Supersingular Curves You Can Trust

Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca de Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, Benjamin Wesolowski

Supersingular Curves You Can Trust

Andrea Basso[1,2], Giulio Codogni[3], Deirdre Connolly[4], Luca De Feo[5], Tako Boris Fouotsa[6], Guido Maria Lido[3], Travis Morrison[7], Lorenz Panny[8], Sikhar Patranabis[9], and Benjamin Wesolowski[10,11,12]

[1] University of Birmingham, Birmingham, United Kingdom
[2] University of Bristol, Bristol, United Kingdom; andrea.basso@bristol.ac.uk
[3] Dipartimento di Matematica, Università degli Studi di Roma Tor Vergata, Via della Ricerca Scientifica, 00133 Roma, Italy; codogni@mat.uniroma2.it, guidomaria.lido@uniroma2.it
[4] Zcash Foundation; durumcrustulum@gmail.com
[5] IBM Research Europe, Zürich, Switzerland; secuer@defeo.lu
[6] EPFL, Lausanne, Switzerland; tako.fouotsa@epfl.ch
[7] Virginia Tech, Blacksburg, Virginia, USA; tmo@vt.edu
[8] Academia Sinica, Taipei, Taiwan; lorenz@yx7.cc
[9] IBM Research India, Bangalore, India; sikhar.patranabis@ibm.com
[10] Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France; benjamin.wesolowski@math.u-bordeaux.fr
[11] INRIA, IMB, UMR 5251, F-33400, Talence, France
[12] ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France

Abstract. Generating a supersingular elliptic curve such that nobody knows its endomorphism ring is a notoriously hard task, despite several isogeny-based protocols relying on such an object. A trusted setup is often proposed as a workaround, but several aspects remain unclear. In this work, we develop the tools necessary to practically run such a distributed trusted-setup ceremony.

Our key contribution is the first statistically zero-knowledge proof of isogeny knowledge that is compatible with any base field. To prove statistical ZK, we introduce isogeny graphs with Borel level structure and prove they have the Ramanujan property. Then, we analyze the security of a distributed trusted-setup protocol based on our ZK proof in the simplified universal composability framework. Lastly, we develop an optimized implementation of the ZK proof, and we propose a strategy to concretely deploy the trusted-setup protocol.

Keywords: Isogenies · Ramanujan Graphs · Zero-knowledge Proofs · Trusted Setup

1 Introduction

Be it foundationally or for efficiency, most of isogeny-based cryptography is built upon supersingular elliptic curves [CLG09, JD11, CLM+18, DMPS19, GPS20, DKL+20, DdF+21]. At the heart of it, lies the *supersingular isogeny graph*: a graph whose vertices represent supersingular elliptic curves (up to isomorphism) and whose edges represent isogenies (up to isomorphism) of some fixed small prime degree between them. A foundational hard problem for isogeny-based cryptography is then: given two supersingular elliptic curves, find a path in the supersingular isogeny graph connecting them.

An endomorphism is an isogeny from a curve E to itself, and their collection forms the *endomorphism ring* $\mathrm{End}(E)$. In recent years, the connection between finding isogeny paths and computing endomorphism rings of supersingular curves has become increasingly important [GPST16, EHL+18, Wes22b, Wes22a]. It is now established that, assuming the generalized Riemann hypothesis, there exists probabilistic polynomial time algorithms for these two problems:

* Author list in alphabetical order; see https://ams.org/profession/leaders/CultureStatement04.pdf. This work began at the Banff International Research Station workshop "Supersingular Isogeny Graphs in Cryptography" (21w5229). This research was funded in part by the MIUR Excellence Department Project MatMod@TOV awarded to the Department of Mathematics, University of Rome Tor Vergata, the Commonwealth Cyber Initiative, the Academia Sinica Investigator Award AS-IA-109-M01, the Agence Nationale de la Recherche under grant ANR MELODIA (ANR-20-CE40-0013), and the France 2030 program under grant agreement No. ANR-22-PETQ-0008 PQ-TLS. Date of this document: 2023-02-24.

1. Given supersingular elliptic curves E_0, E_1 along with descriptions of their endomorphism rings, compute an isogeny path $E_0 \to E_1$;
2. Given a supersingular elliptic curve E_0 along with a description of its endomorphism ring, and given an isogeny path $E_0 \to E_1$, compute a description of the endomorphism ring of E_1.

These algorithms—and variants—have successfully been used both constructively [GPS20, DKL+20, DdF+21] and for cryptanalysis [GPST16, Pet17, dQKL+21, EHL+18, DMPS19, FKMT22].

Without the additional information above, computing the endomorphism ring of an arbitrary supersingular curve remains a hard problem, both for classical and quantum computers. Given the importance of this problem, it is natural to ask whether it is possible to sample supersingular curves such that computing their endomorphism ring is a hard problem, crucially, even for the party who does the sampling. We shall call these objects *Supersingular Elliptic Curves of Unknown Endomorphism Ring*, or SECUER[1] in short.

Applications. Generating a SECUER has turned out to be a delicate task, and no such curve has ever been generated. Yet, several isogeny-based schemes can only be instantiated with a SECUER. This is the case, for example, of isogeny-based verifiable delay functions [DMPS19] and delay encryption [BD21]. The so-called CGL hash function based on supersingular curves [CLG09] has been shown to be broken by the knowledge of the endomorphism ring [EHL+18], and one possible fix is to instantiate it with a SECUER. Other protocols which require a SECUER include hash proof systems, dual mode PKE [ADMP20], oblivious transfer [LGd21], OPRF [Bas23], and commitment schemes [Ste22].

Contributions. We analyze and put into practice a protocol for distributed generation of SECUERs. Our main technical contribution is a key ingredient of the protocol: a new proof of isogeny knowledge (two curves E_0 and E_1 being public, a party wishes to prove that they know an isogeny $E_0 \to E_1$ without revealing it). Our proof is similar to the SIDH proof of knowledge [DFJP14, DDGZ22], but extends it in a way that makes it compatible with any base field, any walk length, and has provable statistical zero-knowledge (unlike any previous proof of isogeny knowledge). In particular, its statistical security makes it fully immune to the recent attacks [CD22, MMP+23, Rob22].

To prove statistical security, we analyze *supersingular ℓ-isogeny graphs with level structure*, a generalization of isogeny graphs that was recently considered in [DKL+20, Arp22]. We prove that these graphs, like classic isogeny graphs, possesses the Ramanujan property, a fact that is of independent interest. Using the property, we analyze the mixing behavior of random walks, which lets us give very precise parameters to instantiate the proof of knowledge at any given security level.

To show that the resulting protocol is practical, we implement it on top of Microsoft's SIDH library[2] and benchmark it for each of the standard SIKE primes [JAC+20]. We must stress that SIDH-style primes are possibly the most favorable to our protocol, in terms of practical efficiency.

Finally, we sketch a roadmap to run the distributed generation protocol for the SIKE primes in a real world setting with hundreds of participants.

Limitations. We must point out that our new proof of knowledge is not well adapted to a secure distributed generation protocol in the case where one wants to generate a SECUER defined over a prime field $\mathbb{F}_p$, instead of $\mathbb{F}_{p^2}$, such as in [ADMP20, LGd21]. Different proofs of knowledge [DG19, BKV19] could be plugged in the distributed protocol for the $\mathbb{F}_p$ case, however their practical usability is dubious.

1.1 Generating a SECUER

The cornerstone of isogeny-based cryptography is the endomorphism ring problem: if it could be solved efficiently, then all of supersingular isogeny-based cryptography would be broken [GPST16, EHL+18, Wes22a], leaving only ordinary isogeny-based cryptography [Cou06, Sto10, DKS18] standing.

Definition 1 (Endomorphism ring problem). *Given a supersingular curve $E/\mathbb{F}_{p^2}$, compute its endomorphism ring* $\mathrm{End}(E)$. *That is, compute an integral basis for a maximal order $\mathcal{O}$ of the quaternion algebra ramified at p and ∞, as well as an explicit isomorphism $\mathcal{O} \simeq \mathrm{End}(E)$.*

[1] The British spelling is SECURE.
[2] https://github.com/microsoft/PQCrypto-SIDH

For any p, there exists a polynomially sized subset of all supersingular curves for which the endomorphism ring can be computed in polynomial time [CPV20, LB20], but the problem is believed to be exponentially hard in general, even for quantum computers. A related problem, commonly encountered in isogeny protocols, is finding paths in supersingular isogeny graphs.

Definition 2 (Isogeny ℓ-walk problem). *Given two supersingular curves $E, E'/\mathbb{F}_{p^2}$ of the same order, and a small prime ℓ, find a walk from E to E' in the ℓ-isogeny graph.*

Such walks are always guaranteed to exist, as soon as they have length in $O(\log(p))$ [Mes86, Piz90, Koh96, CLG09].

The two problems are known to be polynomial time equivalent, assuming GRH [Wes22b]. Indeed, given $\mathrm{End}(E)$ and $\mathrm{End}(E')$, it is easy to compute a path $E \to E'$. Reciprocally, given $\mathrm{End}(E)$ and a path $E \to E'$, it is easy to compute $\mathrm{End}(E')$; and, by random self-reducibility, we can always assume that one of $\mathrm{End}(E)$ or $\mathrm{End}(E')$ is known.

Our goal is to generate a SECUER: a curve for which the endomorphism ring problem is hard, and consequently one for which it is hard to find a path to any other given curve.

What does not work. The supersingular elliptic curves over a finite field k of characteristic p are those such that $\#E(k) = 1 \bmod p$. Any supersingular curve is isomorphic to one defined over a field with p^2 elements, thus, without loss of generality, we are only interested in supersingular curves defined over $\mathbb{F}_{p^2}$. Among the p^2 isomorphism classes of elliptic curves over $\mathbb{F}_{p^2}$, only $\approx p/12$ of them correspond to supersingular curves.

The standard way to construct supersingular curves is to start from a curve with complex multiplication over a number field, and then reduce modulo p. Complex multiplication elliptic curves have supersingular reduction modulo 50% of the primes, thus this technique quickly produces supersingular curves for almost all primes. For example, the curve $y^2 = x^3 + x$, which has complex multiplication by the ring $\mathbb{Z}[i]$ of Gaussian integers, is supersingular modulo p if and only if $p = 3 \bmod 4$. Most isogeny-based protocols are instantiated using precisely this curve as starting point. These curves are not SECUERs, though, because from the information on complex multiplication one can compute the endomorphism ring in polynomial time [CPV20, LB20].

As p grows, the curves with computable[3] complex multiplication form only a negligible fraction of all supersingular curves in characteristic p, so we may still hope to get a SECUER if we can sample a supersingular curve at random from the whole set. The natural way to do so is to start from a well known supersingular curve, e.g. $E_0 : y^2 = x^3 + x$, take a random walk $E_0 \to E_1$ in the isogeny graph, and then select the arrival curve E_1. But, by virtue of the reductions mentioned above, any E_1 constructed this way cannot be called a SECUER either.

Several other techniques have been considered for generating SECUERs, however all attempts have failed so far [BBD+22, MMP22].

Distributed generation of SECUERs. An obvious solution that has been proposed for schemes that need a SECUER is to rely on a trusted party to start from a special curve E_0 and to perform an isogeny walk to a random curve E_1. Although E_1 is not a SECUER, if the trusted party keeps the walk $E_0 \to E_1$ secret, no one else will be able to compute $\mathrm{End}(E_1)$.

Of course, relying on a trusted third party is undesirable. The natural next step is to turn this idea into a distributed protocol with t parties generating a sequence of walks $E_0 \to E_1 \to E_2 \to \cdots \to E_t$. First, suppose that the sequence was generated honestly: the i-th party indeed generated a random isogeny from the previous curve E_{i-1} to a new curve E_i. Then it is sufficient for a *single* party to honestly discard their isogeny, for no path to be known by *anyone* from E_0 to E_t. Then, E_t is a SECUER for all practical purposes.

To make this protocol secure against active adversaries, an additional ingredient is needed. As it is, the last party could cheat as follows: instead of generating an isogeny $E_{t-1} \to E_t$, they could reboot the chain by generating an isogeny $E_0 \to E_t$, and submitting that instead. They could then compute the endomorphism ring of E_t. If only the curves E_i along the path are revealed, it is impossible to detect such

[3] Deuring showed that any supersingular curve can be lifted in several ways to a curve with complex multiplication, but for almost all curves computing such lifts has complexity exponential in $\log(p)$.

misbehavior. To prevent this, each party needs to prove that they know their component of the walk: an isogeny $E_{i-1} \to E_i$ (as first discussed in [BD21]). To this end, we develop a statistically zero-knowledge proof of isogeny knowledge.

1.2 Proof of isogeny knowledge

State-of-the-art. Protocols to prove knowledge of an isogeny have been mostly studied for signatures. The first such protocol is the SIDH-based proof of knowledge of [DFJP14]. Its security proof was found to be flawed and then fixed, either by changing the assumptions [GPV21] or by changing the protocol [DDGZ22]. However, these protocols are now fully broken by the recent polynomial time attacks on SIDH-like protocols [CD22, MMP+23, Rob22]. These attacks can be avoided by relying on ternary challenges [BKW20, DDGZ22].

CSIDH-based proofs of knowledge were first introduced in [DG19], and then improved in [BKV19] for the parameter set CSIDH-512. These are limited to isogeny walks between curves defined over a prime field $\mathbb{F}_p$, and tend to be prohibitively slow outside of the specially prepared parameter set CSIDH-512.

Finally, De Feo and Burdges propose an efficient proof of knowledge tailored to finite fields used in delay protocols [BD21]. However the soundness of this protocol is only conjectural, and, being based on pairing assumptions, is broken by quantum computers.

In summary, no general purpose, quantum-safe, zero-knowledge proof of knowledge of an isogeny walk between supersingular curves defined over $\mathbb{F}_{p^2}$ exists in previous literature.

Overview of our method. Our main technical contribution is a new proof of knowledge that ticks all the boxes above: it is compatible with any base field, any walk length, it has provable statistical zero-knowledge, and is practical—as illustrated by our implementation. The idea is the following. Two elliptic curves E_0 and E_1 being public, some party, the prover, wishes to convince the verifier that they know an isogeny $\phi : E_0 \to E_1$ (of degree, say, 2^m, large enough so it is guaranteed that such an isogeny exists). First, the prover secretly generates a random isogeny walk $\psi : E_0 \to E_2$ of degree, say, 3^n. Defining ϕ' with kernel $\psi(\ker(\phi))$, and ψ' with kernel $\phi(\ker(\psi))$, one obtains the following commutative diagram, known as "SIDH square" in the literature:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi} & E_1 \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi'} \\ E_2 & \xrightarrow[\phi']{} & E_3 \end{array} \tag{1}$$

Now, the prover publishes a hiding and binding commitment to E_2 and E_3. The verifier may now ask the prover to reveal one of the three isogenies ψ, ϕ', or ψ', by drawing a random $\mathsf{chall} \in \{-1, 0, 1\}$ (and open the commitment(s) corresponding to the relevant endpoints). For the prover to succeed with overwhelming probability, they must know all three answers, so they must know an isogeny from E_0 to E_1: the composition $\psi' \circ \phi' \circ \psi : E_0 \to E_1$. This is the idea behind the soundness of the protocol.

So far, this protocol is more or less folklore and superficially similar to [DDGZ22, §5.3]. But does it leak any information? Whereas previous protocols only achieved computational zero-knowledge, we provide a tweak that achieves statistical zero-knowledge: there is a simulator producing transcripts that are statistically indistinguishable from a valid run of the protocol. The simulator starts by choosing the challenge chall *first*, then it generates an isogeny that is statistically indistinguishable from either ψ, ϕ', or ψ', according to the value of chall. Simulating ψ (or ψ') is straightforward: generate a random isogeny walk $\tilde{\psi}$ (or $\tilde{\psi}'$) of degree 3^n from E_0 (or from E_1). The isogeny $\tilde{\psi}$ is a *perfect* simulation of ψ. Simulating ϕ' seems trickier. An obvious approach is to first generate a random E_2 (for instance, by simulating $\psi : E_0 \to E_2$), then generate a random walk isogeny $\tilde{\phi}' : E_2 \to E_3$ of degree 2^m. While this may seem too naive, we in fact prove that when $\deg(\psi)$ is large enough, the distribution of $\tilde{\phi}'$ is statistically close to a honestly generated ϕ'. The key is a proof that the isogeny graph enriched with so-called *level structure* has rapid mixing properties.

Isogeny graphs with level structure. The isogeny ϕ' is essentially characterized by its source, E_2, and its kernel $\ker(\phi')$, a (cyclic) subgroup of order $\deg(\phi')$. We are thus interested in random variables of the form (E, C), where E is an elliptic curve, and C a cyclic subgroup of E, of order some integer d (not divisible by p). We call such a pair (E, C) a *level d Borel structure*.

The simulator proposed above essentially generates $\tilde{\phi}'$ as a uniformly random level 2^m Borel structure $(E, C) = (E_2, \ker(\tilde{\phi}'))$. On the other hand, a honestly generated ϕ' corresponds to a pair $(\psi(E_0), \psi(\ker\phi))$, and ψ is a uniformly random isogeny walk of degree 3^n. This process corresponds to a random walk of length n in the 3-*isogeny graph with level 2^m structure*, with starting point $(E_0, \ker\phi)$. We prove the following result.

Theorem 3. *Let $G = G(p, d, \ell)$ the supersingular ℓ-isogeny graph with level d Borel structure. The adjacency matrix A of G is diagonalizable, with real eigenvalues, and has the Ramanujan property, i.e. the integer $\ell + 1$ is an eigenvalue of A of multiplicity one, while all the other eigenvalues are contained in the Hasse interval $[-2\sqrt{\ell}, 2\sqrt{\ell}]$.*

As a consequence, we prove that random walks quickly converge to the stationary distribution, so $\tilde{\phi}'$ and ϕ' are statistically indistinguishable.

Paper outline. We start in Section 2 with a few technical preliminaries on elliptic curves, isogenies, and proofs of knowledge. Section 3 is dedicated to the proof of Theorem 3. This section can be read independently from the rest. The reader only interested in applications, and willing to accept Theorem 3 (and its consequence on non-backtracking random walks, Theorem 11, page 11), can safely skip to the following section. This theoretical tool at hand, we then describe and analyse the new proof of isogeny knowledge in Section 4. We describe the protocol to generate a SECUER in Section 5, and prove its security. Finally, we report on our implementation in Section 6.

2 Preliminaries

2.1 General Notations

We write $x \leftarrow \chi$ to represent that an element x is sampled at random from a set/distribution $\mathcal{X}$. The output x of a deterministic algorithm $\mathcal{A}$ is denoted by $x = \mathcal{A}$ and the output x' of a randomized algorithm $\mathcal{A}'$ is denoted by $x' \leftarrow \mathcal{A}'$. For $a, b \in \mathbb{N}$ such that $a, b \geq 1$, we denote by $[a, b]$ (resp. $[a]$) the set of integers lying between a and b, both inclusive (the set of integers lying between 1 and a, both inclusive). We refer to $\lambda \in \mathbb{N}$ as the security parameter, and denote by $\mathsf{poly}(\lambda)$, $\mathsf{polylog}(\lambda)$ and $\mathsf{negl}(\lambda)$ any generic (unspecified) polynomial, poly-logarithmic or negligible function in λ, respectively.[4] For probability distributions $\mathcal{X}$ and $\mathcal{Y}$, we write $\mathcal{X} \approx \mathcal{Y}$ if the statistical distance between $\mathcal{X}$ and $\mathcal{Y}$ is negligible.

2.2 Elliptic curves, isogenies and "SIDH squares"

We assume the reader has some familiarity with elliptic curves and isogenies. Throughout the text, p shall be a prime number, $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ the finite fields with p and p^2 elements respectively. Unless specified otherwise, all elliptic curves will be supersingular and defined over $\mathbb{F}_{p^2}$. We write $E[d]$ for the subgroup of d-torsion points of E over the algebraic closure.

Unless specified otherwise, all isogenies shall be separable. If G is a finite subgroup of E, we write $\phi : E \to E/G$ for the unique (up to post-composition with an isomorphism of E/G) separable isogeny with kernel G. If G is cyclic, we say the isogeny is cyclic. We denote by $\hat{\phi}$ the dual isogeny to ϕ. Separable isogenies and their duals can be computed and/or evaluated in time $\mathsf{poly}(\#G)$ using any of the algorithms in [Vél71, BDFLS20], however in some cases, e.g. when $\#G$ only contains small factors, this cost may be lowered to as little as $\mathsf{polylog}(\#G)$.

Given separable isogenies $\phi : E_0 \to E_1$ and $\psi : E_0 \to E_2$ of coprime degrees, we obtain the commutative diagram in (1) by defining $\phi' : E_2 \to E_2/\psi(\ker(\phi))$ and $\psi' : E_1 \to E_1/\phi(\ker(\psi))$. Again, E_3 is only defined up to isomorphism. In categorical parlance, this is the *pushout* of ϕ and ψ, but cryptographers may know it better through its use in the SIDH key exchange. We refer to these commutative diagrams as *SIDH squares* or *SIDH ladders* (see Section 4.2 for more details).

[4] A function $f : \mathbb{N} \to \mathbb{N}$ is said to be negligible in λ if for every positive polynomial p, $f(\lambda) < 1/p(\lambda)$ when λ is sufficiently large.

2.3 Proofs of Knowledge

Our main technical contribution is a Σ-protocol to prove knowledge of an isogeny of given degree between two supersingular elliptic curves. Recall a Σ-protocol for an NP-language $\mathcal{L}$ is a public-coin three-move interactive proof system consisting of two parties: a verifier and a prover. The prover is given a witness w for an element $x \in \mathcal{L}$, his goal is to convince the verifier that he knows w.

Definition 4 (Σ-protocol). *A Σ-protocol Π_Σ for a family of relations $\{\mathcal{R}\}_\lambda$ parameterized by security parameter λ consists of PPT algorithms $(\mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ where V is deterministic and we assume $\mathsf{P}_1, \mathsf{P}_2$ share states. The protocol proceeds as follows:*

1. *The prover, on input $(x, w) \in \mathcal{R}$, returns a commitment $\mathsf{com} \leftarrow \mathsf{P}_1(x, w)$ which is sent to the verifier.*
2. *The verifier flips λ coins and sends the result to the prover.*
3. *Call chall the message received from the verifier, the prover runs $\mathsf{resp} \leftarrow \mathsf{P}_2(\mathsf{chall})$ and returns resp to the verifier.*
4. *The verifier runs $V(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ and outputs a bit.*

A transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$ is said to be valid, or accepting, if $V(x, \mathsf{com}, \mathsf{chall}, \mathsf{resp})$ outputs 1. The main requirements of a Σ-protocol are:

Correctness: If the prover knows $(x, w) \in \mathcal{R}$ and behaves honestly, then the verifier outputs 1.

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

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

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

A special sound Σ-protocol for $\mathcal{R}$ is also called a *Proof of Knowledge (PoK)* for $\mathcal{R}$. Our Σ-protocol will have the peculiar property that the relation used to prove correctness turns out to be a subset of the one used to prove soundness. This will require extra care when proving security in Section 5.

Special honest verifier zero-knowledge (SHVZK): There exists a polynomial-time simulator that, given a statement x and a challenge chall, outputs a valid transcript $(\mathsf{com}, \mathsf{chall}, \mathsf{resp})$ that is indistinguishable from a real transcript.

Definition 5. *A Σ-protocol $(\mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ is computationally special honest verifier zero-knowledge if there exists a probabilistic polynomial time simulator Sim such that for all probabilistic polynomial time stateful adversaries $\mathcal{A}$*

$$\Pr\left[\mathcal{A}(\mathsf{com}, \mathsf{chall}, \mathsf{resp}) = 1 \;\middle|\; \begin{array}{l}(x, w, \mathsf{chall}) \leftarrow \mathcal{A}(1^\lambda);\\ \mathsf{com} \leftarrow \mathsf{P}_1(x, w);\\ \mathsf{resp} = \mathsf{P}_2(\mathsf{chall})\end{array}\right]$$
$$\approx \Pr\left[\mathcal{A}(\mathsf{com}, \mathsf{chall}, \mathsf{resp}) = 1 \;\middle|\; \begin{array}{l}(x, w, \mathsf{chall}) \leftarrow \mathcal{A}(1^\lambda);\\ (\mathsf{com}, \mathsf{resp}) \leftarrow \mathsf{Sim}(x, \mathsf{chall})\end{array}\right].$$

If the above indistinguishability holds statistically against all unbounded adversaries $\mathcal{A}$, the protocol is said to be statistically SHVZK.

2.4 Non-Interactive Zero-Knowledge Proofs

In this paper, we consider non-interactive zero-knowledge (NIZK) proofs in the random oracle model that satisfy correctness, computational extractability and statistical zero-knowledge.

Definition 6. ***(NIZK proofs.)*** *Let $\mathcal{R}$ be a relation and let the language $\mathcal{L}$ be a set of statements $\{\mathsf{st} \in \{0,1\}^n\}$ such that for each statement $\mathsf{st} \in \mathcal{L}$, there exists a corresponding witness wit such that $(\mathsf{st}, \mathsf{wit}) \in \mathcal{R}$. A non-interactive zero-knowledge (NIZK) proof system for $\mathcal{R}$ is a tuple of probabilistic polynomial-time (PPT) algorithms $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ defined as follows (we assume that all algorithms in the description below have access to a common random oracle; we omit specifying it explicitly for ease of exposition):*

– $\mathsf{P}_{\mathsf{NIZK}}(\mathsf{st}, \mathsf{wit})$*: A PPT algorithm that, given a statement* $\mathsf{st} \in \{0,1\}^n$ *and a witness* wit *such that* $(\mathsf{st}, \mathsf{wit}) \in \mathcal{R}$*, outputs a proof* Π.
– $\mathsf{V}_{\mathsf{NIZK}}(\mathsf{st}, \Pi)$*: A deterministic algorithm that, given a statement* $\mathsf{st} \in \{0,1\}^n$ *and a proof* Π*, either outputs* 1 *(accept) or* 0 *(reject).*

The following correctness and security properties should be satisfied:

Correctness. For any $(\mathsf{st}, \mathsf{wit}) \in \mathcal{R}$, letting $\Pi = \mathsf{P}_{\mathsf{NIZK}}(\mathsf{st}, \mathsf{wit})$, we must have $\mathsf{V}_{\mathsf{NIZK}}(\mathsf{st}, \Pi) = 1$.

Computational extractability. There exists an efficient PPT extractor $\mathsf{Ext}_{\mathsf{NIZK}}$ such that for any security parameter $\lambda \in \mathbb{N}$ and for any *polynomially bounded* cheating prover P^* where: (i) $\mathsf{Ext}_{\mathsf{NIZK}}$ has rewinding access to P^*, and (ii) $\mathsf{P}_{\mathsf{NIZK}}$, $\mathsf{Ext}_{\mathsf{NIZK}}$ and P^* all have access to a common random oracle, letting $(\mathsf{st}, \Pi) \leftarrow P^*(1^\lambda)$ and $\mathsf{wit} = \mathsf{Ext}_{\mathsf{NIZK}}(\mathsf{st}, \Pi)$, if $\mathsf{V}_{\mathsf{NIZK}}(\mathsf{st}, \Pi) = 1$, we must have $\Pr[(\mathsf{st}, \mathsf{wit}) \in \mathcal{R}] > 1 - \mathsf{negl}(\lambda)$.

Statistical zero-knowledge. There exists an efficient PPT simulator $\mathsf{Sim}_{\mathsf{NIZK}}$ such that for any security parameter $\lambda \in \mathbb{N}$ and for any non-uniform *unbounded* "cheating" verifier $V^* = (V_1^*, V_2^*)$ where $\mathsf{P}_{\mathsf{NIZK}}$, V_1^* and V_2^* all have access to a common random oracle, and such that $\mathsf{Sim}_{\mathsf{NIZK}}$ is allowed programming access to the same random oracle, we have

$$\left| \Pr\left[V_2^*(\mathsf{st}, \Pi, \xi) = 1 \wedge (\mathsf{st} \in \mathcal{L})\right] - \Pr\left[V_2^*(\mathsf{st}, \widehat{\Pi}, \xi) = 1 \wedge (\mathsf{st} \in \mathcal{L})\right] \right| \leq \mathsf{negl}(\lambda),$$

where $(\mathsf{st}, \mathsf{wit}, \xi) \leftarrow V_1^*(1^\lambda)$, $\Pi \leftarrow \mathsf{P}_{\mathsf{NIZK}}(\mathsf{st}, \mathsf{wit})$, and $\widehat{\Pi} \leftarrow \mathsf{Sim}_{\mathsf{NIZK}}(\mathsf{st})$.

3 Isogeny graphs and expansion

Let p be a prime and d an integer not divisible by p. An elliptic curve with level d Borel structure is a pair (E, C), where E is an elliptic curve defined over a field of characteristic p and C is an order d cyclic subgroup of $E[d]$. We say that two such pairs (E_1, C_1) and (E_2, C_2) are isomorphic if there exists an isomorphism $\phi : E_1 \to E_2$ such that $\phi(C_1) = C_2$. An automorphism of (E, C) is an isomorphism $(E, C) \to (E, C)$. They form the group $\mathrm{Aut}(E, C)$.

Let ℓ be a prime not dividing pd. The supersingular ℓ-isogeny graph with level d structure $G = G(p, d, \ell)$ is defined as follows. The set of vertices of G is a complete set $V = V(p, d) = \{(E_i, C_i)\}$ of representatives of the set of isomorphism classes of supersingular elliptic curves with a level d Borel structure defined over $\mathbb{F}_{p^2}$. We note that each such class over $\overline{\mathbb{F}_{p^2}}$ admits a model defined over $\mathbb{F}_{p^2}$: Each isomorphism class of supersingular elliptic curves has a representative E such that $\#E(\mathbb{F}_{p^2}) = (p+1)^2$ and thus the p^2-Frobenius acts as a scalar multiplication $[-p]$, so the kernel of any ℓ-isogeny is $\mathrm{Gal}(\overline{\mathbb{F}_{p^2}})$-invariant.

Now, the set of edges from (E, C) to (E', C') in G is the set of degree ℓ isogenies from E to E' which map C to C', modulo the action of $\mathrm{Aut}(E', C')$ (by postcomposition). The number of edges is independent of the representative of the isomorphism classes. When $d = 1$, we recover the usual definition of the supersingular ℓ-isogeny graph.

This graph is directed. The out-degree of each vertex is $\ell + 1$, however the in-degree is not always $\ell + 1$, hence the adjacency matrix of the graph is not always symmetric.

3.1 Generalities on the graph and its adjacency matrix

Let $V = \{(E_i, C_i)\}$ for $i = 1, \ldots, n$ be the vertex set of $G = G(p, d, \ell)$. On the complex vector space $\mathbb{C}^V$, we introduce the Hermitian form $Q((E_i, C_i), (E_j, C_j)) = w_i \delta_{ij}$, where δ_{ij} is the Kronecker symbol and $w_i := \frac{1}{2}|\mathrm{Aut}(E_i, C_i)|$. Denote by $\|\cdot\|_Q$ the associated norm. We will compare $\|\cdot\|_Q$ with the L^1 and L^2 norms on $\mathbb{C}^V$. The set Ω of probability distributions on V is the set of vectors with real positive entries and L^1 norm equal to 1. Consider also the vector $\mathcal{E} = \sum_{i=1}^n \frac{1}{w_i}(E_i, C_i)$, and s the probability distribution obtained normalizing $\mathcal{E}$. The following result contains a number of general facts about the adjacency matrix of G, which will be used later on.

Theorem 7. *1. The adjacency matrix A of G is self-adjoint with respect to Q; in particular it is diagonalizable with real eigenvalues and eigenvectors;*

2. *The vector $\mathcal{E}$ is a left-eigenvector of eigenvalue $\ell+1$ of A;*
3. *The vector u with all entries equal to 1 is a right-eigenvector of A; in particular its orthogonal complement S with respect to the L^2 scalar product is preserved by right multiplication by A;*
4. $K \coloneqq \inf\{\,\|v\|_Q \,:\, v\in\mathbb{C}^V \text{ and } \|v\|_{L^1}=1\} = \left(\frac{(p-1)d}{12}\prod_q(1+\frac{1}{q})\right)^{-1/2}$, *where the product index q runs over the prime divisors of d;*
5. $M \coloneqq \sup\{\,\|\pi-s\|_Q \,:\, \pi\in\Omega\} \le \sqrt{3}$.

Proof. First we show 1. Let L_{ij} be the set of degree ℓ isogenies from (E_i,C_i) to (E_j,C_j). If f is in L_{ij}, then the dual isogeny $\hat{f}$ is a degree ℓ isogeny from (E_j,C_j) to $(E_i,\ell C_i)$. Since ℓ is coprime with d, ℓC_i is equal to C_i, and the duality gives a bijection between L_{ij} and L_{ji}. The entry a_{ij} of A is the cardinality of the quotient $L_{ij}/\operatorname{Aut}(E_j,C_j)$, hence $|\operatorname{Aut}(E_i,C_i)|a_{ji} = |\operatorname{Aut}(E_j,C_j)|a_{ij}$. Dividing this equality by two we get $w_i a_{ji} = w_j a_{ij}$. The claim now follows from the definition of Q.

We now prove 2. We have

$$\begin{aligned}\mathcal{E}A &= \sum_{i=1}^n \frac{1}{w_i}(E_i,C_i)A = \sum_{i,j=1}^n \frac{1}{w_i}\frac{|L_{ij}|}{w_i}(E_j,C_j) = \sum_{j=1}^n \frac{1}{w_j}(E_j,C_j)\sum_{i=1}^n \frac{|L_{ji}|}{w_i}\\ &= \sum_{j=1}^n \frac{1}{w_j}(E_j,C_j)(\ell+1) = (\ell+1)\mathcal{E}\,.\end{aligned}$$

To see part 3, observe that the out-degree of each vertex of G is $\ell+1$, hence the sum of the elements of the rows of A is $\ell+1$, so the claim.

We now prove 4. Let $\langle\cdot,\cdot\rangle$ be the Hermitian product on $\mathbb{C}^V$ such that the basis (E_i,C_i) is unitary. Let $w=\sum w_i^{-1/2}(E_i,C_i)$ and, for each $v=\sum v_i(E_i,C_i)$, let $\tilde{v}=\sum w_i^{1/2}|v_i|(E_i,C_i)$. Then, the Cauchy-Schwarz inequality gives

$$\|v\|_{L^1} = \langle\tilde{v},w\rangle \le \sqrt{\langle\tilde{v},\tilde{v}\rangle}\sqrt{\langle w,w\rangle} = \|v\|_Q\sqrt{\langle w,w\rangle} = \|v\|_Q\sqrt{\sum\frac{1}{w_i}}$$

and moreover we get the equality when $\tilde{v} = w/\|w\|_{L^1}$. We now compute $K^{-1} = \sqrt{\sum\frac{1}{w_i}}$. Eichler's formula [Hus04, Section 13.5, Theorem 4.1] gives

$$\sum_{\substack{E/\overline{\mathbb{F}_p}\text{ supersingular,}\\ \text{up to }\overline{F_p}\text{-isomorphism}}} \frac{1}{|\operatorname{Aut}(E)|} = \frac{p-1}{24}\,.$$

We are going to show that, for H the group of upper triangular matrices

$$\sum_{i \text{ such that } E_i\simeq E} \frac{|\operatorname{Aut}(E)|}{|\operatorname{Aut}(E_i,C_i)|} = [\mathrm{GL}_2(\mathbb{Z}/d\mathbb{Z}) : H]\,. \tag{2}$$

Given this equation for granted, K can be computed by writing $d=\prod_q q^{e_q}$ and checking that $|\mathrm{GL}_2(\mathbb{Z}/d\mathbb{Z})| = \prod_q(q^{2e_q}-q^{2e_q-2})(q^{2e_q}-q^{2e_q-1})$ and $|H| = \prod_q q^{e_q}(q^{e_q}-q^{e_q-1})^2$.

Equation (2) is the equation of the orbits for a group action. Fix an elliptic curve E, let X be the set of order d cyclic subgroups of $E[d]$. This set has a natural transitive action by $\operatorname{Aut}(E[d])\cong\mathrm{GL}_2(\mathbb{Z}/d\mathbb{Z})$, which gives a bijection $X\leftrightarrow\mathrm{GL}_2(\mathbb{Z}/d\mathbb{Z})/H$, so the right hand side of Equation (2) is the cardinality of X. Level d Borel structures on E are the orbits of the action of $\operatorname{Aut}(E)$ on X. The left hand side of Equation (2) is again the cardinality of X, obtained summing the cardinalities of each orbit.

Finally we prove 5. Let $\pi=\sum_{i=1}^n\pi_i(E_i,C_i)$ be a probability distribution and let $\lambda=\sum_{i=1}^n\frac{1}{w_i}$, so that $s=\sum_{i=1}^n\frac{1}{\lambda w_i}(E_i,C_i)$. Then, using $\sum\pi_i=1$, we have

$$\begin{aligned}\|\pi-s\|_Q^2 &= \sum_{i=1}^n w_i\left(\pi_i-\frac{1}{\lambda w_i}\right)^2 = \sum_{i=1}^n\left(w_i\pi_i^2-\frac{2\pi_i}{\lambda}+\frac{1}{\lambda^2 w_i}\right)\\ &= \sum_{i=1}^n w_i\pi_i^2-\frac{2}{\lambda}+\frac{1}{\lambda} \le \sum_{i=1}^n w_i\pi_i^2 \le (\max w_i)\sum_{i=1}^n\pi_i = \max w_i\,.\end{aligned}$$

We conclude recalling that $w_i\le 3$ for every i. Notice that for $\pi=(E_i,C_i)$ we get $\|\pi-s\|_Q^2 = w_i - 1/\lambda$, hence the above estimate is not too loose. ∎

3.2 Proof of Theorem 3

We now prove that $G = G(p, d, \ell)$ has the Ramanujan property. This follows from the first three items of Theorem 7 combined with the following result, whose proof heavily relies on the theory of modular forms. An immediate consequence is that G is connected and not bipartite, a different proof of which can be found in [GK21, Theorem 5.3.3].

Theorem 8. *Let $S \subset \mathbb{C}^V$ be the subspace of vectors $\sum_i v_i(E_i, C_i)$ such that $\sum_i v_i = 0$, as in Theorem 7. The eigenvalues of the action of A on S are all contained in the Hasse interval $[-2\sqrt{\ell}, 2\sqrt{\ell}]$.*

To prove Theorem 8, we assume standard notations and results about quadratic forms and modular forms, such as the ones from [DS05, Sch74, HPS89]. Given two elliptic curves with level structure (E_i, C_i) and (E_j, C_j), we denote by Λ_{ij} the lattice of isogenies $\phi\colon E_i \to E_j$ such that $\phi(C_i) \subset C_j$. The degree defines a quadratic form deg on Λ_{ij}. This quadratic module has rank four, level dp and determinant d^2p^2. We can thus define the theta series

$$\Theta_{ij}(\tau) = \frac{1}{|\mathrm{Aut}(E_j, C_j)|} \sum_{\phi \in \Lambda_{ij}} q^{\deg(\phi)}, \quad \text{with } q = e^{2\pi i \tau}.$$

This function is in $M_2(\Gamma_0(dp))$, the space of modular forms of weight two for the modular group $\Gamma_0(dp)$, by [HPS89, Theorem 4.2] (observe that in *loc. cit.* the exponential is one because $Q(h)$ is an integer; moreover, we choose $P = 1$) or [Sch74, Chapter IX, Theorem 5, page 218]. The above construction extends to an Hermitian pairing

$$\Theta\colon \mathbb{C}^V \otimes \mathbb{C}^V \to M_2(\Gamma_0(dp)) : ((\alpha_i)_i \otimes (\beta_j)_j) \longmapsto \sum_{i,j} \alpha_i \beta_j \Theta_{ij}.$$

We call this pairing the Brandt pairing, even though there is a little ambiguity[5] in this set-up. The Brandt pairing is non-degenerate: let $v = \sum c_i(E_i, C_i)$, then the coefficient of q of $\Theta(v, v)$ is the Hermitian norm of the vector of coefficients $(\dots, c_i, \dots)$. We will prove the following two key propositions.

Proposition 9. *The Brandt pairing intertwines the adjacency matrix A of G and the Hecke operator T_ℓ; in symbols $T_\ell \Theta(w, v) = \Theta(wA, v)$ for all $w, v \in \mathbb{C}^V$.*

Proposition 10. *For every three elliptic curves with level structure (E_1, C_1), (E_2, C_2) and (E_3, C_3), we have a cusp form*

$$\Theta((E_1, C_1), (E_3, C_3)) - \Theta((E_2, C_2), (E_3, C_3)).$$

The combination of these two results tells that the spectrum of the action of A restricted to S is contained into the spectrum of the action of the Hecke operator T_ℓ on the space of cusp modular forms of weight two for $\Gamma_0(dp)$. The Ramanujan Conjecture, proved by Eichler, predicts that this second spectrum is contained in the Hasse interval, and hence proves Theorem 8.

We refer to [Del74, Theorem 8.2] for a proof of the Ramanujan Conjecture. In *loc. cit.* this result is proven only for eigenvectors of T_ℓ which are new-forms. An eigenvector which is an old form will come from an embedding $\iota\colon S_2(\Gamma_0(m)) \to S_2(\Gamma_0(dp))$ with m that divides dp. Since ℓ is coprime with dp, the map ι is T_ℓ-equivariant (cf. [DS05, proof of Proposition 5.6.2]), so we can still deduce our result from [Del74, Theorem 8.2]. It is worth recalling that [Del74, Theorem 8.2] is stronger than what we need, as it applies to modular forms of every weight.

Proof of Proposition 9. We prove that both sides have the same q-expansions. For a power series $F \in \mathbb{C}[[q]]$, denote $a_n(F)$ the coefficient of q^n. By definition

$$a_n(\Theta((E_i, C_i), (E_j, C_j))) = |\mathrm{Aut}(E_j, C_j)|^{-1} \cdot |\mathrm{Hom}^n((E_i, C_i), (E_j, C_j))|,$$

[5] Rather than using the condition $\phi(C_i) \subset C_j$, we could have defined Λ_{ij} using $\phi(C_i) = C_j$. The second definition does not give a lattice but still permits to define a pairing. This second pairing generalizes to all level structures, so it might deserve better the name of Brandt pairing. However, the second pairing gives a more complicated proof in the Borel case, so we have opted for the first one.

where $\mathrm{Hom}^n((E_i,C_i),(E_j,C_j))$ is the set of degree n isogenies in Λ_{ij}. For $f \in M_2(\Gamma_0(dp))$, we have $a_n(T_\ell f) = a_{n\ell}(f) + \ell a_{n/\ell}(f)$ (see e.g. [DS05, Proposition 5.2.2]), where $a_{n/\ell}(f)$ is set to zero in the case $n/\ell \notin \mathbb{Z}$. In particular,

$$\begin{aligned} &a_n(T_\ell\, \Theta((E_i,C_i),(E_j,C_j))) = \\ &= |\mathrm{Aut}(E_j,C_j)|^{-1}\Big(|\mathrm{Hom}^{n\ell}((E_i,C_i),(E_j,C_j))| + \ell|\mathrm{Hom}^{n/\ell}((E_i,C_i),(E_j,C_j)|\Big) \end{aligned} \tag{3}$$

On the other side,

$$\begin{aligned} a_n(\Theta((E_i,C_i)A,(E_j,C_j))) &= \sum_C a_n(\Theta((E_i/C,\pi_C(C_i)),(E_j,C_j))) = \\ &= |\mathrm{Aut}(E_j,C_j)|^{-1}\sum_C |\mathrm{Hom}^n((E_i/C,\pi_C(C_i)),(E_j,C_j))| \end{aligned} \tag{4}$$

where C varies among the cyclic non-trivial subgroups of $E_i[\ell]$ of cardinality ℓ, and π_C is the projection $E_i \to E_i/C$. For each C let

$$\begin{aligned} F_C\colon \mathrm{Hom}^n((E_i/C,\pi_C(C_i)),(E_j,C_j)) &\longrightarrow \mathrm{Hom}^{n\ell}((E_i,C_i),(E_j,C_j)) \\ f &\longmapsto f\circ\pi_C\,, \end{aligned}$$

and let F be the disjoint union of the above maps. The map F is surjective: if $\alpha\colon (E_i,C_i)\to(E_j,C_j)$ has degree $n\ell$, then $\ker(\alpha)\cap E_i[\ell] \neq \{0\}$, hence there exists a cyclic non-trivial $C\subset \ker(\alpha)\cap E_i[\ell]$, and we can write $\alpha = f\circ\pi_C$. In particular, let us compute the cardinality of the fiber $F^{-1}(\alpha)$ for α in the codomain. Each F_C is injective, hence $|F^{-1}(\alpha)|$ is equal to the number of subgroups C such that $F_C^{-1}(\alpha)$ is not empty, that is the number of subgroups C contained in $\ker(\alpha)\cap E_i[\ell]$. Hence

$$|F^{-1}(\alpha)| = \begin{cases} \ell+1 & \text{if } \alpha = \ell\beta \text{ for some } \beta\in\mathrm{Hom}^{n/\ell}((E_i,C_i),(E_j,C_j)),\\ 1 & \text{otherwise}\end{cases}$$

By (4), the domain of F has size exactly $|\mathrm{Aut}(E_j,C_j)|\cdot a_n(\Theta(A(E_i,C_i),(E_j,C_j)))$, hence the proposition follows from (3) together with the above formula summed over α in the codomain. □

Proof of Proposition 10 We have to show that, for any two pairs (E,C) and (E',C') and any cusp of $X_0(dp)$, the residue r of $\Theta((E,C),(E',C'))d\tau$ does not depend on (E,C) and (E',C') at the cusp but only on p, d and the cusp.

By the discussion in [DS05, Section 3.8, page 103] each cusp can be represented as $\left(\begin{smallmatrix}a\\c\end{smallmatrix}\right)$ with c dividing dp, and r is equal to $a_0(\Theta((E,C),(E',C'))|_M)$ for M any matrix in $\mathrm{SL}_2(\mathbb{Z})$ of the form $\left(\begin{smallmatrix}a&b\\c&\delta\end{smallmatrix}\right)$.

By [Sch74, Chapter IX, Equation (21), page 213], we have

$$r = \frac{1}{c^2pd}\sum_{\nu,\lambda\in\Lambda/c\Lambda} e\left(\frac{(a-1)\deg(\lambda)+\deg(\lambda+\nu)+(\delta-1)\deg(\nu)}{c}\right)$$

where $e(z) = e^{2\pi i z}$, and Λ is the lattice of isogenies from (E,C) to (E',C') which map C into C'. The above formula tells us that r only depends on M and on the quadratic form $\deg\colon \Lambda/c\Lambda\to\mathbb{Z}/c\mathbb{Z}$. Writing $c = c_0p^\epsilon$ with c_0 dividing N and $\epsilon = 0,1$ and using the Chinese remainder theorem we can split the quadratic form in two parts

$$\Lambda/c\Lambda = \Lambda/c_0\Lambda\times\Lambda/p^\epsilon\Lambda \xrightarrow{\deg\times\deg} \mathbb{Z}/c_0\mathbb{Z}\times\mathbb{Z}/p^\epsilon\mathbb{Z}\cong\mathbb{Z}/c\mathbb{Z}\,.$$

The quadratic module $(\Lambda/c_0\Lambda,\deg)$ is (non-canonically) isomorphic to a Borel subalgebra of the algebra $(\mathrm{End}((\mathbb{Z}/c_0\mathbb{Z})^{\oplus 2}),\det)$. An isomorphism can be obtained mapping it to $\mathrm{Hom}(E[c_0],E'[c_0])$, and then choosing a symplectic basis.

If $\epsilon = 0$ we are done, otherwise $\epsilon = 1$. Since $[\mathrm{Hom}(E,E'):\Lambda] = d$ is prime to p, we have $\Lambda/p = \mathrm{Hom}(E,E')/p = (\mathrm{Hom}(E,E')\otimes\mathbb{Z}_p)/p$, and the quadratic $\mathbb{Z}_p$-module $\mathrm{Hom}(E,E')\otimes\mathbb{Z}_p$ does not depend on the pair because, by the Deuring correspondence (see [Voi21, Theorem 42.3.2.]) and by [Voi21, Lemma 19.6.6], it is isomorphic to $\lambda\mathcal{O}_p$ with the reduced norm, where $\mathcal{O}_p$ is the maximal order in the non-ramified quaternions over $\mathbb{Q}_p$, and λ is an element of norm prime to p. □

3.3 Mixing time of non-backtracking walks

We finally analyze the behavior of random walks in $G = G(p, d, \ell)$, which we will ultimately use to prove statistical indistinguishability of distribution arising from our proof of knowledge. First, observe that Theorem 7 item 2 shows that the probability distribution s introduced in Subsection 3.1 is the stationary distribution on G. This is nearly the uniform distribution: all curves are equally likely, with the possible exception of the two curves with extra automorphisms, $j = 1728$ and $j = 0$, which are respectively twice and thrice less likely.

We are going to determine the speed at which random walks converge to the stationary distribution. We focus on non-backtracking walks, which are the most useful for cryptographic protocols, but, because the graph is directed, we need some care to define them. Edges of G are equivalence classes of isogenies, so we choose a representative for each class. For an edge α we define its dual edge as the chosen representative β for the class $\mathrm{Aut}(E, C)\hat{\alpha}$, so that $\beta\alpha = u\ell$ for $u \in \mathrm{Aut}(E, C)$. Notice that the dual of β (as an edge) might be different from α, but this is not relevant for us. We say that a random walk on G is non-backtracking walk if an edge is never followed by its dual.

With this "duality", we have that isogenies of degree a power of ℓ and with cyclic kernel (up to the equivalence $\alpha \sim \beta$ iff $\ker \alpha = \ker \beta$) correspond to non-backtracking walks.

Theorem 11 (Mixing time). *Let π be a probability distribution on G, and $\pi^{(k)}$ the distribution obtained after a non-backtracking random walk of length k. Then we have*

$$d_{TV}(\pi^{(k)}, s) \leq \frac{1}{2} K^{-1} M \frac{(\ell+1)(k+1) - 2}{(\ell+1)\sqrt{\ell^k}},$$

where K and M are as in Theorem 7 and d_{TV} denotes the total variation distance.

Proof. Denote by $A^{(k)}$ the matrix whose (i, j) entry is the number of non-backtracking walks from i to j. Since each edge has a unique dual, we get the same recurrence formula as in the non-oriented case, namely

$$A^{(1)} = A, \quad A^{(2)} = A^2 - (\ell+1), \quad A^{(k+1)} = AA^{(k)} - \ell A^{(k-1)}.$$

Observe that the sum of all the entries in a fixed row of $A^{(k)}$ is $(\ell+1)\ell^{k-1}$. We denote by $P^{(k)}$ its normalization

$$P^{(k)} := \frac{1}{(\ell+1)\ell^{k-1}} A^{(k)}.$$

Hence, $P^{(k)}$ is a polynomial in A, see e.g. [ABLS07, Section 2]. Let us call this polynomial $\mu_k(x)$ (here, the use of the symbol μ_i is slightly different from the one of [ABLS07]). The matrix $P^{(k)}$ is diagonalizable, it has the same eigenvectors as A, and has eigenvalues $\mu_k(\ell+1) = 1$ and $\mu_k(\lambda_i)$, where λ_i is any eigenvalue of A different from $\ell + 1$.

Combining the proof of [ABLS07, Lemma 2.3] and Theorem 3, we get

$$\mu_k(\lambda_i) = \frac{1}{\sqrt{(\ell+1)\ell^{k-1}}} \left(\sqrt{\frac{\ell}{\ell+1}} \frac{\sin((k+1)\theta)}{\sin(\theta)} - \frac{1}{\sqrt{(\ell+1)\ell}} \frac{\sin((k-1)\theta)}{\sin(\theta)} \right) \tag{5}$$

where $\cos(\theta) = \lambda_i/(2\sqrt{\ell})$. Recall that $|\sin(x+y)| \leq |\sin(x)| + |\sin(y)|$, hence $|\sin(m\theta)| \leq m|\sin(\theta)|$ and we can achieve the bound:

$$|\mu_k(\lambda_i)| \leq \frac{1}{\sqrt{(\ell+1)\ell^{k-1}}} \left(\sqrt{\frac{\ell}{\ell+1}}(k+1) + \frac{1}{\sqrt{(\ell+1)\ell}}(k-1) \right) = \frac{(\ell+1)(k+1) - 2}{(\ell+1)\sqrt{\ell^k}}. \tag{6}$$

Now observe that $\pi^{(k)} = \pi P^{(k)}$, and hence $\pi^{(k)} - s = (\pi - s)\, P^{(k)}$. The difference of two probability distributions is orthogonal for the standard L^2 scalar product to the vector u from Theorem 7 item 3. Since $\mathcal{E}$ is not orthogonal to u, by Theorem 7 item 3 we conclude that $\pi - s$ is in the linear span of the eigenvectors of A corresponding to eigenvalues different from $\ell + 1$. Since A is self-adjoint with respect to Q, using Equation (6) we have

$$\|(\pi - s)P^{(k)}\|_Q \leq \frac{(\ell+1)(k+1) - 2}{(\ell+1)\sqrt{\ell^k}} \|\pi - s\|_Q \tag{7}$$

The definition of K and M from Theorem 7 tells that $K\|\pi^{(k)} - s\|_{L^1} \leq \|\pi^{(k)} - s\|_Q$, and $\|\pi - s\|_Q \leq M$. We obtain the result recalling that the total variation distance between two probability distributions is half of the L^1 distance, see e.g. [LP17, Proposition 4.2]. □

Under the assumption that the eigenvalues of the adjacency matrix of G are strictly contained in the Hasse interval (so there are no eigenvalues equal to $\pm 2\sqrt{\ell}$), Theorem 11 can be improved: the linear factor $(k+1)$ can be replaced by a constant which does not depend on k. Indeed, as $\pm 2\sqrt{\ell}$ is not an eigenvalue, $\sin(\theta)$ in Equation 5 never vanishes. If we write $|\sin(\theta)| \geq \varepsilon$ for some $\varepsilon > 0$, we obtain

$$|\mu_k(\lambda_i)| \leq \left(\varepsilon\sqrt{\ell^k}\right)^{-1}$$

which can be used in place of Equation 6. Observe that, even with this improvement, the bound will not be sharp, because in the bound of Equation 7 we consider only the eigenvalues with greatest modulus, but the other eigenvalues of A have smaller modulus.

This argument in turn improves Lemma 14, where the linear factor k can be replaced by a constant independent of k.

4 Proof of Knowledge

Our goal is to provide a PoK of an isogeny walk $\phi : E_0 \to E_1$ between two supersingular curves defined over $\mathbb{F}_{p^2}$ that can be seamlessly plugged in a distributed SECUER generation protocol. For this, we need the following properties:

1. Compatible with any pair of curves (E_0, E_1); this rules out [GPS17, GPS20], which is restricted to a special starting curve E_0, and [DG19] and derivatives, which are restricted to curves defined over $\mathbb{F}_p$.
2. Statistically ZK, so that the security of the final SECUER does not hinge on computational assumptions brought in by the PoK; this rules out all other isogeny-based PoKs in the literature.
3. Post-quantum secure, possibly relying on as few additional assumptions as possible; this rules out many generic ZK proof systems.
4. Possibly compatible with any walk length and any base field $\mathbb{F}_{p^2}$.
5. Usable in practice for cryptographically sized finite fields.

Our new PoK inherits from the SIDH-based Σ-protocol of De Feo, Jao and Plût [DFJP14], and from the recent developments of De Feo, Dobson, Galbraith and Zobernig [DDGZ22]. The common theme to all of them is to construct random SIDH squares (see (1)) on top of the secret isogeny $\phi : E_0 \to E_1$ and to reveal some, but not all of the edges ψ, ψ', ϕ' in response to a challenge. The reason these protocols are not statistically ZK is that the side ϕ' is strongly correlated to the parallel side ϕ (often unique given E_2) and can thus easily be distinguished by an unbounded adversary.

Our first idea is to *make the walk ψ long enough* that the distribution of (E_2, ϕ') becomes statistically close to the uniform distribution on supersingular curves with isogenies of degree $\deg(\phi)$. To prove it, we will use the properties of isogeny graphs with level structure analyzed in Section 3.

But making ψ longer is easier said than done. SIDH-based protocols are constrained in the lengths of ϕ and ψ by the form of the prime p: typically, $p + 1 = 2^a 3^b$ and then $\deg(\phi) = 2^a$ and $\deg(\psi) = 3^b$. Our second idea is to *glue several SIDH squares together* to make longer walks (see Fig. 2). We call these larger diagrams *SIDH ladders*.

A valuable side-effect of gluing SIDH squares together is that we can free ourselves from the constraints on p. All we need is that isogenies of a small prime degree ℓ coprime to $\deg(\phi)$ can be computed efficiently, then we stack vertically sufficiently many SIDH squares to make $\deg(\psi) = \ell^n$ as large as we need. In practice, we will take $\deg(\phi) = 2^m$, $\deg(\psi) = 3^n$, and the protocol will be most efficient for SIDH primes, but in full generality our protocol works for any base field and any isogeny degree.

4.1 Protocol description and analysis

Let E_0, E_1 be supersingular curves defined over a finite field $\mathbb{F}_{p^2}$, and let $\phi : E_0 \to E_1$ be a cyclic separable isogeny of smooth degree d. Let ℓ be a small prime not dividing pd. Let $\mathsf{C}(m; r)$ be a statistically hiding and computationally binding commitment scheme. Our Σ-protocol is described in Fig. 1; it depends on a parameter n, controlling the length of the ℓ-isogeny walks, that we will determine in Definition 15. The

prover consists of two stateful algorithms $(\mathsf{P}_1, \mathsf{P}_2)$: the former is randomized and produces a commitment $(\mathsf{com}_2, \mathsf{com}_3)$, the latter receives a ternary challenge $\mathsf{chall} \in \{-1, 0, 1\}$ and produces a deterministic response resp. The verifier is a deterministic algorithm that receives $\big((\mathsf{com}_2, \mathsf{com}_3), \mathsf{chall}, \mathsf{resp}\big)$ and outputs a bit indicating whether or not the proof is accepted.

$\mathsf{P}_1(E_0, E_1, \phi, n)$:
1: Sample a random cyclic isogeny $\psi : E_0 \to E_2$ of degree ℓ^n;
2: Construct the SIDH ladder $(E_0, E_1, E_2, E_3, \phi', \psi')$ on (ϕ, ψ);
3: Sample random strings r_2, r_3;
4: **return** $\big(\mathsf{C}(E_2; r_2), \mathsf{C}(E_3; r_3)\big)$.

$\mathsf{P}_2(\mathsf{chall})$:
1: **if** $\mathsf{chall} == -1$ **then**
2: **return** (ψ, E_2, r_2);
3: **else if** $\mathsf{chall} == 1$ **then**
4: **return** (ψ', E_3, r_3);
5: **else if** $\mathsf{chall} == 0$ **then**
6: **return** $(\phi', E_2, r_2, E_3, r_3)$.

$\mathsf{V}(E_0, E_1, d, n, (\mathsf{com}_2, \mathsf{com}_3), \mathsf{chall}, \mathsf{resp})$:
1: **if** $\mathsf{chall} == -1$ **then**
2: $(\psi, E_2, r_2) = \mathsf{resp}$;
3: Check $\mathsf{com}_2 = \mathsf{C}(E_2; r_2)$;
4: Check ψ is an ℓ^n-isogeny $E_0 \to E_2$;
5: **else if** $\mathsf{chall} == 1$ **then**
6: $(\psi', E_3, r_3) = \mathsf{resp}$;
7: Check $\mathsf{com}_3 = \mathsf{C}(E_3; r_3)$;
8: Check ψ' is an ℓ^n-isogeny $E_1 \to E_3$;
9: **else if** $\mathsf{chall} == 0$ **then**
10: $(\phi', E_2, r_2, E_3, r_3) = \mathsf{resp}$;
11: Check $\mathsf{com}_2 = \mathsf{C}(E_2; r_2)$;
12: Check $\mathsf{com}_3 = \mathsf{C}(E_3; r_3)$;
13: Check ϕ' is a cyclic d-isogeny $E_2 \to E_3$.

Fig. 1: Interactive proof of knowledge of a cyclic isogeny $\phi : E_0 \to E_1$ of degree d.

Proposition 12. *The Σ-protocol in Fig. 1 is correct for the relation*

$$\mathcal{R}_d = \{((E_0, E_1), \phi) \mid \phi : E_0 \to E_1 \text{ is a cyclic } d\text{-isogeny}\}.$$

Assuming the commitment C is computationally binding, it is 3-special sound for the relation

$$\mathcal{R}^\star = \{((E_0, E_1), \chi) \mid \chi : E_0 \to E_1 \text{ is a cyclic } \ell^{2i}d\text{-isogeny for some } 0 \le i \le n\}.$$

More precisely, there is a probabilistic polynomial time algorithm that, given three successful transcripts of the protocol with same commitments and distinct challenges, either recovers a witness $\chi : E_0 \to E_1$, or opens one of the commitments $\mathsf{C}(E_i; r_i)$ to two distinct values (breaking the binding property).

Proof. Correctness. Suppose that the prover $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ and the verifier V follow the protocol. First note that, since the degree d of ϕ is smooth, the SIDH ladder in P_1 can be constructed as described in Section 4.2. Then it is clear that the commitments open successfully, and the verifier accepts the transcript for any challenge.
3-special soundness. Given three accepting transcripts $(\mathsf{com}, -1, \mathsf{resp}_{-1})$, $(\mathsf{com}, 0, \mathsf{resp}_0)$ and $(\mathsf{com}, 1, \mathsf{resp}_1)$, recover $(\phi', E_2, r_2, E_3, r_3) = \mathsf{resp}_0$ where $\phi' : E_2 \to E_3$ is an isogeny. If the curves in resp_{-1} and resp_1 are not equal to E_2 and E_3 respectively, then we can open one of the commitments $\mathsf{C}(E_2; r_2)$ or $\mathsf{C}(E_3; r_3)$ to two distinct outputs. Otherwise, we have $\mathsf{resp}_{-1} = (\psi, E_2, r_2)$ and $\mathsf{resp}_1 = (\psi', E_3, r_3)$ where $\psi : E_0 \to E_2$ and $\psi' : E_1 \to E_3$ are isogenies. Therefore $\chi' = \widehat{\psi'} \circ \phi' \circ \psi$ is an isogeny from E_0 to E_1 of degree $\ell^{2n}d$. Factoring out the non-cyclic part of χ', we extract a cyclic isogeny $\chi : E_0 \to E_1$ of degree $\ell^{2i}d$ such that $\chi' = [\ell^{2(n-i)}] \circ \chi$ for some $0 \le i \le n$; however, like in the original SIDH PoK [DDGZ22, GPV21], we cannot guarantee that $i = 0$. □

We are now going to define the simulator for proving ZK. Simulating $\mathsf{chall} = \pm 1$ is easy, however how well we can simulate the case $\mathsf{chall} = 0$ depends on the parameter n given to P_1. The opening $(E_2, \phi' : E_2 \to E_3)$ can be equivalently viewed as the curve with level d Borel structure $(E_2, \ker(\phi'))$. Our goal is to have this opening distributed like a "random" vertex in the graph $G = G(p, d, \ell)$. To this effect, we define two sequences $D_1(k)$ and $D_2(k)$ of probability distributions on G, and we show that they converge as k grows.

Definition 13. *Let $\phi : E_0 \to E_1$ be a cyclic separable isogeny of degree d. Define*

$$\begin{aligned} \mathcal{D}_1(k) &= \big\{(E_0/K, \tau(\ker(\phi)) \;\big|\; K \leftarrow \mathcal{C}_E(\ell^k),\ \tau : E_0 \to E_0/K\big\}, \\ \mathcal{D}_2(k) &= \big\{(E_0/K, C) \;\big|\; K \leftarrow \mathcal{C}_E(\ell^k),\ C \leftarrow \mathcal{C}_{E_0/K}(d)\big\}, \end{aligned} \tag{8}$$

where $\mathcal{C}_E(f)$ is the uniform distribution on the cyclic subgroups of order f of E, up to $\mathrm{Aut}(E)$.

Lemma 14. *Keep notations as above, fix a positive real number ε, and let k be a positive integer such that*

$$\tau(p,d,\ell,k) = \tfrac{1}{4}(p-1)^{1/2}\Big(1+\sqrt{d}\prod_{\substack{q\mid d\\ q\ prime}}(1+\tfrac{1}{q})^{1/2}\Big)\cdot\Big(k+\tfrac{\ell-1}{\ell+1}\Big)\cdot\ell^{-k/2}\leq\varepsilon\,,$$

then $d_{TV}\left(\mathcal{D}_1(k),\mathcal{D}_2(k)\right)\leq\varepsilon$, where d_{TV} is the total variation distance between the two distributions, also known as statistical distance.

Proof. We bound the statistical distance of each of $\mathcal{D}_1(k)$ and $\mathcal{D}_2(k)$ from the stationary distribution of $G(p,d,\ell)$, as determined in Theorem 7, then we conclude with the triangle inequality. For $\mathcal{D}_1(k)$, we can directly apply Theorem 11, but $\mathcal{D}_2(k)$ needs more care.

Let G_0 be the classical isogeny graph. This can be thought of as the graph with $d=1$ Borel level structure. Let s_0 be the stationary distribution on G_0. Consider the projection map $P\colon G\to G_0$ which forgets the level structure. The push-forward distribution $P_*\mathcal{D}_2(k)$ is the distribution of the length k non-backtracking walks starting at E_0, so we can bound its total variation distance from s_0 using Theorem 11. For any probability distribution π on G_0 let us denote $\widetilde{\pi}$ the distribution on G obtained by first choosing E with distribution π and then choosing C uniformly inside the set of cyclic subgroups of order d. Notice that for each two subgroups C,C', the pair (E,C) defines the same vertex as (E,C') if and only if there exists an automorphism of E sending C to C'. This, together with the fact that the set of C's for a single E has cardinality $[\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z}):H]$, implies

$$\widetilde{\pi}((E,C)) = \frac{|\operatorname{Aut}(E)/\operatorname{Aut}(E,C)|}{[\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z}):H]}\pi(E)\,,$$

where H is the subgroup of upper triangular matrices. The above formula, together with (2), implies that for every probability distribution π on G_0 and every subset A of G_0, one has $\widetilde{\pi}(P^{-1}(A))=\pi(A)$. In turn, this means that for π_1,π_2 probability measures on G_0, we have $d_{TV}\left(\pi_1,\pi_2\right)=d_{TV}\left(\widetilde{\pi}_1,\widetilde{\pi}_2\right)$. One can then check by direct computation that $s=\widetilde{s_0}$. We conclude that $d_{TV}(\mathcal{D}_2(k),s)=d_{TV}(P_*\mathcal{D}_2(k),s_0)$, and the right hand side can be bound using Theorem 11. □

Definition 15. *Given p, d, ℓ and m, define*

$$n(p,d,\ell,m)=\min\left\{k\in\mathbb{Z}\mid\tau(p,d,\ell,k)\leq 2^{-m}\right\}.$$

Proposition 16. *Let λ be a security parameter and let $n=n(p,d,\ell,\lambda)$. The Σ-protocol of Fig. 1 is statistically SHVZK for the relation $\mathcal{R}_d$ defined in Proposition 12, assuming the commitment C is statistically hiding.*

Proof. We simulate the honest prover for each of the three challenges as follows.

$\mathsf{chall}=-1$. Sample a random isogeny $\psi: E_0\to E_2$ of degree ℓ^n, and random strings r_2,r_3. Set $\mathsf{com}_2=\mathsf{C}(E_2;r_2)$ and set $\mathsf{com}_3=\mathsf{C}(\bot;r_3)$. Return $(\mathsf{com}_2,\mathsf{com}_3),\mathsf{chall},(\psi,E_2,r_2)$.

The isogeny ψ is distributed exactly like in the real protocol, thus this transcript is valid. Because C is statistically hiding, an adversary cannot distinguish com_3 from a real commitment.

$\mathsf{chall}=1$. This is nearly identical to the above. The simulator samples $\psi': E_1\to E_3$ of degree ℓ^n and random strings r_2,r_3. It sets $\mathsf{com}_2=\mathsf{C}(\bot;r_2)$ and $\mathsf{com}_3=\mathsf{C}(E_3;r_3)$, and returns $(\mathsf{com}_2,\mathsf{com}_3)$, chall, (ψ',E_3,r_3).

Because ℓ is coprime to d, if ψ is uniformly distributed so is ψ'. Then, the transcript is indistinguishable from a real one as before.

$\mathsf{chall}=0$. Sample a random isogeny $\psi: E_0\to E_2$ of degree ℓ^n, and then a random isogeny $\rho: E_2\to E_3$ of degree d. Sample random strings r_2,r_3 and set $\mathsf{com}_2=\mathsf{C}(E_2;r_2)$ and $\mathsf{com}_3=\mathsf{C}(E_3;r_3)$. Return $(\mathsf{com}_2,\mathsf{com}_3),\mathsf{chall},(\rho,E_2,r_2,E_3,r_3)$.

Thanks to Lemma 14, the statistical distance between the simulated $(E_2,\ker(\rho))$ and $(E_2,\psi(\ker(\phi)))$ is negligible. Because ρ is uniquely determined from $\ker(\rho)$, and the real response ϕ' by $\psi(\ker(\phi))$, an adversary has negligible probability of distinguishing the transcript output by the simulator. □

4.2 Executing the protocol

The protocol we just described crucially depends on the ability to construct a commutative square with sides of degrees d and ℓ^n. The SIDH setting has $p+1 = d \cdot \ell^n$ so that the square can be constructed by simply pushing a single kernel point for ψ through ϕ and vice versa. We refer to such a square as an *SIDH square*. For more general choices of ℓ^n and d, the kernels are typically generated by points defined over very large extension fields, requiring superpolynomial space. We efficiently construct such "larger" squares by gluing together several SIDH squares in what we call *SIDH ladders*, as depicted in Fig. 2.

For simplicity, we shall present the case $d = (2^a)^w$ and $\ell^n = (3^b)^h$, where 2^a and 3^b are the side lengths of an SIDH square, and w and h are positive integers defining the **w**idth and **h**eight of the ladders in units of SIDH squares. However, the technique generalizes easily to any coprime d and ℓ^n, as long as isogenies of degrees d and ℓ can be efficiently computed.

First, notice that there always exist some choice of a and b such that points (and hence kernel subgroups) of orders 2^a and 3^b can be represented efficiently. This is clear if the prime p is a SIDH prime where $2^a 3^b \mid (p+1)$, but for a generic prime p, one can set $a = b = 1$: Points of order 2 and 3 are defined over a small extension field and can thus be efficiently represented. Moreover, any isogeny of degree $(3^b)^h$ is the composition of h isogenies of degree 3^b each, which can be stored as a sequence of h kernel generators which are efficiently representable.

This means that the prover can generate the isogeny $\psi : E_0 \to E_2$ in step 2 of P_1 by generating a random kernel $K_{1,0}$ on E_0, computing the isogeny $\psi_{1,0} : E_0 \to E_0/K_{1,0} =: E_{1,0}$, generating a random kernel $K_{2,0}$ on $E_{1,0}$ such that $K_{2,0} \cap \ker \widehat{\psi_{1,0}} = \{0\}$ to prevent backtracking, and repeating the process h times to obtain a chain of h isogenies $\psi_{i,0} : E_{i-1,0} \to E_{i,0}$. The curve E_2 is the codomain of the last isogeny $\psi_{h,0}$, i.e., $E_2 = E_{h,0}$.

If the width w of the ladder is one, the prover can now recursively push the kernel G of the isogeny $\phi = \phi_{0,1}$ through the isogenies $\psi_{i,0}$ to obtain its image G_i on each curve $E_{i,0}$. Each horizontal isogeny $\phi_{0,i}$ has kernel G_i, and the prover can compute the kernel of the right-side vertical isogeny $\psi'_{i,0}$ as the image of the kernel of $\psi_{i,0}$ under the isogeny $\phi_{i-1,1}$. Since each square composed of $(E_{i,0}, E_{i+1,0}, E'_{i,0}, E'_{i+1,0})$ is a commutative diagram, so is the larger square (E_0, E_1, E_2, E_3). In the general case where $w > 1$, the prover can use a similar approach for the horizontal isogeny ϕ as used for the vertical isogeny ψ: The isogeny ϕ can be written as the composition of w isogenies $\phi_{0,w} \circ \ldots \circ \phi_{0,1}$ of degree 2^a and their kernels can be mapped through the vertical isogenies. In other words, the prover can glue horizontally w compatible ladders, one for each factor $\phi_{0,i}$ of ϕ. The right descending isogenies of each ladder are used as the left descending isogenies of the next one. This allows the prover to compute $w \times h$ SIDH squares in such a way that the curves (E_0, E_1, E_2, E_3) and the isogenies between them form a commutative diagram. This is illustrated in Fig. 2. For the challenges $\mathsf{chall} = \pm 1$, the prover reveals the isogenies $\psi_{i,0}$ of the leftmost squares, or the isogenies $\psi_{i,w}$ of the rightmost squares. For the challenge $\mathsf{chall} = 0$, the prover responds with the isogenies $\phi_{h,i}$ of the bottom squares.

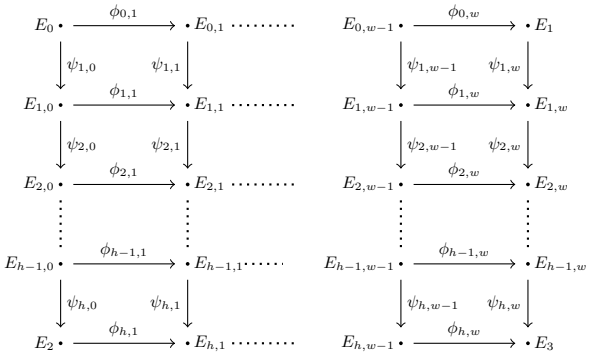

Fig. 2: An SIDH ladder.

Verification consists of evaluating (depending on the challenge) either w or h isogenies of degree 2^a or 3^b, which can be done efficiently. Generating the proof is slower, as the prover needs to fill in all the $w \times h$ SIDH squares that make up the ladder. The proving complexity is thus quadratic in w and h, while the verification complexity is linear in w and h. However, the complexity of computing an SIDH square with

$\mathsf{P}_{\mathsf{NIZK}}(E_0, E_1, \phi, n, N)$:

1: For each $i \in [N]$, sample $(\mathsf{com}_{2,i}, \mathsf{com}_{3,i}) \leftarrow \mathsf{P}_1(E_0, E_1, \phi, n)$.
2: Set $(\mathsf{chall}_1, \ldots, \mathsf{chall}_N) = H\big((\mathsf{com}_{2,1}, \mathsf{com}_{3,1}), \ldots, (\mathsf{com}_{2,N}, \mathsf{com}_{3,N})\big)$.
3: For each $i \in [N]$, set $\mathsf{resp}_i = \mathsf{P}_2(\mathsf{chall}_i)$.
4: **return** $\Pi = \big(\{(\mathsf{com}_{2,i}, \mathsf{com}_{3,i}, \mathsf{resp}_i)\}_{i\in[N]}\big)$.

$\mathsf{V}_{\mathsf{NIZK}}(E_0, E_1, \Pi, N)$:

1: Parse Π as $\big(\{(\mathsf{com}_{2,i}, \mathsf{com}_{3,i}, \mathsf{resp}_i)\}_{i\in[N]}\big)$.
2: Compute $(\mathsf{chall}_1, \ldots, \mathsf{chall}_N) = H\big((\mathsf{com}_{2,1}, \mathsf{com}_{3,1}), \ldots, (\mathsf{com}_{2,N}, \mathsf{com}_{3,N})\big)$.
3: For each $i \in [N]$, compute $b_i = \mathsf{V}(E_0, E_1, (\mathsf{com}_{2,i}, \mathsf{com}_{3,i}), \mathsf{chall}_i, \mathsf{resp}_i)$.
4: Output $b = \wedge_{i\in[N]} b_i$.

Fig. 3: The NIZK.

degrees 2^a or 3^b is only quasilinear in a and b using sparse strategies [DFJP14]; thus, maximizing the size of SIDH squares improves performance, which explains why SIDH primes are the most efficient scenario for this proof. If the degree of the isogenies and the size of the underlying field are kept constant, in the SIDH setting we have that $2^a 3^b \mid (p+1)$ for large values of a and b (in the order of several hundreds), and thus w and h can be small. For a generic prime, the prover might need to set $a = b = 1$ and work with large values of w and h, incurring a quadratic cost, besides possibly having to compute points over an extension field of degree bounded by a small constant.

Remark 17. Above, we assumed that the degree of the witness ϕ was $d = (2^a)^w$. As mentioned before, this can be generalized to any witness ϕ of smooth degree $d = d_1 \ldots d_w$ as far as the d_i-torsion groups are accessible (ideally, one should have $E_0[d_i] \subseteq E_0(\mathbb{F}_{p^2})$). In this case, one factors ϕ as $\phi = \phi_{0,w} \circ \ldots \circ \phi_{0,1}$ where each isogeny $\phi_{0,i}$ has degree d_i, and constructs compatible ladders for each $\phi_{0,i}$.

5 Distributed SECUER Setup and its Security

In this section, we formally describe the distributed SECUER setup protocol and prove its security under a security definition using the simplified universal composability (SUC) framework due to Canetti, Cohen, and Lindell [CCL15] in the real/ideal world paradigm. Our security definitions consider a *dishonest majority* corruption model, wherein the adversary can corrupt up to $t-1$ of the t participating parties in the distributed SECUER setup protocol. The protocol uses a non-interactive version of the Σ-protocol described in Section 4. We begin by formally describing this non-interactive zero-knowledge (NIZK) PoK protocol.

5.1 The NIZK protocol

We transform the Σ-protocol of Section 4 into a NIZK using the standard Fiat-Shamir heuristic [FS87] for transforming interactive PoK protocols into NIZK proofs, albeit with the difference that soundness and zero-knowledge hold for slightly different languages.

The NIZK construction. Let E_0, E_1 be supersingular curves defined over a finite field $\mathbb{F}_{p^2}$, let $\phi : E_0 \to E_1$ be a separable isogeny of smooth degree d and let $\mathsf{C}(m; r)$ be a statistically hiding and computationally binding commitment scheme. Additionally, let $\Sigma = (\mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ be the interactive PoK protocol described in Section 4, let $\lambda \in \mathbb{N}$ be the security parameter, let ℓ be a small prime not dividing dp, let $n = n(p, d, \ell, \lambda)$, and let $N = \mathsf{poly}(\lambda)$ be a fixed polynomial. Finally, let $H : \{0,1\}^* \to \{-1, 0, 1\}^N$ be a random oracle. The NIZK proof system consists of a pair of algorithms $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ as described in Fig. 3. The prover algorithm $\mathsf{P}_{\mathsf{NIZK}}$ is randomized and produces a proof Π. The verifier algorithm $\mathsf{V}_{\mathsf{NIZK}}$ is deterministic; it receives the proof Π and outputs a bit $b \in \{0, 1\}$ indicating whether or not the proof is accepted.

Correctness, Extractability and ZK. Correctness follows immediately from the correctness of the underlying Σ-protocol. We state and prove the following propositions for extractability and ZK.

Proposition 18. *Assuming that $\Sigma = (\mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ satisfies 3-special soundness with respect to the relation $\mathcal{R}^\star$ (described in Proposition 12) and that H is a random oracle, the NIZK $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ satisfies extractability (and hence soundness) with respect to the relation $\mathcal{R}^\star$.*

Proof. We provide an informal proof overview. We begin by noting that Σ is a public-coin protocol, and that there exists a probabilistic polynomial-time algorithm that extracts a witness from 3 accepting transcripts corresponding to N parallel executions of Σ w.r.t. the same statement. Consequently, we can invoke the generalized forking lemma of [BCC+16] to argue the existence of a probabilistic polynomial-time witness-extraction algorithm for NIZK. This completes the proof of extractability (and hence, soundness) for NIZK. □

Proposition 19. *Assuming that $\Sigma = (\mathsf{P}_1, \mathsf{P}_2, \mathsf{V})$ is statistically SHVZK for the relation R_d (described in Proposition 16) and that H is a random oracle, the NIZK $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ is statistically ZK for the relation R_d.*

Proof. We again provide an informal proof overview. Let Sim_Σ be a ZK simulator that simulates an accepting transcript for the underlying Σ-protocol (as described in the proof of ZK for Σ). We construct a ZK simulator $\mathsf{Sim}_{\mathsf{NIZK}}$ that simulates an accepting proof as follows:

1. $\mathsf{Sim}_{\mathsf{NIZK}}$ simulates the random oracle H as follows: it maintains a local table consisting of tuples of the form $(x, y) \in \{0,1\}^* \times \{-1,0,1\}^N$. On receiving a query $x \in \{0,1\}^*$ from the adversary $\mathcal{A}$, it looks up this table to check if an entry of the from (x, y) exists. If yes, it responds with y. Otherwise, it responds with a uniformly sampled $y \leftarrow \{-1,0,1\}^N$, and programs the random oracle as $H(x) := y$ by adding the entry (x, y) to the table.
2. For each $i \in [N]$, $\mathsf{Sim}_{\mathsf{NIZK}}$ internally invokes the simulator Sim_Σ for the underlying Σ-protocol to obtain the i-th accepting transcript of the form
$$((\mathsf{com}_{2,i}, \mathsf{com}_{3,i}), \mathsf{chall}_i, \mathsf{resp}_i)\,.$$
3. At this point, $\mathsf{Sim}_{\mathsf{NIZK}}$ aborts if the adversary $\mathcal{A}$ has already issued a random oracle query on the input $x = \big((\mathsf{com}_{2,1}, \mathsf{com}_{3,1}), \ldots, (\mathsf{com}_{2,N}, \mathsf{com}_{3,N})\big)$.
4. Otherwise, $\mathsf{Sim}_{\mathsf{NIZK}}$ programs the random oracle as
$$H\big((\mathsf{com}_{2,1}, \mathsf{com}_{3,1}), \ldots, (\mathsf{com}_{2,N}, \mathsf{com}_{3,N})\big) := (\mathsf{chall}_1, \ldots, \mathsf{chall}_N),$$
and outputs the simulated proof as $\Pi = \big(\{(\mathsf{com}_{2,i}, \mathsf{com}_{3,i}, \mathsf{resp}_i)\}_{i\in[N]}\big)$.

We note that $\mathsf{Sim}_{\mathsf{NIZK}}$ runs in polynomial time as long as Sim_Σ runs in polynomial time. Additionally, if $\mathsf{Sim}_{\mathsf{NIZK}}$ does not abort, it outputs a simulated proof that is distributed in a statistically indistinguishable manner from the distribution of a real proof, assuming that Sim_Σ outputs a simulated accepting transcript with distribution statistically indistinguishable from a real accepting transcript for Σ. Finally, $\mathsf{Sim}_{\mathsf{NIZK}}$ aborts with only negligible probability, since the adversary $\mathcal{A}$ guesses $((\mathsf{com}_{2,i}, \mathsf{com}_{3,i}), \mathsf{chall}_i, \mathsf{resp}_i)$ for each $i \in [n]$ with at most negligible probability. This completes the proof of statistical ZK for NIZK. □

5.2 Our distributed SECUER setup protocol

We now move to the distributed SECUER setup protocol. Let $P_1, \ldots, P_t$ be a set of t participating parties and let E_0 be some fixed starting curve. In a nutshell, the idea is to have the parties act sequentially: each P_i at its own turn performs a secret random walk $E_{i-1} \to E_i$ and broadcasts E_i and a NIZK PoK of the secret walk. We claim that, as long as one party is honest, the final curve E_t is a SECUER.

To get any security guarantee, we need to carefully set the parameters of the random walk $E_{i-1} \to E_i$. The natural choice is to fix some small prime q, not dividing ℓp, and to take a random walk long enough that the distribution of E_i is negligibly far from the stationary distribution on the q-isogeny graph $G(p, 1, q)$. For example we may set $q = 2$ and $\ell = 3$, then Theorem 11 provides a precise bound to set the length $\delta = n(p, 1, q, \lambda)$ of the q-walk as a function of the security parameter, and ultimately the parameter $n(p, q^\delta, \ell, \lambda)$ of the PoK.

Remark 20. For increased efficiency, we may choose to perform shorter q-walks $E_{i-1} \to E_i$ of length $\log_q(p)$. This length approximates the diameter of the supersingular q-isogeny graph; hence, it ensures that the secret isogeny can reach almost any curve in the graph.

Under mild assumptions, this choice would still yield a secure protocol, but it would also make the security proof somewhat more involved. For this reason, we shall stick here to the more conservative choice of walking long enough to ensure nearly stationary distribution of E_i.

We formally describe the protocol (referred to as Γ_{SECUER} henceforth). Assume that E_0 is known to all the parties at the start. Let $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ be the non-interactive proof as described above. The protocol Γ_{SECUER} proceeds in t rounds while only using broadcast channels of communication, where round-i for each $i \in [t]$ is as follows:

- Party P_i performs a q-isogeny walk starting at curve E_{i-1} and ending at curve E_i (where E_{i-1} and E_i are both supersingular curves defined over $\mathbb{F}_{p^2}$), such that party P_i knows a separable isogeny $\phi_i : E_{i-1} \to E_i$ of degree q^δ, where $\delta = n(p, 1, q, \lambda)$.
- Party P_i generates $\Pi_i \leftarrow \mathsf{P}_{\mathsf{NIZK}}(E_{i-1}, E_i, \phi_i, n, N)$, where $n = n(p, q^\delta, \ell, \lambda)$, and broadcasts (E_i, Π_i) to all other parties.
- Each party P_j for $j \in [t] \setminus \{i\}$ verifies the NIZK proof Π_i by computing $b_i = \mathsf{V}_{\mathsf{NIZK}}(E_{i-1}, E_i, \Pi_i, N)$. If $b_i = 0$ (i.e., the proof is invalid), P_j aborts.

At the end of round-t, all parties output E_t to be the final output curve.

Correctness. Correctness of Γ_{SECUER} follows immediately from the correctness guarantees of the NIZK.

5.3 Proof of security for Γ_{SECUER}

We now present the proof of security for Γ_{SECUER} using the simplified universal composability (SUC) framework [CCL15] in the real/ideal world paradigm. We consider a *dishonest majority* corruption model, wherein the adversary can corrupt up to $(t-1)$ of the t participating parties.

The ideal functionality. Intuitively, the ideal functionality for distributed SECUER setup should simply take as input the initial curve E_0 and output a SECUER E_t. It is however not obvious how to model the property of being a SECUER in the plain SUC model: a game based definition, stating that an adversary who can compute $\text{End}(E_t)$ can be used to break some other assumption, appears to be more appropriate.

Thus, we prove security in two steps. First, we prove that Γ_{SECUER} securely emulates a *less-than-ideal* functionality $\mathcal{F}^*_{\text{SECUER}}$ (described in Fig. 4) that enforces that: (a) for each $i \in [t]$, if a corrupt party P_i outputs a curve E_i, it must know a valid isogeny $\phi_i : E_{i-1} \to E_i$, and (b) for each $i \in [t]$, if an honest party P_i outputs a curve E_i, then the corresponding isogeny $\phi_i : E_{i-1} \to E_i$ is hidden from the adversary. This step relies on the extractability and ZK properties of the NIZK protocol described above. Next, we prove that, assuming the hardness of the endomorphism ring problem in the $\mathcal{F}^*_{\text{SECUER}}$-hybrid model, the output curve E_t is a SECUER, i.e. that the (malicious) adversary cannot compute $\text{End}(E_t)$.

Theorem 21. *Assuming that* $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ *satisfies extractability and zero-knowledge, and assuming the hardness of the endomorphism ring problem (Definition 1) and GRH, the output* E_t *of the protocol* Γ_{SECUER} *is a* SECUER *if at least one party* P_{i^*} *for some* $i^* \in [t]$ *is honest.*

Secure emulation of $\mathcal{F}^*_{\text{SECUER}}$. We now prove that Γ_{SECUER} securely emulates the less-than-ideal functionality $\mathcal{F}^*_{\text{SECUER}}$. Our proof is in the real/ideal world paradigm defined formally as follows.

The real world. The following entities engage in the real protocol Γ_{SECUER}: (i) a set $\mathcal{H} \subseteq [t]$ of honest parties, (ii) a real-world adversary $\mathcal{A}$ controlling a set $\mathcal{C} \subset [t]$ of corrupt parties, and (iii) the environment $\mathcal{E}$ that provides E_0 to each party, interacts with the real-world adversary $\mathcal{A}$, receives the final output curve E_t from the honest parties, and eventually outputs a bit $b \in \{0, 1\}$.

$\underline{\mathcal{F}^*_{\text{SECUER}}(E_0, i \in [t])}$

- Let $\mathcal{H}_i \subseteq [i-1]$ be the set of honest parties, and let $\mathcal{C}_i \subseteq [i-1]$ be the set of corrupt parties among the first $(i-1)$ parties $P_1, \ldots, P_{(i-1)}$.
- For each $j \in \mathcal{H}_i$, $\mathcal{F}^*_{\text{SECUER}}$ receives as input from P_j a tuple of the form (E_j, ϕ_j).
- For each $j' \in \mathcal{C}_i$, $\mathcal{F}^*_{\text{SECUER}}$ receives as input from the simulator Sim a tuple of the form $(E_{j'}, \phi_{j'})$.
- If for any $j \in [i-1]$, ϕ_j is not an isogeny from the curve E_{j-1} to the curve E_j, $\mathcal{F}^*_{\text{SECUER}}$ outputs $\perp$ and aborts.
- Otherwise, $\mathcal{F}^*_{\text{SECUER}}$ takes a random walk starting from the $(i-1)$-th curve E_{i-1} and ending in a curve E_i such that $\mathcal{F}^*_{\text{SECUER}}$ knows $\phi_i : E_{i-1} \to E_i$, where ϕ_i is a separable isogeny of degree d.
- Finally, $\mathcal{F}^*_{\text{SECUER}}$ outputs (E_i, ϕ_i) to the party P_i, and outputs E_i to the simulator Sim and to all parties P_j for $j \neq i$.

Fig. 4: The Ideal functionality $\mathcal{F}^*_{\text{SECUER}}$

The ideal world. The following entities interact with the functionality $\mathcal{F}^*_{\text{SECUER}}$: (i) A set $\mathcal{H} \subseteq [t]$ of honest parties, where for each $i \in \mathcal{H}$, party P_i directly forwards its secret isogeny to $\mathcal{F}^*_{\text{SECUER}}$, (ii) an ideal-world simulator Sim that sends inputs to $\mathcal{F}^*_{\text{SECUER}}$ on behalf of a set $\mathcal{C} \subset [t]$ of corrupt parties, and (iii) the environment $\mathcal{E}$ that provides each party with the starting curve E_0, interacts with the simulator Sim, receives the final output curve E_t from the functionality, and eventually outputs a bit $b \in \{0, 1\}$.

For any t-party SECUER setup protocol Γ_{SECUER}, any adversary $\mathcal{A}$, any simulator Sim, and any environment $\mathcal{E}$, we define the following random variables:

- $\mathsf{real}_{\Gamma_{\text{SECUER}}, \mathcal{A}, \mathcal{E}}$: denotes the output of the environment $\mathcal{E}$ after interacting with the adversary $\mathcal{A}$ during a real-world execution of Γ_{SECUER}.
- $\mathsf{ideal}_{\mathcal{F}^*_{\text{SECUER}}, \mathsf{Sim}, \mathcal{E}}$: denotes the output of the environment $\mathcal{E}$ after interacting with the simulator Sim in the ideal world.

Theorem 22. *Assuming that* $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ *satisfies extractability and zero-knowledge, for any security parameter* $\lambda \in \mathbb{N}$ *and any probabilistic polynomial time (PPT) adversary* $\mathcal{A}$*, there exists a PPT simulator* Sim *such that, for any PPT environment* $\mathcal{E}$*, we have*

$$\left|\Pr\left[\mathsf{real}_{\Gamma_{\text{SECUER}}, \mathcal{A}, \mathcal{E}} = 1\right] - \Pr\left[\mathsf{ideal}_{\mathcal{F}^*_{\text{SECUER}}, \mathsf{Sim}, \mathcal{E}} = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

Proof. We prove this theorem by constructing a PPT simulator Sim that simulates the view of the environment $\mathcal{E}$ in the ideal world. The simulator Sim receives E_0 from the environment $\mathcal{E}$, internally runs the real-world adversary $\mathcal{A}$ and the NIZK simulator $\mathsf{Sim}_{\mathsf{NIZK}}$, and proceeds in round-$i$ for $i \in [t]$ as described next. Note that we implicitly assume that Sim has rewinding access to the adversary $\mathcal{A}$ and programming access to the random oracle in the analysis below.

Case-1: Party P_i is corrupt. In this case, Sim internally runs the real-world adversary $\mathcal{A}$ to obtain the broadcast message (E_i, Π_i) corresponding to the corrupt party P_i. It then uses the extraction algorithm of NIZK to extract the corresponding witness ϕ_i. If extraction fails, Sim outputs $\perp$ and aborts. Otherwise, Sim stores (E_i, Π_i, ϕ_i) internally, and broadcasts (E_i, Π_i) as the message corresponding to the corrupt party P_i.

Case-2: Party P_i is honest. In this case, Sim invokes the ideal functionality to obtain E_i. Concretely, let $\mathcal{C}_i \subseteq [i-1]$ be the set of corrupt parties among the first $(i-1)$ parties $P_1, \ldots, P_{(i-1)}$. Sim invokes the ideal functionality $\mathcal{F}^*_{\text{SECUER}}(E_0, i)$ with the set $\{(E_{j'}, \phi_{j'})\}_{j' \in [C_i]}$. If $\mathcal{F}^*_{\text{SECUER}}$ outputs $\perp$, Sim outputs $\perp$ and aborts. Otherwise, Sim receives from $\mathcal{F}^*_{\text{SECUER}}$ the corresponding curve E_i. At this point, it invokes the simulator $\mathsf{Sim}_{\mathsf{NIZK}}$ of the NIZK protocol to obtain a simulated proof as $\overline{\Pi}_i \leftarrow \mathsf{Sim}_{\mathsf{NIZK}}(E_{i-1}, E_i, N)$, and broadcasts $(E_i, \overline{\Pi}_i)$ as the message corresponding to the honest party P_i.

Indistinguishability of views. We now prove that for the above construction of Sim, the view of $\mathcal{E}$ in the ideal world is indistinguishable from that in the real world. We prove this by a sequence of hybrids as described below (recall that $\mathcal{H} \subseteq [t]$ and $\mathcal{C} \subset [t]$ denote the set of honest and corrupt parties, respectively).

– **Hybrid-0:** In this hybrid, the distribution of messages broadcast by each party is identical to the real-world protocol γ_{SECUER}.
– **Hybrid-1:** In this hybrid, for each corrupt party P_j such that $j \in \mathcal{C}$, instead of verifying the NIZK proof Π_j using $\mathsf{V}_{\mathsf{NIZK}}$ (as in the real protocol), extract the witness ϕ_j using the the extraction algorithm of NIZK. If extraction fails, output $\perp$.
– **Hybrid-2:** In this hybrid, for each honest party P_i such that $i \in \mathcal{H}$, instead of generating the NIZK proof $\Pi_i \leftarrow \mathsf{P}_{\mathsf{NIZK}}(E_{i-1}, E_i, \phi_i, n, N)$ (as in the real protocol), generate a simulated proof as $\overline{\Pi}_i \leftarrow \mathsf{Sim}_{\mathsf{NIZK}}(E_{i-1}, E_i, N)$.
– **Hybrid-3:** In this hybrid, the distribution of messages broadcast by each party is identical to the ideal-world messages broadcast by Sim.

Lemma. *Assuming that* $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ *satisfies extractability, hybrid-0 and hybrid-1 are indistinguishable.*

Note that for $\mathcal{E}$ to distinguish between hybrid-0 and hybrid-1 with non-negligible probability, the adversary $\mathcal{A}$ must be able to produce with non-negligible probability a proof Π_j corresponding to a corrupt party P_j for $j \in \mathcal{C}$ such that $\mathsf{V}_{\mathsf{NIZK}}(E_{j-1}, E_j, \Pi_j, N) = 1$ but extraction fails. This immediately violates extractability of NIZK, thus completing the proof of the lemma.

Lemma. *Assuming that* $\mathsf{NIZK} = (\mathsf{P}_{\mathsf{NIZK}}, \mathsf{V}_{\mathsf{NIZK}})$ *satisfies ZK, hybrid-1 and hybrid-2 are indistinguishable.*

Note that for $\mathcal{E}$ to distinguish between hybrid-1 and hybrid-2 with non-negligible probability, there must exist an honest party P_i for $i \in \mathcal{H}$ and a distinguisher $\mathcal{D}$ such that

$$\left|\Pr[\mathcal{D}(E_0, E_1, \Pi_i) = 1] - \Pr[\mathcal{D}(E_0, E_1, \overline{\Pi}_i) = 1]\right| > \mathsf{negl}(\lambda),$$

where $\Pi_i \leftarrow \mathsf{P}_{\mathsf{NIZK}}(E_{i-1}, E_i, \phi_i, n, N)$, $\overline{\Pi}_i \leftarrow \mathsf{Sim}_{\mathsf{NIZK}}(E_{i-1}, E_i, N)$, and λ is the security parameter. This immediately violates the ZK property of NIZK, thus completing the proof of the lemma.

Finally, hybrid-2 and hybrid-3 are identical by inspection, thus completing the proof of Theorem 22. □

Analyzing E_t in $\mathcal{F}^*_{\text{SECUER}}$-hybrid model. Based on the above secure emulation guarantee, we now analyze the output E_t of Γ_{SECUER} in the $\mathcal{F}^*_{\text{SECUER}}$-hybrid model. Concretely, we state and prove the following theorem.

Theorem 23. *Assuming the hardness of the endomorphism ring problem and GRH, the output E_t of $\mathcal{F}^*_{\text{SECUER}}(E_0, t)$ is a* SECUER *if at least one party is honest.*

To prove this theorem, we first prove the following lemma.

Lemma 24. *Assuming the hardness of the endomorphism ring problem, the output E_i of $\mathcal{F}^*_{\text{SECUER}}(E_0, i)$ for $i \in [t]$ is a* SECUER *whenever P_i is honest.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ corrupting a dishonest majority of the parties that efficiently computes the endomorphism ring of E_i with non-negligible probability. Also assume that $\mathcal{A}$ corrupts all of $P_1, \ldots, P_{i-1}$. We can use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ that solves the endomorphism ring problem. The algorithm $\mathcal{B}$ receives as input a uniformly random curve $E^*/\mathbb{F}_{p^2}$, internally runs the adversary $\mathcal{A}$ to emulate the outputs of the corrupt parties $P_1, \ldots, P_{i-1}$, and finally feeds $\mathcal{A}$ with $E_i := E^*$. The view of the adversary $\mathcal{A}$ is properly simulated by $\mathcal{B}$, since E_i output by $\mathcal{F}^*_{\text{SECUER}}$ and E^* provisioned by $\mathcal{B}$ are statistically indistinguishable (here we use Theorem 11, which crucially follows from the honest party taking a q-walk of length $n(p, 1, q, \lambda)$). Finally, $\mathcal{B}$ uses $\mathcal{A}$ to recover the endomorphism ring of E^* with non-negligible probability. This concludes the proof of Lemma 24. □

We now prove Theorem 23. We break the proof into two cases: (i) when P_t is honest, and (ii) when P_t is corrupt. The proof for case (i) is immediate from Lemma 24. Hence, we focus on case (ii). Let $\mathcal{H} \subseteq [t]$ be the set of honest parties, and let $i^* = \max(\{i : P_i \in \mathcal{H}\})$. By Lemma 24, E_{i^*} must be a SECUER. Now, suppose that E_t is not a SECUER, i.e., there exists an adversary $\mathcal{A}$ corrupting dishonest majority of the parties that efficiently computes the endomorphism ring of E_t with non-negligible probability. Since all of $P_{i^*+1}, \ldots, P_t$ are corrupt, $\mathcal{A}$ knows a walk from E_{i^*} to E_t in the ℓ-isogeny graph. However, since E_t is not a SECUER, $\mathcal{A}$ can use the reduction [Wes22b] (assuming GRH) to recover $\mathrm{End}(E_{i^*})$, thereby violating Lemma 24. This completes the proof of Theorem 23. □

Finally, the proof of Theorem 21 follows immediately from the proofs of Theorem 22 and Theorem 23, which completes the proof of security for our distributed SECUER setup protocol Γ_{SECUER}.

Table 1: Parameters and corresponding secret/proof size for each of the four SIKE finite fields.

		Degree		SIDH Squares		Size (kB)	
$\log(p)$	Reps	2-isog.	3-isog.	Columns	Rows	Secret	Proof
434	219	705	890	4	7	0.99	191.19
503	219	774	977	4	7	1.13	215.75
610	329	1010	1275	4	7	1.39	404.32
751	438	1280	1616	4	7	1.69	662.63

6 Implementation and Results

In this section, we report on our proof-of-concept implementation of our proof of knowledge (Section 4), including a discussion of proof sizes and running times. Moreover, we lay out concretely how one may deploy the trusted setup protocol from Section 5 in the real world.

Parameter selection. The base-field primes p in our proof-of-knowledge implementation are taken from the four SIKE parameter sets `p434`, `p503`, `p610`, and `p751`. As discussed in Section 4.2, our proof of knowledge achieves its optimal efficiency for SIDH-style primes. Moreover, those primes have been featured extensively in the literature, and thus appear to be the obvious choice to demonstrate our proof of knowledge. That said, we stress once more that our techniques are generic and can be applied in any choice of characteristic.

We use the degree $q = 2$ for the random walks $E_i \to E_{i-1}$, and $\ell = 3$ for the random walks of the Σ-protocol of Fig. 1. Like Section 5, we set $\delta = n(p, 1, 2, \lambda)$ for the length of the 2-walks, and $n = n(p, 2^\delta, 3, \lambda)$ for the 3-walks. Lastly, the Σ-protocol needs to be repeated several times to achieve a negligible soundness error. Since one repetition has soundness error $2/3$, the protocol needs to be repeated $-\lambda/\log(2/3)$ times to achieve $2^{-\lambda}$ soundness error. We target the same security levels as the corresponding SIKE parameter sets, i.e., $\lambda = 128$ for `p434` and `p503`, $\lambda = 192$ for `p610`, and $\lambda = 256$ for `p751`. The resulting conservative parameters are summarized in Table 1.

Implementation. We developed an optimized implementation[6] of our proof of knowledge (Section 4.1) for the trusted-setup application (Section 5) based on version 3.5.1 of Microsoft's SIDH library[7]. Our implementation inherits and benefits from all lower-level optimizations contained in that library, and it supports a wide range of platforms with optimized code for a variety of Intel and ARM processors. Compiling our software produces two command-line tools `prove` and `verify`, which use a simple ASCII-based interface to communicate the data contributed to the trusted setup.

The implementation closely follows the strategy outlined in Section 4.2. This includes the choices $d = (2^a)^w$ and $\ell^n = (3^b)^h$; thus, both the witness and the commitment isogenies are uniformly random cyclic isogenies of degree d and ℓ^n respectively. To reduce latency, we additionally exploit parallelism: Recall that the proof of knowledge is repeated many times to achieve a low soundness error; indeed most of the computations are independent between those repetitions and can thus easily be performed at the same time on a multi-core system. This is confirmed by experimental results, where our implementation is observed to parallelize almost perfectly when run on an eight-core processor.

Sampling purely random large-degree isogenies with code from SIDH comes with two caveats: First, the sampling of "small" squares must avoid backtracking between the individual squares being glued to ensure that the composition is cyclic in the end; in both cases this is done by keeping track of the kernel of the dual of the last prime-degree step of the previous square and avoiding points lying above this "forbidden" kernel when choosing the next square. Besides that, the specific isogeny formulas used in SIDH fail for the 2-torsion point $(0, 0)$, which can be resolved by changing to a different Montgomery model each time this kernel point is encountered. For curves revealed in the proof, the choice of Montgomery model should be randomized to avoid leakage. Similarly, the kernel generators of the horizontal isogeny ϕ' also need to be randomized, as Lemma 14 only distinguishes cyclic subgroups and revealing specific generators may leak.

[6] The source code is available at https://github.com/trusted-isogenies/SECUER-pok.
[7] https://github.com/microsoft/PQCrypto-SIDH

Table 2: Benchmarks for instance generation, proving, and verification of our proof of isogeny knowledge for each of the four SIKE finite fields.

	Single-core Time (s)			Eight-core Time (s)		
$\log(p)$	Instance	Prove	Verify	Instance	Prove	Verify
434	0.01	18.15	1.93	0.01	2.96	0.32
503	0.01	25.70	2.71	0.01	4.17	0.44
610	0.02	74.82	7.69	0.02	12.12	1.24
751	0.04	162.47	17.01	0.04	26.07	2.89

Our software sacrifices some performance for simplicity, which aids auditability and hence helps increase trust in the results of a trusted-setup ceremony. Some unused optimizations: Two-isogenies are faster to compute than three-isogenies, and since the SIDH ladder is taller than wider, swapping the role of two- and three-isogenies in the trusted-setup application could somewhat improve the resulting performance. For simplicity, our implementation also only uses full SIDH squares, and thus all isogeny degrees are rounded up to the closest multiple of an SIDH square; shortening the sides of some of the squares can save time. We also did not apply all optimizations to reduce the proof size. This includes applying SIDH-style compression techniques [CJL+17] to the points contained in the proof, cutting their size approximately in half. Moreover, applying a slight bias when sampling the challenges chall_i means smaller responses can appear more often, at the expense of requiring slightly more repetitions; we investigated this tradeoff and determined that the potential improvement is essentially void.

Results. We benchmarked the three algorithms (instance generation, proving, and verification) that make up the zero-knowledge proof of knowledge. We run our tests on an ARM Apple M1 Pro with eight cores, and we averaged the running times of 100 iterations for the parallel implementation and the running times of 50 iterations of the single-core version. The resulting timings are shown in Table 2. They demonstrate that the algorithm is highly practical and can realistically be used within a trusted setup protocol: Generating proofs of knowledge for all four base fields takes less than five core-minutes on a modern CPU. Note that these algorithms need to be run only once per contributor.

Real-world deployment. We briefly discuss how we intend to deploy the trusted setup protocol proposed in Section 5. The goals of such a deployment include include a transparent setup that allows parties to trust the process, a low bar of entry to participate in the protocol, and a secure system that can withstand Sybil and Denial-of-Service (DoS) attacks.

Firstly, we are releasing at https://github.com/trusted-isogenies/ a set of tools that participants can download and run to generate a valid addition to the trusted setup, and for ceremony orchestrators to validate protocol submissions on the server-side. To increase user trust, we also provide higher-level versions (e.g., in SageMath) of some components. Moreover, the proof format is made public, so that any party can—if they choose to—re-implement the algorithms and generate a compatible proof.

Then, we propose leveraging the existing infrastructure of git and GitHub to host our distributed protocol. Thus, each party E_i can generate a random walk from the latest curve E_{i-1} to a new curve E_i, generate a PoK of their secret isogeny walk, and submit the new curve and the PoK to the server as a pull request (PR). The server is a separate git repository and execution environment maintaining the sequence of curves and the proofs, with checks that are run automatically against submissions from parties. The repository automation verifies that the submitted PoK of the isogeny between the current curve E_{i-1} at the end of the walk (the 'tip' curve) and the new proposed curve E_i is valid, and that the PR does not rewrite any previous history. If the checks pass, the PR is rebased on top of the main branch, adding the new PoK of the latest hop, and updating the tip curve to E_i. New parties in the protocol will generate isogeny walks starting from the new tip curve.

If the chain of isogenies diverges, i.e. if some party submits a new curve and PoK starting from a curve other than the tip, the new submission is rejected. This may happen when several parties try to contribute at the same time. To minimize the amount of wasted prover work, we parallelize verification and reject invalid proofs as early as possible.

The configuration for the continuous integration checks is maintained in a separate repository to prevent modification from protocol parties. Hosting the protocol on GitHub raises the bar to Sybil attacks,

as it requires all parties to have a GitHub account with a verified email address. Using our tool requires generation of a GitHub personal access token to authenticate when generating the submission, which further complicates automation / collusion.

The end result of the protocol is a public git repository whose final commit contains a sequence of curves and valid PoKs of isogenies between them, the last of which is the final SECUER E_t, a curve with unknown endomorphism ring, in a parsable hex encoding. Anyone can pull down this artifact and verify the sequence of curves and proofs independently if they wish.

7 Conclusion

In this work, we analyzed a distributed SECUER generation protocol, and proposed a concrete instantiation with strong security guarantees based on a novel proof of isogeny knowledge. To demonstrate the practical feasibility of our protocol, we are going to run a distributed SECUER generation ceremony, scaling to hundreds of participants, using the technology outlined in Section 6.

Our new PoK is especially well-suited for SIDH-like base fields, but can be used reasonably well with fields $\mathbb{F}_{p^2}$ of any characteristic. Generic ZK proof systems, such as the SumCheck protocol used in [CSRT22], would be an alternative to our PoK. After this work was published, Cong, Lai and Levin [CLL23] designed an R1CS encoding of 2-isogeny walks that they fed to various generic proof systems. Their results show that Aurora [BCR+19], in particular, can be quite competitive, giving a measurable speed boost at the cost of a moderate increase in proof size. Currently, the question of which proof system to use appears to be context-dependent.

None of the currently known techniques are particularly well suited for proving knowledge of an isogeny walk over $\mathbb{F}_p$: our PoK and generic proof systems are much more efficient when the walks consist of isogenies of small degree such as 2 or 3, which is not possible over $\mathbb{F}_p$. SeaSign-like techniques [DG19, DPV19] are at least one order of magnitude slower than our PoK, and scale much worse. CSI-FiSh [BKV19] is reasonably efficient, but limited to the base field of CSIDH-512. We think generating SECUERs over $\mathbb{F}_p$ efficiently is an interesting open problem.

To show the security of the proof of knowledge, we developed the theory of supersingular isogeny graphs with level structure, in particular proving that they possess the Ramanujan property. In this work we only focused on the so-called Borel level structure, however similar properties can be proven for more general level structures. In a follow-up work, we will develop the general theory of these graphs, prove bounds on their eigenvalues, and discuss consequences for isogeny-based cryptography.

Acknowledgments. We are grateful to the reviewers and to Shai Levin for helping catch several mistakes and misprints. We thank Jeff Burdges for valuable discussions during the preparation of this work.

References

ABLS07. Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Commun. Contemp. Math.*, 9(4):585–603, 2007. doi:10.1142/S0219199707002551. 11

ADMP20. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64834-3_14. 2

Arp22. Sarah Arpin. Adding level structure to supersingular elliptic curve isogeny graphs, 2022. arXiv:2203.03531. doi:10.48550/ARXIV.2203.03531. 2

Bas23. Andrea Basso. A post-quantum round-optimal oblivious prf from isogenies. Cryptology ePrint Archive, Paper 2023/225, 2023. URL: https://eprint.iacr.org/2023/225. 2

BBD+22. Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E. Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig. Failing to hash into supersingular isogeny graphs. Cryptology ePrint Archive, Report 2022/518, 2022. https://eprint.iacr.org/2022/518. 3

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_12. 17

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, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019. doi:10.1007/978-3-030-17653-2_4. 23

BD21. Jeffrey Burdges and Luca De Feo. Delay encryption. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 302–326. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77870-5_11. 2, 4

BDFLS20. Daniel J Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020. doi:10.2140/obs.2020.4.39. 5

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, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019. doi:10.1007/978-3-030-34578-5_9. 2, 4, 23

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 *LNCS*, pages 520–550. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64834-3_18. 4

CCL15. Ran Canetti, Asaf Cohen, and Yehuda Lindell. A simpler variant of universally composable security for standard multiparty computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 3–22. Springer, Heidelberg, August 2015. doi:10.1007/978-3-662-48000-7_1. 16, 18

CD22. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Report 2022/975, 2022. https://eprint.iacr.org/2022/975. 2, 4

CJL+17. Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 679–706. Springer, Heidelberg, April / May 2017. doi:10.1007/978-3-319-56620-7_24. 22

CLG09. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, January 2009. doi:10.1007/s00145-007-9002-x. 1, 2, 3

CLL23. Kelong Cong, Yi-Fu Lai, and Shai Levin. Efficient isogeny proofs using generic techniques. Cryptology ePrint Archive, Report 2023/037, 2023. https://eprint.iacr.org/2023/037. 23

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 Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-03332-3_15. 1

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291. 2

CPV20. Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. Rational isogenies from irrational endomorphisms. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 523–548. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_18. 3

CSRT22. Jorge Chávez-Saab, Francisco Rodríguez-Henríquez, and Mehdi Tibouchi. Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In Riham AlTawy and Andreas Hülsing, editors, *SAC 2021*, volume 13203 of *LNCS*, pages 441–460. Springer, Heidelberg, September / October 2022. doi:10.1007/978-3-030-99277-4_21. 23

DdF+21. Luca De Feo, Cyprien de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski. Séta: Supersingular encryption from torsion attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 249–278. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92068-5_9. 1, 2

DDGZ22. Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 310–339. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_11. 2, 4, 12, 13

Del74. Pierre Deligne. La conjecture de Weil : I. *Publications Mathématiques de l'IHÉS*, 43:273–307, 1974. URL: http://www.numdam.org/item/PMIHES_1974__43__273_0/. 9

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. doi:10.1515/jmc-2012-0015. 2, 4, 12, 16

DG19. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Heidelberg, May 2019. doi:10.1007/978-3-030-17659-4_26. 2, 4, 12, 23

DKL+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, Part I*, volume 12491 of *LNCS*, pages 64–93. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64837-4_3. 1, 2

DKS18. Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 365–394. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-03332-3_14. 2

DMPS19. Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 248–277. Springer, Heidelberg, December 2019. doi:10.1007/978-3-030-34578-5_10. 1, 2

DPV19. Thomas Decru, Lorenz Panny, and Frederik Vercauteren. Faster SeaSign signatures through improved rejection sampling. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 271–285. Springer, Heidelberg, 2019. doi:10.1007/978-3-030-25510-7_15. 23

dQKL+21. Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion-point attacks on SIDH variants. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 432–470, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84252-9_15. 2

DS05. Fred Diamond and Jerry Shurman. *A first course in modular forms*, volume 228 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005. 9, 10

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, Part III*, volume 10822 of *LNCS*, pages 329–368. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_11. 1, 2

FKMT22. Tako Boris Fouotsa, Péter Kutas, Simon-Philipp Merz, and Yan Bo Ti. On the isogeny problem with torsion point information. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 142–161. Springer, Heidelberg, March 2022. doi:10.1007/978-3-030-97121-2_6. 2

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12. 16

GK21. Eyal Z. Goren and Payman L Kassaei. p-adic dynamics of Hecke operators on modular curves. *Journal de Théorie des Nombres de Bordeaux*, 33(2):387–431, 2021. URL: https://www.jstor.org/stable/48618785. 9

GPS17. Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70694-8_1. 12

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, January 2020. doi:10.1007/s00145-019-09316-0. 1, 2, 12

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, Part I*, volume 10031 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_3. 1, 2

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. 4, 13

HPS89. Hiroaki Hijikata, Arnold K. Pizer, and Thomas R. Shemanske. The basis problem for modular forms on $\Gamma_0(N)$. *Mem. Amer. Math. Soc.*, 82(418):vi+159, 1989. doi:10.1090/memo/0418. 9

Hus04. Dale Husemöller. *Elliptic curves*, volume 111 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2004. 8

JAC+20. 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. SIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. 2

JD11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011. doi:10.1007/978-3-642-25405-5_2. 1

Koh96. David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkley, 1996. URL: https://www.i2m.univ-amu.fr/perso/david.kohel/pub/thesis.pdf. 3

LB20. Jonathan Love and Dan Boneh. Supersingular curves with small noninteger endomorphisms. *Open Book Series*, 4(1):7–22, 2020. doi:10.2140/obs.2020.4.7. 3

LGd21. Yi-Fu Lai, Steven D. Galbraith, and Cyprien de Saint Guilhem. Compact, efficient and UC-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 213–241. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77870-5_8. 2

LP17. David A. Levin and Yuval Peres. *Markov chains and mixing times*. American Mathematical Society, Providence, RI, 2017. doi:10.1090/mbk/107. 12

Mes86. Jean-François Mestre. La méthode des graphes. Exemples et applications. In *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata, 1986)*, Nagoya, 1986. Nagoya University. URL: https://wstein.org/msri06/refs/mestre-method-of-graphs/mestre-fr.pdf. 3

MMP22. Marzio Mula, Nadir Murru, and Federico Pintore. Random sampling of supersingular elliptic curves. Cryptology ePrint Archive, Report 2022/528, 2022. https://eprint.iacr.org/2022/528. 3

MMP+23. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *To appear in EUROCRYPT 2023*, LNCS, Heidelberg, 2023. Springer. URL: https://eprint.iacr.org/2022/1026. 2, 4

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 330–353. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70697-9_12. 2

Piz90. Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bulletin of the American Mathematical Society (N.S.)*, 23(1), 1990. doi:10.1090/S0273-0979-1990-15918-X. 3

Rob22. Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Report 2022/1038, 2022. https://eprint.iacr.org/2022/1038. 2, 4

Sch74. Bruno Schoeneberg. *Elliptic modular functions: an introduction*. Die Grundlehren der mathematischen Wissenschaften, Band 203. Springer-Verlag, New York-Heidelberg, 1974. 9, 10

Ste22. Bruno Sterner. Commitment schemes from supersingular elliptic curve isogeny graphs. *Mathematical Cryptology*, 1(2):40–51, Mar. 2022. URL: https://journals.flvc.org/mathcryptology/article/view/130656. 2

Sto10. Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Advances in Mathematics of Communications*, 4(2):215–235, 2010. doi:10.3934/amc.2010.4.215. 2

Vél71. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, 273:305–347, 1971. 5

Voi21. John Voight. *Quaternion algebras*, volume 288 of *Graduate Texts in Mathematics*. Springer, Cham, 2021. doi:10.1007/978-3-030-56694-4. 10

Wes22a. Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 345–371. Springer, Heidelberg, May / June 2022. doi:10.1007/978-3-031-07082-2_13. 1, 2

Wes22b. Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd FOCS*, pages 1100–1111. IEEE Computer Society Press, February 2022. doi:10.1109/FOCS52979.2021.00109. 1, 3, 20