with access to a quantum computer is willing to compute a class group and publish the result (which can be verified efficiently with classical algorithms). However, unlike SeaSign, the asymptotic performance is not thought to be polynomial time.

4.1 CSI-FiSh sigma protocol

We will call the first protocol the CSI-FiSh protocol, even though a variant was already known well before the CSI-FiSh paper. It is a straightforward generalization of the graph isomorphism protocol from Section 3.3 and was already described in the group actions setting by Couveignes [Cou06], Rostovstev and Stolbunov [RS06], and in more detail by De Feo and Galbraith in the CSIDH setting [DFG19]. An optimization of the protocol that uses quadratic twists was added to the protocol in the CSI-FiSh paper [BKV19].

In this section, we assume[5] for simplicity that the class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ is cyclic with a generator $\bar{\mathfrak{g}}$ of known order N, and that we know the discrete logarithms $a_1, \ldots, a_k$ of the ideal classes $\bar{\mathfrak{l}}_1, \ldots, \bar{\mathfrak{l}}_r$ with respect to $\bar{\mathfrak{g}}$. This includes the case of the CSIDH-512 parameter set, proposed by [CLM$^+$18], with $r = 74$, and where the first 73 small primes $\ell_1, \ldots, \ell_{73}$ are the first 73 odd primes, and where $\ell_{74} = 587$. For this choice of prime p, the class group was computed by Beullens, Kleinjung, and Vercauteren [BKV19]. It turned out that the class group is cyclic of order

$$N = 254652442229484275177030186010639202161620514305486423592570860975597611726191\,,$$

and that the ideal class of the first ideal $\mathfrak{l}_1 = (3, \sqrt{-p} - 1)$ generates the entire class group. The discrete logarithms of the remaining $\bar{\mathfrak{l}}_i$ with $i > 1$, as well as a reduced basis for the relation lattice are publicly available.

Given an integer $x \in \mathbb{Z}/N\mathbb{Z}$ and a curve $E \in \mathcal{E}\ell\ell$ we want to compute the action of $\overline{\mathfrak{g}^x}$ on E. Naively computing the action of $\bar{\mathfrak{g}}$ a total of x times would require an exponential amount of time, so this is not efficient. Instead, since we know the discrete logarithms a_i such that $\overline{\mathfrak{g}^{a_i}} = \bar{\mathfrak{l}}_i$ for all i from 1 to r, we can use lattice algorithms to find a short vector $\mathbf{x} \in \mathbb{Z}^r$ such that $\sum_{i=1}^r a_i x_i = x \mod N$. Once we have such a vector we can evaluate $[\mathbf{x}]E$ efficiently. Asymptotically, this could be inefficient, because the lattice algorithms are too slow or produce vectors that are too large, but in practice (at least for the CSIDH-512 parameter set) this is not a problem: for CSIDH-512 the lattice algorithms are much faster than the isogeny computations, and the resulting vector $\mathbf{x}$ is close to optimal. In total, computing the action of $\overline{\mathfrak{g}^x}$ on $E \in \mathcal{E}\ell\ell$ for a random $x \in \mathbb{Z}/N\mathbb{Z}$ is only 15% slower than computing $[\mathbf{x}]E$ for a random $\mathbf{x} \in [-5, 5]^{74}$ (which is done in the CSIDH-512 key exchange protocol). For more details, we refer to [BKV19].

With these details out of the way, we have a group action of $\mathbb{Z}/N\mathbb{Z}$ on $\mathcal{E}\ell\ell$ (instead of a group action of $\mathbb{Z}^r$). With a mild abuse of notation, we denote the action of $x \in \mathbb{Z}/N\mathbb{Z}$ on $E \in \mathcal{E}\ell\ell$ also by $[x]E$. Figure 4 shows the CSI-FiSh sigma protocol, which is an adaptation of the sigma protocol for graph isomorphism to this group action (replacing the action of S_n on graphs of order n by the action of $\mathbb{Z}/N\mathbb{Z}$ on $\mathcal{E}\ell\ell$). However, a difference is that we can make the challenge space slightly larger ($\{-1, 0, 1\}$ instead of $\{0, 1\}$), by exploiting the fact that if E^t is the quadratic twist of $E = [x]E_0$, then E^t is $\mathbb{F}_p$-isomorphic to $[-x]E_0$. A cheating prover who does not know x can win each round with a probability of 1/3.

[5] All the results generalize to the more general case where the class group is not necessarily cyclic.

The proofs that this sigma protocol is complete, 2-special sound for the relation $\mathcal{R}_{\text{CSIDH}}$, and honest-verifier zero-knowledge are straightforward; we refer to [BKV19].

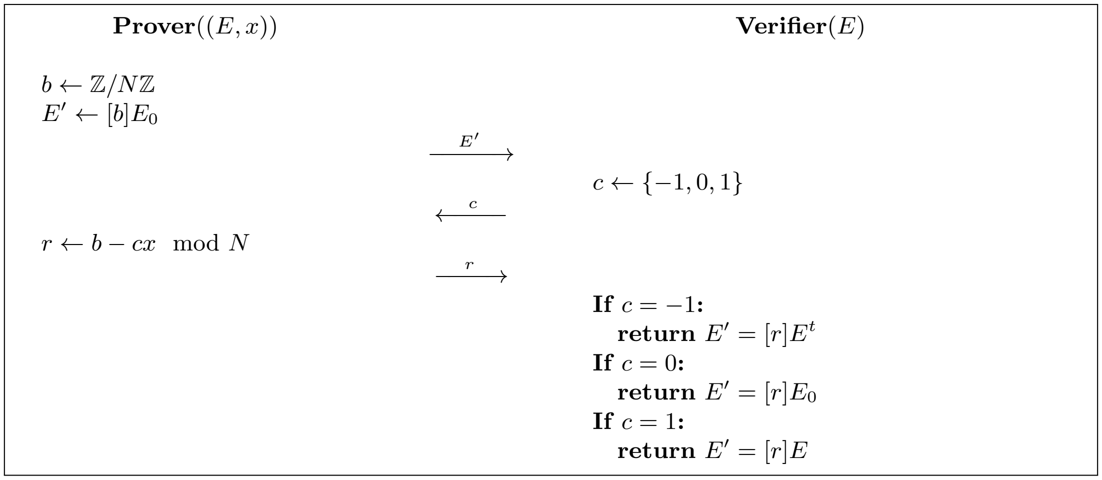

Fig. 4. The CSI-FiSh sigma protocol. Here, and in subsequent figures, the equal sign denotes the equality predicate and the return statements return either true or false.

4.2 CSI-FiSh non-interactive proofs/signatures

We can obtain a non-interactive proof for the CSIDH relation by applying the Fiat-Shamir transform to the sigma protocol of Fig. 4, after amplifying the soundness. The resulting protocol is called CSI-FiSh (Commutative Supersingular Isogeny Fiat-Shamir). Since the base sigma protocol has a challenge space of size 3 and is 2-special sound, the soundness error is 1/3. This means we need to repeat the protocol $k = \lceil \lambda / \log 3 \rceil$ times to get λ bits of security. Note that the verifier can compute the E' himself, so they do not need to be included in the proof. Therefore, a proof is of the form $\sigma = \{c^{(i)}, r^{(i)}\}_{i \in [k]}$. For λ bits of classical security, we need $N \approx 2^{2\lambda}$, so the total proof size is

$$k(2 + 2\lambda) \approx 1.26\lambda^2 \text{ bits}\,.$$

We can use this non-interactive proof as a signature scheme. However, if the goal is to obtain efficient signatures, it is possible to significantly reduce the signature size at the cost of increasing the size of the public keys.

Protocol with larger challenge space. In a nutshell, the idea is that instead of letting the public key be a single curve E, we let the public key consist of S curves $E_1 = [x_1]E_0, \ldots, E_S = [x_S]E_0$, where the $x_1, \ldots, x_S$ are the new secret key. The new sigma protocol is similar to that of Fig. 4, but has a challenge space $\{-S, \ldots, S\}$ (of size $2S+1$), and in response to challenge c, the prover sends a response r, such that $[r]E_c = E'$

if $(c \geq 0)$, or such that $[r]E^t_{-c} = E'$ in case $c \leq 0$. One can show in the random oracle model that a forger against this protocol can be turned into an algorithm that takes as input the curves $E_1, \ldots, E_S$ and outputs a triple (i, j, x) such that $1 \leq i < j \leq S$ and $[x]E_i = E_j$; this is believed to be a hard problem but it is not the problem of computing a witness for the relation $\mathcal{R}_{\text{CSIDH}}$ so the protocol is not a proof of knowledge for this relation. The advantage of this sigma protocol is that the challenge space is larger, so the protocol only needs to be repeated $\lambda/\log(2S+1)$ times for soundness error $2^{-\lambda}$. The signature size of the new signature is approximately

$$\frac{2}{\log(2S+1)}\lambda^2 \text{ bits}.$$

However, the size of the public key is now $4S\lambda$. So the parameter S gives a trade-off between small signatures (large S) and small public keys (small S). For more details (and a technique based on Merkle trees to reduce the size of the public key) we refer to the SeaSign or CSI-FiSh papers [DFG19,BKV19].

4.3 SeaSign sigma protocol

If the structure of the class group of $\mathbb{Z}[\sqrt{-p}]$ is not known, then we cannot efficiently compute the action of $\mathbb{Z}/N\mathbb{Z}$ on $\mathcal{E}\ell\ell$, so we cannot directly use the CSI-FiSh protocol. The naive way to solve this problem would be to just work with the action of $\mathbb{Z}^r$ instead: To prove knowledge of $\mathbf{x}$ such that $E = [\mathbf{x}]E_0$, the prover picks $\mathbf{b} \in [-B, B]^r$ uniformly at random and sends $[\mathbf{b}]E_0$ to the verifier, who responds with a challenge $c \in \{-1, 0, 1\}$, and then the prover sends his response $\mathbf{r} = \mathbf{b} - c\mathbf{x}$. Unfortunately, this sigma protocol is not zero knowledge, because in the case $c = 1$, the response is biased towards $-\mathbf{x}$, and if $c = -1$, the response is biased towards $\mathbf{x}$. After observing a number of executions of the protocol, an attacker could just compute the average of $-c\mathbf{r}$ to get a good estimate of $\mathbf{x}$.

In the CSI-FiSh case we chose b uniformly at random, so the response $r = b + cx \mod N$ does not reveal information about x. In the new protocol, since $\mathbb{Z}^r$ is infinite, we cannot choose $\mathbf{b}$ uniformly at random, so the response $\mathbf{r} = \mathbf{b} + c\mathbf{x}$ leaks information about $\mathbf{x}$.

One approach to the problem is to sample $\mathbf{b}$ from a box $[-\delta B, \delta B]^r$ where $\delta > 1$, which is larger than the box $[-B, B]^r$ from which the secret $\mathbf{x}$ is sampled. The hope is that if δB is much larger than B, then the distribution of $\mathbf{b} + \mathbf{x}$ is close to the distribution of $\mathbf{b}$, so it does not leak information about $\mathbf{x}$. Unfortunately, to make the two distributions indistinguishable δB would need to be exponentially larger than B. This is impractical because then evaluating the action of $[\mathbf{b}]E_0$ would take an exponential amount of time.

However the following observation can help us: The response $\mathbf{r} = \mathbf{b} - c\mathbf{x}$ can take values in $[-(\delta+1)B, (\delta+1)B]$, but $\mathbf{r}$ only leaks information about $\mathbf{x}$ if $\mathbf{r}$ is close to the boundary of this box. For example, if in the $c = 1$ case one of the coefficients r_i is equal to $-(\delta+1)B$, then this reveals that $x_i = -B$. Conversely, if $\mathbf{r}$ is sufficiently far away from the boundary, then it does not leak information: If $\mathbf{r} \in [-(\delta-1)B, (\delta-1)B]^r$ (which happens with probability $\left(\frac{2(\delta-1)B+1}{2\delta B+1}\right)^r \approx \left(1-\frac{1}{\delta}\right)^r$), then all values of $\mathbf{x}$ are consistent with $\mathbf{r}$, and the probability of seeing the response $\mathbf{r}$ is independent of $\mathbf{x}$, so $\mathbf{r}$ does not reveal any information about $\mathbf{x}$.

Using this observation, we can design a sigma protocol with aborts (a concept introduced by Lyubashevsky [Lyu09]) as follows: we pick δ large enough such that with reasonably large probability (e.g. at least 1/2), the response $\mathbf{r}$ lies in the "safe" box $[-(\delta-1)B, (\delta-1)B]^r$. Before the prover sends a response $\mathbf{r}$ they first check if $\mathbf{r}$ lies in $[-(\delta-1)B, (\delta-1)B]^r$. If this is the case, then the prover sends the response to the verifier, and otherwise, the prover aborts the sigma protocol to avoid leaking information. This way, we guarantee that the responses do not leak any information about $\mathbf{x}$. We refer to [DFG19] for asymptotic families of parameters that result in a polynomial-time protocol.

The protocol is summarized in Fig. 5. We can prove that the protocol aborts with probability ϵ close to $1-\left(1-\frac{1}{\delta}\right)^r$, that the protocol is correct (i.e. if in an honest execution the prover does not abort, then the verifier will accept), that the protocol has special soundness, and that the protocol has non-abort honest verifier zero-knowledge, meaning that non-aborting transcripts of the protocol can be simulated without knowledge of $\mathbf{x}$.

Prover$((E, \mathbf{x}))$		**Verifier**(E)
$\mathbf{b} \leftarrow [-\delta B, \delta B]^r$ $E' \leftarrow [\mathbf{b}]E_0$		
	$\xrightarrow{E'}$	
		$c \leftarrow \{-1, 0, 1\}$
	$\xleftarrow{c}$	
$\mathbf{r} \leftarrow \mathbf{b} - c\mathbf{x}$ **If** $\mathbf{r} \notin [-(\delta-1)B, (\delta-1)B]^r$**:** Prover aborts		
	$\xrightarrow{\mathbf{r}}$	

		If $c = -1$**:**
		return $E' = [\mathbf{r}]E^t$
		If $c = 0$**:**
		return $E' = [\mathbf{r}]E_0$
		If $c = 1$**:**
		return $E' = [\mathbf{r}]E$

Fig. 5. The SeaSign sigma protocol with abort.

4.4 SeaSign non-interactive proofs/signatures

Just like with a normal sigma protocol, the soundness of a sigma protocol with aborts can be amplified by repeating the protocol k times in parallel. In this case, since the protocol is 2-special sound we have $k = \lceil \lambda / \log 3 \rceil$. However, this increases the probability of an abort from ϵ to $1-(1-\epsilon)^k$. We can choose $\delta = kr$, such that the probability that none of the k repetitions of the sigma protocols abort is approximately $(1-\frac{1}{\delta})^{kr} \approx 1/e$.

Then, we can transform the amplified sigma protocol into a signature scheme with the Fiat-Shamir transform. If during the generation of a signature the prover aborts, then the signer can just restart the signing algorithm. As long as the success probability is not too small (e.g., $\approx 1/e$ if $\delta = kr$) the signing algorithm will succeed after a reasonable number of attempts.

Optimizing SeaSign. The technique of using multiple curves in the public key, which we described in Section 4.2 can also be used to reduce the signature size of SeaSign (at the cost of larger keys). In fact, this technique was introduced in the SeaSign paper.

Note that if $c = 0$, then the response is just $\mathbf{r} = \mathbf{b}$, which does not leak information about $\mathbf{x}$. Therefore, the prover does not need to abort if $\mathbf{r}$ lies outside of the "safe" box $[-(\delta-1)B, (\delta-1)B]^r$. This optimization reduces the rate of aborts, which means we can reduce δ, which in turn makes the signing algorithms faster (and the signatures slightly smaller). This optimization was described in a paper by Decru, Panny, and Vercauteren [DPV19], along with some additional optimizations which are beyond the scope of this survey.

5 The generic supersingular setting

We now move to settings that are specific to supersingular curves. We focus on the relations $\mathcal{R}_{\text{isog}}$ and $\mathcal{R}_{\text{deg}}$. As mentioned in Section 2.2 it suffices to consider elliptic curves defined over a finite field $\mathbb{F}_{p^2}$. Lacking a well behaved group action on the set of supersingular curves, we have to get more creative in order to define secure protocols.

Along with the SIDH key exchange, De Feo, Jao and Plût [DFJP14] sketched the first sigma protocol to prove knowledge of an isogeny between two supersingular curves over $\mathbb{F}_{p^2}$, provided p is an "SIDH prime". We start by presenting this simple protocol, which we refer to as DFJP. We highlight some issues with its soundness and zero-knowledge, and explain how to fix them, following De Feo, Dobson, Galbraith and Zobernig [DFDGZ22]. Then, we present a recent generalization of DFJP which applies to any characteristic and achieves statistical zero-knowledge [BCC$^+$23]. Finally, we discuss some open questions.

5.1 The DFJP protocol

We are in the SIDH setting, hence $p = 2^n 3^m f - 1$ and supersingular curves over $\mathbb{F}_{p^2}$ have group structure isomorphic to $(\mathbb{Z}/(p+1)\mathbb{Z})^2$. We can thus efficiently construct SIDH squares

$$\begin{tikzcd}
E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\
E_2 \arrow[r, "\phi'"] & E_3
\end{tikzcd} \tag{9}$$

where the degrees of ϕ and ψ are coprime (usually degrees 2^n and 3^m respectively).

Suppose a prover wants to prove knowledge of an isogeny $\phi : E_0 \to E_1$ of degree $d = 2^n$. The idea of DFJP is simply to choose a random ψ of degree co-prime to the degree of ϕ, commit to (E_2, E_3), and then reveal some, but not all, of ψ, ψ', ϕ'. They observe that revealing (ψ, ϕ') or (ψ', ϕ') is insecure, as that would immediately reveal the secret ϕ by pushing ϕ' (or its dual) through ψ (or ψ'). However they note that revealing ϕ' or (ψ, ψ') only appears to leak a limited amount of information on ϕ, and thus suggest the following protocol:

1. The prover chooses a random cyclic group $G = \ker(\psi)$ of order $D = 3^m$, sets $E_2 = E_0/G$ and $E_3 = E_1/\phi(G)$ and so constructs the commutative diagram (9), and sends (E_2, E_3) to the verifier;
2. The verifier challenges with a random bit $\mathsf{chall} \in \{0, 1\}$;
3. The prover responds with $(\ker(\psi), \ker(\psi'))$ if $\mathsf{chall} = 0$, and with $\ker(\phi')$ otherwise;
4. If $\mathsf{chall} = 0$ the verifier checks that $E_2 \cong E_0/\ker(\psi)$ and $E_3 \cong E_1/\ker(\psi')$, otherwise it checks that $E_3 \cong E_2/\ker(\phi')$.

There are two issues with the above idea. First, having binary challenges, it must be repeated λ times to achieve a soundness error of $2^{-\lambda}$, and is thus not particularly efficient. Second, as we explain next, it is not zero-knowledge, at least not for the $\mathcal{R}_{\text{deg}}$ relation.

Remark 4. DFJP represent ϕ (resp. ψ) by a generator K_ϕ (resp. K_ψ) of its kernel, and then represent ϕ' (resp. ψ') by $\psi(K_\phi)$ (resp. $\phi(K_\psi)$). This representation conveys more information than necessary, and makes the protocol provably less secure. We instead assume isogenies are represented by their whole kernel, which in practice is done by transmitting a kernel generator chosen at random, or deterministically in a way that only depends on the isogeny.

Zero Knowledge. The reason the DFJP protocol is not zero-knowledge is that the pair $(\ker(\psi), \phi(\ker(\psi)))$ revealed when $\mathsf{chall} = 0$ leaks enough information to recover the witness ϕ. Indeed, thanks to [FP22, Lemma 1], three such pairs are sufficient to compute a torsion basis $\langle P, Q\rangle = E_0[D]$ and points $P' = \lambda\phi(P)$, $Q' = \lambda\phi(Q)$ for some λ such that $\lambda^2 = 1 \bmod D$. But $\lambda = \pm 1$ because $D = 3^m$, hence the attacks of Castryck and Decru [CD23], Maino and Martindale [MMP+23], and Robert [Rob23] apply and recover ϕ.

Note that $D = 3^m$ is important here. If, instead, D is taken to contain many distinct prime factors, like in [FMP23], there are exponentially many possibilities for λ, and it is not currently known how to systematically apply the SIDH attacks to this case. Indeed, (roughly) following De Feo, Jao and Plût, we can prove that their protocol is zero-knowledge for the $\mathcal{R}_{\mathrm{M-SIDH}}$ relation, and is thus a non-trivial proof of knowledge for instances where $\mathcal{R}_{\mathrm{M-SIDH}}$ is still believed to be hard.

There is also a second, less dramatic issue associated to the $\mathsf{chall} = 1$ case. Indeed, the response $\ker(\phi') = \psi(\ker(\phi))$ appears to be correlated to ϕ, and thus hard to simulate without knowledge of the witness. DFJP simulate $\ker(\phi')$ with a randomly chosen group, thus reducing zero-knowledge to the hardness of the following computational problem, stating that it is difficult to distinguish pairs of "parallel" isogenies from random pairs of isogenies of the same degree.

Definition 4 (Decisional Supersingular Product Problem (DSSP)). *Let E_0 and E_1 be supersingular curves over $\mathbb{F}_{p^2}$ and let $\phi : E_0 \to E_1$ be an isogeny of degree d. Let D be an integer coprime to d. The DSSP problem is, knowing ϕ, to distinguish between the two following distributions:*

- *$D_0 = \{\phi' : E_2 \to E_3\}$, where $\psi : E_0 \to E_2$ is uniformly sampled among all cyclic D-isogenies starting from E_0, and $\ker(\phi') = \psi(\ker(\phi))$; and*
- *$D_1 = \{\phi' : E_2 \to E_3\}$, where $\psi : E_0 \to E_2$ is sampled as above, and ϕ' is a uniformly sampled cyclic d-isogeny starting from E_2.*

Soundness. DFJP claim their protocol is sound for the weaker relation $\mathcal{R}_{\mathrm{deg}}$. Unfortunately, this claim was independently shown to be wrong by De Feo, Dobson, Galbraith and Zobernig [DFDGZ22] and by Ghantous, Pintore and Veroni [GPV21], with explicit counterexamples presented in both papers.

It is possible, and in fact very simple, to show that DFJP is 2-special sound for $\mathcal{R}_{\mathrm{isog}}$, i.e. that it is a proof of knowledge of *an isogeny*, without any further qualifications.

Lemma 2. *The DFJP protocol is a 2-special-sound proof of knowledge for the relation $\mathcal{R}_{\mathrm{isog}} = \big\{\big((E_0, E_1), \phi\big) \,\big|\, \phi : E_0 \to E_1$ is an arbitrary isogeny$\big\}$.*

Proof. Consider an extractor that is given responses to two challenges for the same commitment (E_2, E_3). The extractor has isogenies $\psi : E_0 \to E_2$, $\psi' : E_1 \to E_3$ and the isogeny $\phi' : E_2 \to E_3$. Then the isogeny $\widehat{\psi'} \circ \phi' \circ \psi$ is an isogeny from E_0 to E_1.

Additionally, Ghantous, Pintore, and Veroni [GPV21] prove that, in some circumstances, DFJP is sound for $\mathcal{R}_{\mathrm{deg}}$ according to a weaker definition of 2-special soundness that assumes the existence of a witness. In summary DFJP is a sound protocol for $\mathcal{R}_{\mathrm{isog}}$, and is computationally ZK for $\mathcal{R}_{\mathrm{M-SIDH}}$, which may or may not be a hard relation. While this may be enough for some applications, e.g., signatures, it falls short in many ways.

5.2 A sigma protocol for $\boldsymbol{\mathcal{R}}_{\mathbf{deg}}$

De Feo, Dobson, Galbraith and Zobernig [DFDGZ22] modified the DFJP protocol to achieve both soundness and ZK for the relation $\mathcal{R}_{\mathrm{deg}}$.

The first fix concerns ZK: to avoid leaking the action of ϕ on $E_0[D]$, they move from binary to ternary challenges, revealing only one of ψ, ψ' or ϕ' at a time. The idea of ternary challenges for isogeny problems originates in the work of Boneh, Kogan and Woo [BKW20]. This is not sufficient: the commitment (E_2, E_3) still leaks information. To prevent this leakage they resort to a statistically hiding commitment scheme C, to securely hide the values of E_2 and E_3 until the response step.[6]

The second fix concerns soundness, and is more involved. The main obstacle to extracting ϕ in DFJP is that the three sides ψ, ψ', ϕ' of a diagram do not necessarily imply the existence of a fourth side of degree d:

$$\begin{tikzcd}
E_0 \arrow[rrr, dotted, "??"] \arrow[dr, "\psi"'] & & & E_1 \arrow[dl, "\psi'"] \\
& E_2 \arrow[r, "\phi'"] & E_3 &
\end{tikzcd}$$

The existence of ϕ parallel to ϕ' is guaranteed if and only if $\widehat{\psi}$ and $\widehat{\psi}'$ are proven to be parallel with respect to ϕ'. The key idea then is to "flip the SIDH square" and to treat $\phi' : E_2 \to E_3$ as the base for the square. By publishing torsion point information associated to E_2 and E_3, one can prove that $\widehat{\psi}$ and $\widehat{\psi}'$ are indeed parallel.

Putting all ideas together gives the following protocol (also see Figure 6):

1. The prover:
   - Chooses a random cyclic group $G = \ker(\psi)$ in E_0 of order D, and constructs the commutative diagram (9) (meaning: constructs $\phi' : E_2 \to E_3$ with $\ker(\phi') = \psi(\ker(\phi))$).
   - Chooses a random basis P_2, Q_2 of $E_2[D]$, computes $P_3 = \phi'(P_2), Q_3 = \phi'(Q_2)$.
   - Computes integers (a, b) such that $\ker(\widehat{\psi}) = \langle [a]P_2 + [b]Q_2 \rangle$.
   - Sends commitments $\mathsf{C}_L = \mathsf{C}(E_2, P_2, Q_2; r_L)$, $\mathsf{C}_R = \mathsf{C}(E_3, P_3, Q_3; r_R)$ and $\mathsf{C} = \mathsf{C}(a, b; r)$ to the verifier;
2. The verifier challenges with a random value $\mathsf{chall} \in \{-1, 0, 1\}$;

[6] Such a commitment scheme can be easily instantiated as $\mathsf{C}(m; r) = H(m \| r)$, where H is a hash function and r is a sufficiently long random string.

3. The prover opens:
   - C_L and C if $\mathsf{chall} = -1$,
   - C_R and C if $\mathsf{chall} = 0$,
   - C_L and C_R if $\mathsf{chall} = 1$, and additionally sends $\ker(\phi')$. Note that in practice one usually sends a generator for $\ker(\phi')$, in which case this generator should be sampled uniformly from the set of all generators of the subgroup;
4. The verifier checks that the opened commitments are well formed, and:
   - that $E_0 = E_2/\langle [a]P_2 + [b]Q_2 \rangle$ if $\mathsf{chall} = -1$,
   - that $E_1 = E_3/\langle [a]P_3 + [b]Q_3 \rangle$ if $\mathsf{chall} = 0$,
   - that $E_3 = E_2/\ker(\phi')$ and that $P_3 = \phi'(P_2), Q_3 = \phi'(Q_2)$ if $\mathsf{chall} = 1$.

Prover$((E_0, E_1, d, D), \phi)$		**Verifier**(E_0, E_1, d, D)
Choose random order-D cyclic group $G \subseteq E_0$ Define $\psi : E_0 \to E_2$ with kernel G and construct the diagram (9) Choose a random basis P_2, Q_2 of $E_2[D]$ Set $P_3 \leftarrow \phi'(P_2), Q_3 \leftarrow \phi'(Q_2)$ Compute integers (a, b) such that $\ker(\widehat{\psi}) = \langle [a]P_2 + [b]Q_2 \rangle$ Choose random r_L, r_R, r Set $\mathsf{C} \leftarrow \mathsf{C}(a, b; r)$, $\mathsf{C}_L \leftarrow \mathsf{C}(E_2, P_2, Q_2; r_L)$, $\mathsf{C}_R \leftarrow \mathsf{C}(E_3, P_3, Q_3; r_R)$		
	$\xrightarrow{\mathsf{C}_L, \mathsf{C}_R, \mathsf{C}}$	
		$\mathsf{chall} \leftarrow \{-1, 0, 1\}$
	$\xleftarrow{\mathsf{chall}}$	
If $\mathsf{chall} = -1$: $\mathsf{resp} \leftarrow$ opening of C_L, C **If** $\mathsf{chall} = 0$: $\mathsf{resp} \leftarrow$ opening of C_R, C **If** $\mathsf{chall} = 1$: $\mathsf{resp} \leftarrow$ opening of $\mathsf{C}_L, \mathsf{C}_R$, and $\ker(\phi')$		
	$\xrightarrow{\mathsf{resp}}$	

		Check that opened commitments are well formed, and (P_2, Q_2) and/or (P_3, Q_3) are D-torsion bases, and $\gcd(D, a, b) = 1$
		If $\mathsf{chall} = -1$:
		  return $E_0 = E_2/\langle [a]P_2 + [b]Q_2 \rangle$
		If $\mathsf{chall} = 0$:
		  return $E_1 = E_3/\langle [a]P_3 + [b]Q_3 \rangle$
		If $\mathsf{chall} = 1$:
		  Check that $\# \ker(\phi') = d$
		  Construct ϕ' from $\ker(\phi')$
		  return $E_3 = E_2/\ker(\phi')$
		    and $P_3 = \phi'(P_2)$
		    and $Q_3 = \phi'(Q_2)$

Fig. 6. The sigma protocol for $\mathcal{R}_{\text{deg}}$.

Proposition 1. *The protocol of Figure 6 is a computationally ZK 3-special sound proof of knowledge for the relation*

$$\mathcal{R}_{\text{deg}} = \left\{ \left((E_0, E_1, d), \phi \right) \,\middle|\, \phi : E_0 \to E_1 \textit{ is an isogeny of degree } d \right\},$$

assuming DSSP is hard and C *is a statistically hiding and computationally binding commitment scheme.*

Proof. (Sketch) Correctness is similar to DFJP. The only additional property to check is that, if the SIDH square is generated honestly, one can efficiently find integers (c, d) such that $\ker(\widehat{\psi}) = \langle [a]P_2 + [b]Q_2 \rangle$ and $\ker(\widehat{\psi}') = \langle [a]P_3 + [b]Q_3 \rangle$. A generator for $\ker(\widehat{\psi})$ can be found by pushing $E_0[D]$ through ψ, then the integers (a, b) can be computed by solving a generalized discrete logarithm in $E_2[D]$, which is easy because D is smooth. Then $\ker(\widehat{\psi}') = \langle [a]P_3 + [b]Q_3 \rangle$ follows from the fact that ψ and ψ' are parallel, and thus $\ker(\widehat{\psi}') = \phi'(\ker(\widehat{\psi}))$.

Zero Knowledge. The simulator for the case $\mathsf{chall} = -1$ picks a random isogeny $\psi : E_0 \to E_2$, a random basis (P_2, Q_2), computes (a, b), and finally computes the commitments C_L and C as in the protocol. As the commitment C_R will not be opened, it is replaced by a random value.

The case $\mathsf{chall} = 0$ is nearly identical, but with the goal of ensuring C_R and C can be opened in the protocol.

Finally, the case $\mathsf{chall} = 1$ is simulated by taking a random $\psi : E_0 \to E_2$, then a random $\phi' : E_2 \to E_3$ not necessarily parallel to ϕ, a random basis $\langle P_2, Q_2 \rangle = E_2[D]$, and points $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$. The commitment C will not be opened and is replaced by a random value. Like in DFJP, this part of the simulation is only indistinguishable from the real protocol assuming DSSP is hard.

Soundness. Because the protocol uses ternary challenges, we show it is 3-special sound. Since the commitment scheme C is computationally binding, the openings of the commitments match in each of the three valid transcripts. Hence, we extract $\widehat{\psi}$, $\widehat{\psi}'$ and ϕ', with the additional property that $\ker(\widehat{\psi}') = \phi'(\ker(\widehat{\psi}))$. Hence $\widehat{\psi}$ and $\widehat{\psi}'$ are parallel, proving the existence of a d-isogeny ϕ with kernel $\widehat{\psi}(\ker(\phi'))$ parallel to ϕ'. □

Note that a cheating prover can correctly answer any two of the challenges without knowing the witness, so the soundness error for this protocol is 2/3.

Sigma protocols for $\mathcal{R}_{\text{SIDH}}$ *and* $\mathcal{R}_{\text{M}-\text{SIDH}}$. In the same work [DFDGZ22], De Feo, Dobson, Galbraith and Zobernig further modify the previous protocol to achieve a ZK proof of knowledge for the relation $\mathcal{R}_{\text{SIDH}}$, i.e. the knowledge of an SIDH secret key. The idea is, roughly, to run two correlated instances of the protocol, and to let the verifier check that they are both consistent with the torsion point information that is part of the public key. While this protocol is not anymore relevant for SIDH/SIKE, a simple tweak yields a proof of knowledge for $\mathcal{R}_{\mathrm{M-SIDH}}$ (see Basso [Bas23]), which is non-trivial in cases where $\mathcal{R}_{\mathrm{M-SIDH}}$ is thought to be hard.

5.3 From computational to statistical zero-knowledge

The last obstacle standing between the protocol above and a fully zero-knowledge one is the DSSP problem. To avoid such an assumption we would need to be able to simulate the distribution of isogenies $\phi' : E_2 \to E_3$ parallel to the witness $\phi : E_0 \to E_1$, without having the knowledge of ϕ. This is difficult for the protocol of Figure 6 because the pair (E_2, E_3) is far from being well distributed among all pairs of supersingular curves. More precisely, the isogenies ϕ and ϕ' are parallel with respect to an isogeny ψ of degree $D \ll p$. Because D is so small, ψ is almost always uniquely determined by E_2, and so is ϕ'. Hence, after having chosen E_2, computing the right ϕ' is as hard as computing ϕ, thus we cannot expect a simulator to be able to do it, if we believe $\mathcal{R}_{\mathrm{deg}}$ is hard.

In a recent work [BCC$^+$23], a host of authors introduce a modification to the ternary-challenge DFJP protocol that makes the distribution of $\phi' : E_2 \to E_3$ easy to simulate. The intuition is quite simple: by the expansion properties of the isogeny graph, if we increase the degree D of ψ, the number of isogenies $E_0 \to E_2$ also increases. Eventually, we expect the number to become so large that any isogeny ϕ' of degree d starting from E_2 becomes parallel to ϕ for many isogenies ψ.

To prove this formally, [BCC$^+$23] defines a new 3-isogeny graph whose vertices are pairs (E, G), where E is a supersingular curve and $G \subset E$ a cyclic subgroup of order d. For example, $(E_0, \ker(\phi))$ represents ϕ (up to post-composition with an isomorphism), and $(E_2, \ker(\phi'))$ represents ϕ'. Two vertices (E, G) and (E', G') are connected if there is a 3-isogeny $\psi : E \to E'$ such that $\psi(G) = G'$. Such graphs are called *isogeny graphs with Borel level structure* in [BCC$^+$23], and it is proved they are Ramanujan, which is exactly what is needed to make the intuition above work.

Putting it all together yields the "meta-protocol" of Figure 7. For technical reasons that will be explained below, this protocol cannot use the "flipping the SIDH square" trick of Figure 6, and is thus only a proof of knowledge for $\mathcal{R}_{\mathrm{isog}}$, like the DFJP protocol. However, the response to challenge $\mathsf{chall} = 1$ contains the degree $d = \deg(\phi)$, thus the protocol can only be zero-knowledge for $\mathcal{R}_{\mathrm{deg}}$, as the authors show.

Proposition 2. *The protocol of Figure 7 is a 3-special sound proof of knowledge for $\mathcal{R}_{\mathrm{isog}}$, assuming C is computationally binding. Furthermore, if C is statistically hiding, there exists an explicit constant $\gamma > 1$, depending only on the security parameter, such that it is statistically zero-knowledge for $\mathcal{R}_{\mathrm{deg}}$.*

One obstacle to efficient implementation of this idea in practice is that the degree $D = 3^m$ is way too large for the curves to have rational points of order D defined over $\mathbb{F}_{p^2}$, or even

```
$\mathbf{Prover}((E_0, E_1, d, \gamma, \phi))$                                    $\mathbf{Verifier}(E_0, E_1, d, \gamma)$

Let $m$ be such that $m/\log(m) \geq \gamma p d$
Choose random cyclic group $G \subseteq E_0$
of order $D = 3^m$
Define $\psi : E_0 \to E_2$ with kernel $G$ and construct the diagram (9)
Choose random $r_L$, $r_R$
$\mathsf{C}_L \leftarrow \mathsf{C}(E_2; r_L)$, $\mathsf{C}_R \leftarrow \mathsf{C}(E_3; r_R)$

                              $\xrightarrow{\mathsf{C}_L, \mathsf{C}_R}$
                                                   $\mathsf{chall} \leftarrow \{-1, 0, 1\}$
                              $\xleftarrow{\mathsf{chall}}$
If $\mathsf{chall} = -1$:
    $\mathsf{resp} \leftarrow$ opening of $\mathsf{C}_L$, and $\ker(\psi)$
If $\mathsf{chall} = 0$:
    $\mathsf{resp} \leftarrow$ opening of $\mathsf{C}_R$, and $\ker(\psi')$
If $\mathsf{chall} = 1$:
    $\mathsf{resp} \leftarrow$ opening of $\mathsf{C}_L, \mathsf{C}_R$, and $\ker(\phi')$
                              $\xrightarrow{\mathsf{resp}}$
                                                   Check that opened commitments are well
                                                   formed and kernels have the expected size
                                                   If $\mathsf{chall} = -1$:
                                                       return $E_2 = E_0/\ker(\psi)$
                                                   If $\mathsf{chall} = 0$:
                                                       return $E_3 = E_1/\ker(\psi')$
                                                   If $\mathsf{chall} = 1$:
                                                       return $E_3 = E_2/\ker(\phi')$
```

Fig. 7. A "meta-protocol" for $\mathcal{R}_{\text{isog}}$.

over an extension field of polynomial degree. Instead of using a single kernel generator to represent and compute isogenies, [BCC^{+}23] observes that SIDH squares can be efficiently "glued" together to form what they call *SIDH ladders*. The protocol response provides information about the intermediate curves and isogenies in the left or right hand sides of the ladder. This is the reason why the "flipping the SIDH square" idea cannot be used anymore. We picture such a ladder in Figure 8.

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

Fig. 8. An SIDH ladder.

Additionally, one may glue SIDH squares both vertically and horizontally, thus freeing the protocol from the constraint of having the isogeny degrees divide $p + 1$.

5.4 SIDH signatures

The Fiat-Shamir heuristic can be used to make sigma protocols non-interactive. The basic DFJP protocol from Section 5.1 has been used to construct a signature scheme in Yoo et al [YAJ^{+}17] and [GPS20], based on the hardness of DSSP and SIDH key exchange. These schemes are no longer secure, due to the attacks on SIDH, but it is fairly straightforward to construct secure signatures based on any of the protocols with ternary challenges above.

5.5 Questions and perspectives

We now discuss some of the limitations of the protocols above.

Efficiency. A 3-special sound protocol comes at a cost: to achieve soundness error of $2^{-\lambda}$, one needs $\lambda/(\log_2(3) - 1)$ iterations, each iteration computing one or more SIDH squares. A major open problem is whether there exist protocols for any of these relations with small soundness error, possibly exponentially small. SQISign, presented in Section 6.2, will provide a beginning of an answer for the relation $\mathcal{R}_{\text{isog}}$.

Statistical ZK for $\mathcal{R}_{\text{deg}}$. Currently, there appears to be a tension between soundness and zero-knowledge for $\mathcal{R}_{\text{deg}}$: the "flipping the SIDH square" technique of Section 5.2 is incompatible with the "SIDH ladders" of Section 5.3. Indeed, in an SIDH ladder the D-torsion is not anymore defined over the base field, and it is thus not possible to define the torsion bases $(P_2, Q_2), (P_3, Q_3)$ used to prove special soundness for $\mathcal{R}_{\text{deg}}$.

It is an interesting question to find alternative representations for D-torsion bases that are compatible with SIDH ladders.

ZK for $\mathcal{R}_{\text{isog}}$ All the protocols we presented so far appear to leak $\deg(\phi)$ to the verifier when responding with the isogeny ϕ' parallel to ϕ. It seems thus difficult to make them zero-knowledge for $\mathcal{R}_{\text{isog}}$, rather than $\mathcal{R}_{\text{deg}}$.

6 GPS and SQISign

The GPS and SQISign protocols we describe in this section work with the full supersingular isogeny graph (unlike the CSIDH-based protocols of Section 2.4 which only consider curves defined over $\mathbb{F}_p$), and use the quaternion algorithms described in Section 2.5 to design a GMW-like protocol which does not rely on a SIDH diagram. The GPS protocol provides steps towards a general solution to proving the relation $\mathcal{R}_{\text{isog}}$. In particular, the GPS protocol achieves statistical ZK, and no auxiliary points or other information are needed to obtain zero-knowledge.

6.1 GPS sigma protocol

The GPS protocol, due to Galbraith, Petit, and Silva [GPS20], heavily uses the ideas of Section 2.5. In particular, we assume that E_0 is a special supersingular curve such as $y^2 = x^3 + x$. We wish to prove knowledge of an isogeny $\phi : E_0 \to E_1$, and will use properties of $\text{End}(E_0)$. So our protocol proves $\mathcal{R}_{\text{isog}}$ for any field, but in the special case where E_0 is a curve with known endomorphism ring.

To prove knowledge of an isogeny between two "arbitrary" curves E' and E'' one can apply this protocol if one knows an isogeny from E_0 to E' (and hence one can construct an isogeny from E_0 to E''). However, if E' and E'' are curves whose endomorphism ring is not known to the prover then the methods of this section cannot be applied.

The sigma protocol is as follows. First, fix parameters B, N_1, N_2 such that $N_k = \prod_i \ell_{k,i}^{e_{k,i}}$, where $\ell_{k,i}^{e_{k,i}} < B$, $\gcd(N_1, N_2) = 1$, $\log(N_2) > \frac{7}{2}\log(p)$, and for each $k \in \{1, 2\}$

$$\prod_i \left(\frac{2\sqrt{\ell_{k,i}}}{\ell_{k,i} + 1} \right)^{e_{k,i}} < (p^{1+\epsilon})^{-1}.$$

This last formula is needed for uniform mixing of random walks in the isogeny graph. We let I be the left $\mathcal{O}_0$-ideal corresponding to the secret isogeny $\phi : E_0 \to E_1$.

To construct the commitment com, perform a random isogeny walk of degree N_1 from the curve E_1 to a curve E_2 and set $\mathsf{com} = j(E_2)$. The isogeny $\psi : E_1 \to E_2$ can be represented in many ways (e.g., kernel points of order $\ell_i^{e_i}$ or a sequence of j-invariants). Let J be the left-End(E_1)-ideal corresponding to ψ. Let the challenge be $\mathsf{chall} \in \{0, 1\}$. When $\mathsf{chall} = 0$ respond with ψ. When $\mathsf{chall} = 1$ the KLPT algorithm is needed. We compute the ideal IJ, which corresponds to $\psi \circ \phi$. Then run KLPT to get an equivalent $\mathcal{O}_0$-ideal J' of norm N_2. The response is the isogeny $\psi' : E_0 \to E_2$ corresponding to J'. The verifier checks that the response is an isogeny from $E_{1-\mathsf{chall}}$ to E_2. The protocol is repeated until the verifier is convinced.

The special soundness of the protocol is easy to prove. Given valid responses ψ' and ρ to the challenges 0 and 1 for the same commitment, the extractor can compute an isogeny $\rho \circ \hat{\psi}'$ from E_0 to E_1. Note that this isogeny is a priori not the same one used by the prover to create the responses, but this is irrelevant to special soundness[7].

The three key requirements for this to be zero-knowledge and practical are that:

1. E_2 is close to uniformly distributed in the isogeny graph.
2. The isogeny ψ' is independent of ϕ.
3. The isogenies ψ and ψ' in the response have a compact representation and can be computed efficiently.

The first two properties are needed for zero-knowledge. The simulator, who knows the challenge but who does not know ϕ or I, will behave as in the honest protocol for the case $\mathsf{chall} = 0$, but when $\mathsf{chall} = 1$ will take a random isogeny from E_0 to E_2 and then run KLPT to get an ideal J' as in the real protocol. It is necessary that the curves E_2 generated by the simulator when $\mathsf{chall} = 1$ are distributed close to identically as in the original protocol. This is the purpose of the first requirement. It is also necessary that the isogenies ψ' are distributed identically in both cases. We refer to [GPS20] for the details.

As the protocol uses one-bit challenges, it must be repeated λ times to obtain a scheme with $2^{-\lambda}$ soundness error. The protocol can be made non-interactive and used as a signature scheme by the Fiat-Shamir transform. While polynomial time in theory, the resulting signature scheme is considered impractical.

[7] Additionally, both isogenies can in fact be mapped to the same "canonical" one (for example, using LLL to compute a minimal norm ideal in the ideal class, followed if needed by some deterministic version of KLPT to get a powersmooth norm ideal).

6.2 SQISign

A key source of inefficiency in GPS signatures (and other signatures based on isogenies) is the need to repeat the zero-knowledge protocol multiple times to reduce the soundness error. This comes from the fact that the protocol has single bit challenges.

To increase the challenge space, the SQISign protocol by De Feo, Kohel, Leroux, Petit, and Wesolowski [DFKL$^+$20] modifies the basic GMW-based protocol as follows. Given a secret isogeny $\phi : E_0 \to E_1$, the prover first computes a random isogeny $\psi : E_0 \to E_2$ and commits to E_2. Then instead of challenging the prover with a single bit, the verifier computes and sends a third random isogeny $\varphi : E_2 \to E_3$, sends it to the prover, and challenges the prover to compute an isogeny $\sigma : E_1 \to E_3$ (see Figure 9).

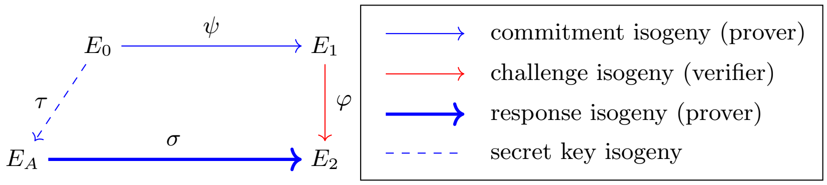
E_0
ψ
E_1
φ
τ
σ
E_A
E_2
commitment isogeny (prover)
challenge isogeny (verifier)
response isogeny (prover)
secret key isogeny

Fig. 9. A picture of SQIsign's identification protocol [DFKL$^+$20]

A naive version of this protocol is not secure: A dishonest prover could compute the commitment by computing a random isogeny ψ' from E_1. Call E_2 the image curve. Since the verifier sends an isogeny φ the prover can respond with $\varphi \circ \psi' : E_1 \to E_3$. The way to prevent this is for the verifier to check that $\hat{\varphi} \circ \sigma$ has cyclic kernel; we refer to [DFKL$^+$20] for the details. The protocol also imposes conditions on the isogeny $\phi : E_0 \to E_1$, namely that E_0 has known endomorphism ring and that ϕ has "medium" prime degree. Finally, the protocol is only computationally ZK based on an ad hoc assumption. Hence SQISign is currently far from a general solution to the relation $\mathcal{R}_{\text{isog}}$.

Special soundness relies on the problem of computing an endomorphism of E_1, a problem equivalent to computing an isogeny between two random supersingular curves [GPS20,EHL$^+$18,Wes22].

Indeed, given two valid responses to two challenges for the same commitment, an extractor can compose both challenges and responses in an appropriate way to compute an endomorphism of E_1 (see [DFKL$^+$20]).[8]

The astute reader will have noticed that computing σ now requires to re-randomize an isogeny between two random supersingular curves, whereas the tools described in Section 2.5 assumed one of the curves was "special", i.e. with known and very special endomorphism ring. One can trivially generalize these tools to the general case (in fact this was

[8] A similar approach in the case of graph isomorphism would provide the extractor with an automorphism of one graph. This does not immediately solve the graph isomorphism problem.

already done in [KLPT14]), but in a way that, used in the above signature scheme, will always leak the secret (the isogeny σ will always go through the curve E_0). The key contribution in [DFKL$^+$20] is a new generalization of the KLPT algorithm which conjecturally avoids this problem.

SQISign signatures are an order of magnitude smaller than all other post-quantum isogeny-based signature schemes in the signature-plus-public-key metric. Key generation and verification time are reasonable (at 0.6s and 50ms respectively) but signing takes 2.5s, mostly because of the conversions between isogenies and ideals (which are polynomial time but slow in practice). An improved version of the protocol obtained a more than two-fold speedup on signing time [DFLLW23]. It is an open question whether SQISign can be fast enough for many applications.

Further work should aim at improving the signing time, but also to address some outstanding security concerns:

- SQISign is not known to be secure in the quantum oracle model. This is because the sigma protocol does not have any of the properties that allow proving the security of Fiat-Shamir signatures in this model [KLS18]. Possible solutions include switching to the Unruh transform [Unr15] (at some efficiency cost) or developing an ad hoc proof.
- The SQISign security proof does not provide an extractor that outputs an isogeny from E_0 to E_1. So it is not technically a proof of knowledge for $\mathcal{R}_{\text{isog}}$. However, the extractor does return a "random" endomorphism on E_1.

  Heuristically, $O(1)$ independent endomorphisms should generate the whole endomorphism ring, and one can build [EHL$^+$20] a k-special sound extractor that computes the whole endomorphism ring (for $k = O(1)$). Proving this formally remains an open problem. Once $\mathrm{End}(E_1)$ is known then an isogeny from E_0 to E_1 can be computed [Wes22].
- Some issues with the zero-knowledge of SQISign were pointed out in [DFLLW23], and tweaks to the algorithms to avoid the issue are provided.
- The security definition for computational honest verifier zero-knowledge in Definition 2 allows the distinguisher to be provided with a witness, but the arguments for the computational honest verifier zero-knowledge property in [DFKL$^+$20] do not apply in this case. Indeed with a witness one can easily distinguish between the response isogeny and a random isogeny of the same degree between the same two curves (as a quick study of [DFKL$^+$20, Figure 3] shows). One approach to solve this problem could be to randomize the output of the *EquivalentPrimeIdeal* and/or the particular secret isogeny τ used in the generalized KLPT algorithm [DFKL$^+$20]. However, both approaches will increase the norm of the ideal that is output by this algorithm, and hence they will further slow down signature generation. We also leave the security analysis of these approaches to further work.

7 Conclusion and open problems

We have explained that proving knowledge of an isogeny is an important problem with several strong motivations in post-quantum cryptography. We have given a number of sigma protocols for different relations and contexts. Generally, the case of group actions is simpler than the more general isogeny problems.

One of the biggest open problems in this area is to develop more efficient protocols by lowering the soundness error. In the CSIDH setting, we have signature schemes that have improved soundness error, but those approaches do not solve the problem for the original relation $\mathcal{R}_{\text{isog}}$ we are interested in. The only example of a protocol with negligible soundness error for a single round is SQISign. Any progress on reducing the soundness error would have major implications on the efficiency of isogeny signatures.

Another major open problem is to design a protocol for the relation $\mathcal{R}_{\text{isog}}$ that can be applied when $\text{End}(E)$ is not known and that does not leak the degree of the witness. The GPS scheme [GPS20] only works when $\text{End}(E)$ is known, while SQISign is currently not a proof of knowledge.

Further open questions that we have discussed in the paper include: How effective are general tools for ZK proofs when applied to isogeny problems? Can zero-knowledge of SQISign (or a variant of it) be proved based on more standard assumptions? Are there ways to get signatures with a tight reduction to (relatively) standard assumptions?

References

AFKM05. C. Adams, S. Farrell, T. Kause, and T. Mononen. Internet X.509 public key infrastructure Certificate Management Protocol (CMP). https://www.rfc-editor.org/rfc/rfc4210, 2005. 1

Bas23. Andrea Basso. A post-quantum round-optimal oblivious prf from isogenies. eprint 2023/225, 2023. 5.2

BCC$^+$23. Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437. Springer, 2023. 1, 5, 5.3, 5.3

BCR$^+$19. Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128. Springer, 2019. 3.2

BDFLS20. Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *The Open Book Series*, 4(1):39–55, 2020. 2.1

BFPW07. Alexandra Boldyreva, Marc Fischlin, Adriana Palacio, and Bogdan Warinschi. A closer look at PKI: Security and efficiency. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 458–475. Springer, 2007. 1

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019. 1, 4, 4.1, 4.2

BKW20. Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 520–550. Springer, 2020. 5.2

CD20. Wouter Castryck and Thomas Decru. CSIDH on the surface. In Jintai Ding and Jean-Pierre Tillich, editors, *PQCrypto 2020*, volume 12100 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 2020. 2.4

CD23. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Cham, 2023. Springer. 1, 2.3, 5.1

CLG09. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptol.*, 22(1):93–113, 2009. 2.2, 3.2

CLL23. Kelong Cong, Yi-Fu Lai, and Shai Levin. Efficient isogeny proofs using generic techniques. In Mehdi Tibouchi and Xiaofeng Wang, editors, *ACNS 2023*, volume ?? of *LNCS*, page ?? Springer, eprint 2023/037, 2023. 3.2

CLM$^+$18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, *ASIACRYPT 2018*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427, Cham, 2018. Springer. 2.4, 4, 4.1

Cos20. Craig Costello. B-SIDH: Supersingular Isogeny Diffie-Hellman using twisted torsion. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463, Cham, 2020. Springer International Publishing. 2.3

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. eprint 2006/291, 2006. 2.4, 4.1

CSRHT22. Jorge Chavez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography*, pages 441–460, Cham, 2022. Springer International Publishing. 3.2, 3.2, 3.2

DFDGZ22. Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Proceedings, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 310–339. Springer, 2022. 5, 5.1, 5.2, 5.2

DFG19. Luca De Feo and Steven D. Galbraith. SeaSign: compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 759–789. Springer, 2019. 1, 4.1, 4.2, 4.3

DFJP14. 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. 1, 2.3, 5

DFKL$^+$20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020. 1, 2.5, 6.2, 9, 6.2

DFKS18. Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 365–394. Springer, 2018. 2.4

DFLLW23. Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the Deuring correspondence - towards practical and secure SQISign signatures. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023 Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023. 2.5, 6.2

DPV19. Thomas Decru, Lorenz Panny, and Frederik Vercauteren. Faster SeaSign signatures through improved rejection sampling. In Jintai Ding and Rainer Steinwandt, editors, *PQCrypto 2019*, volume 11505 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2019. 4.4

EHL^{+}18. Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018*, volume 10822 of *Lecture Notes in Computer Science*, pages 329–368, 2018. 2.5, 6.2

EHL^{+}20. Kirsten Eisenträger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. In *Proceedings of the Fourteenth Algorithmic Number Theory Symposium*, volume 4 of *Open Book Ser.*, pages 215–232. Math. Sci. Publ., Berkeley, CA, 2020. 6.2

FMP23. Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: Countering SIDH attacks by masking information. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 282–309, Cham, 2023. Springer. 1, 2.3, 5.1

FP22. Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on SIDH. In Steven D. Galbraith, editor, *Topics in Cryptology – CT-RSA 2022*, volume 13161 of *Lecture Notes in Computer Science*, pages 322–344, Cham, 2022. Springer International Publishing. 5.1

Gal12. Steven D. Galbraith. *Mathematics of Public Key Cryptography.* Cambridge University Press, 2012. 2.1, 1, 2.1

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. 1, 3.2, 3.3

Gol01. Oded Goldreich. *Foundations of Cryptography: Basic tools.* Cambridge, 2001. 3.1

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. 1, 2.5, 5.4, 6.1, 6.1, 6.2, 7

GPST16. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91, 2016. 1

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`. 5.1, 5.1

JAC^{+}17. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017. 1, 2.3

JDF11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34, Berlin, Heidelberg, 2011. Springer. 2.3

KLPT14. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17A:418–432, 2014. 2.5, 6.2

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586. Springer, 2018. 3.4, 6.2

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, pages 177–194, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 3.2

LFKN92. Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, oct 1992. 3.2

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009. 4.3

Mic00. Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. 3.2

MMP$^+$23. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Cham, 2023. Springer. 1, 2.3, 5.1

PS18. Christophe Petit and Spike Smith. An improvement to the quaternion analogue of the l-isogeny path problem. Poster presentation at MathCrypt2018, 2018. 2.5

Rob23. Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503, Cham, 2023. Springer. 1, 2.3, 5.1

RS06. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptol. ePrint Arch.*, page 145, 2006. 2.4, 4.1

Sil09. Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, Dordrecht, second edition, 2009. 2.1, 1, 2.2

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015. 3.4, 6.2

Vél71. Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:A238–A241, 1971. 2.1

Voi21. John Voight. *Quaternion Algebras*, volume 288. Springer Graduate Text Math., 2021. 2.2

Was08. Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography (2nd edition)*. CRC Press, 2008. 2.1

Wat69. William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 2(4):521–560, 1969. 2.4, 2.5

Wes22. Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111, 2022. 6.2

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*, volume 10322 of *Lecture Notes in Computer Science*, pages 163–181. Springer, Springer, 2017. 5.4