Threshold Schemes from Isogeny Assumptions

Luca De Feo[1] and Michael Meyer[2,3]*

[1] IBM Research Zürich, Switzerland
[2] University of Applied Sciences Wiesbaden, Germany
[3] University of Würzburg, Germany

Abstract. We initiate the study of threshold schemes based on the Hard Homogeneous Spaces (HHS) framework of Couveignes. Quantum-resistant HHS based on supersingular isogeny graphs have recently become usable thanks to the record class group precomputation performed for the signature scheme CSI-FiSh.
Using the HHS equivalent of the technique of *Shamir's secret sharing in the exponents*, we adapt isogeny based schemes to the threshold setting. In particular we present threshold versions of the CSIDH public key encryption, and the CSI-FiSh signature schemes.
The main highlight is a threshold version of CSI-FiSh which runs almost as fast as the original scheme, for message sizes as low as 1880 B, public key sizes as low as 128 B, and thresholds up to 56; other speed-size-threshold compromises are possible.

Keywords: Threshold cryptography · Hard homogeneous spaces · Isogeny-based cryptography · CSIDH · CSI-FiSh

1 Introduction

Threshold cryptography and secret sharing are large areas of interest in the cryptographic community since the late 1970s, when Shamir [51] and Blakley [7] published the first secret sharing schemes. In 1989, Desmedt and Frankel [21] constructed a practical threshold cryptosystem based on Shamir's secret sharing and ElGamal encryption [26].

The goal of a k-out-of-n, or (k, n)-threshold scheme is to split a secret key into multiple shares and distribute them among n parties, each party receiving one share. Then, for a certain threshold $k \leq n$, any k collaborating parties must be able to compute the cryptographic operation, e.g. decrypt or sign, without learning the secret key, while any set of less than k parties must be unable to do so.

After the publication of Desmedt and Frankel's scheme, several other threshold protocols were proposed; among others, a threshold variant of ElGamal signatures by Harn [34], a threshold DSA scheme by Gennaro *et al.* [32], and Desmedt and Frankel's and Shoup's threshold RSA signature schemes [22,53]. More recently, applications of threshold schemes in the context of blockchains

* Supported by Elektrobit Automotive, Erlangen, Germany.

and cryptocurrencies led to a renewed interest in threshold ECDSA schemes [24,31].

However, all of these schemes are either based on discrete logarithm or integer factorization problems, and are thus not quantum-resistant, since they fall prey to Shor's algorithm [52]. Only very recently, Cozzo and Smart [14] reviewed the post-quantum signature schemes that entered the second round of the NIST PQC standardization process [43] for threshold variants. Their main observation is that only the multivariate-based schemes LUOV [5] and Rainbow [23] allow for a natural threshold construction.

Another popular family of post-quantum schemes is provided by isogeny-based cryptography [36,35]. While this family is not represented in the NIST PQC track for signatures, isogeny-based signatures have recently attracted much attention [16,19,4]. In this work we introduce the first isogeny-based threshold encryption and signature schemes, based on Shamir's secret sharing.

Our schemes are simple adaptations of Desmedt and Frankel's and related schemes to the *Hard Homogeneous Spaces (HHS)* framework. This framework was introduced by Couveignes [13], to generalize both discrete logarithm and isogeny-based schemes. Encryption schemes for HHS were first proposed by Couveignes [13] and Rostovtsev and Stolbunov [49], then improved by De Feo, Kieffer and Smith [17], eventually lead to the development of CSIDH by Castryck, Lange, Martindale, Panny, and Renes [10].

The possibility of signature schemes based on HHS was first suggested by Couveignes [13] and Stolbunov [55,56], although no instantiation was known until recently, when Beullens, Kleinjung, and Vercauteren introduced CSI-FiSh [4]. Before that, an alternative signature scheme based on a weaker notion of HHS, named SeaSign, was presented by De Feo and Galbraith [16].

Our Contributions. We introduce threshold variants of the Couveignes–Rostovtsev–Stolbunov encryption and signature schemes, based on Shamir's secret sharing. To make the results more easily accessible to non-experts, we first present our schemes in an abstract way, using the language of HHS, and only later we analyze their instantiation using CSIDH / CSI-FiSh.

The encryption scheme is a direct adaptation of [21]; its security can only be proven in a *honest-but-curious* security model [9], we thus spend little time on it. The signature scheme is similar to threshold versions of Schnorr signatures [50]; we prove its security in a *static corruptions* model, under a generalization of the Decision Diffie-Hellman Group Action (DDHA) assumption of Stolbunov, however it does not achieve *robustness*.

We conclude with an analysis of the instantiations of the schemes based on isogeny graphs, in particular on the supersingular isogeny graphs used in CSIDH and CSI-FiSh.

We view this work as an initial step towards practical threshold schemes based on HHS and isogenies. Several technical improvements, such as better security properties and proofs, are necessary before these protocols can be considered truly practical. We discuss these issues at the end of this work.

Outline. Section 2 recalls basic facts on secret sharing, threshold cryptography, and HHS. Section 3 then introduces threshold encryption and signature schemes based on HHS, and reviews their security features. In Section 4, we give details about the instantiation of these threshold schemes using isogeny graphs. We conclude by summarizing open problems towards practical applications of our schemes.

2 Preliminaries

We briefly recall here two fundamental constructions in group-theoretic cryptography. The first, Shamir's secret sharing [51], lets a *dealer* split a secret s into n *shares*, so that any k shares are sufficient to reconstruct s; it is a basic primitive upon which several threshold protocols can be built.

The second, Couveignes' *Hard Homogeneous Spaces* (HHS) [13], is a general framework that abstracts some isogeny protocols, and that eventually inspired CSIDH [10]. Although most popular isogeny-based primitives are not, strictly speaking, instances of HHS, the protocols introduced in this work require an instance of an HHS in the strictest sense, and will thus be presented using that formalism.

2.1 Shamir's secret sharing & threshold cryptosystems

Shamir's scheme relies on polynomial interpolation to construct a k-out-of-n threshold secret sharing, for any pair of integers $k \leq n$.

Concretely, a prime $q > n$ is chosen, and the secret s is sampled from $\mathbb{Z}/q\mathbb{Z}$. To break the secret into shares, the dealer samples random coefficients $c_1, \ldots, c_{k-1} \in \mathbb{Z}/q\mathbb{Z}$ and forms the polynomial

$$f(x) = s + \sum_{i=1}^{k-1} c_i x^i;$$

then they form the shares $s_1 = f(1), \ldots, s_n = f(n)$ and distribute them to the n participants, denoted by $\mathcal{P}_1, \ldots, \mathcal{P}_n$. We shall call i the *identifier* of a participant $\mathcal{P}_i$, and s_i his *share*.

Any k participants, but no less, can reconstruct f using Lagrange's interpolation formula, and then recover s by evaluating f at 0. Explicitly, a set of participants $\mathcal{P}_i$, with indices taken from a set $S \subset \{1, \ldots, n\}$ of cardinality at least k, can recover the secret s in a single step through the formula

$$s = f(0) = \sum_{i \in S} f(i) \cdot \prod_{\substack{j \in S \\ j \neq i}} \frac{j}{j - i}.$$

Shamir's secret sharing enjoys *perfect* or *information theoretic* security, meaning that less than k shares provide no information on the secret. Indeed, assuming that $k-1$ participants, w.l.o.g. $\mathcal{P}_1, \ldots, \mathcal{P}_{k-1}$, put their shares together, the map

$$(s, c_1, \ldots, c_{k-1}) \mapsto \big(f(0), f(1), \ldots, f(k-1)\big)$$

is, by Lagrange's formula, an isomorphism of $(\mathbb{Z}/q\mathbb{Z})$-vector spaces; hence, each tuple $\big(s = f(0), f(1), \ldots, f(k-1)\big)$ is equally likely to occur.

Threshold schemes. A major step towards practical threshold schemes based on Shamir's secret sharing was Desmedt and Frankel's threshold variant of ElGamal decryption [21]; a similar approach to design threshold signatures was proposed by Harn [34]. Many other threshold protocols follow a similar pattern, colloquially referred to as *secret sharing in the exponents*, that we are now going to briefly recall.

Let the secret $s \in \mathbb{Z}/q\mathbb{Z}$ and the shares s_i be distributed as above. Let G be a cyclic group of order q, and let g be a generator. Assuming that discrete logarithms are hard in G, the participants' goal is to compute the *shared key* g^s without letting anyone learn the secret s. We can again use Lagrange interpolation, but this time in the exponent:

$$g^s = g^{\sum s_i \prod \frac{j}{j-i}}.$$

To make this idea into a protocol, each party computes g^{s_i} from its share s_i, and sends it to all other parties. Given k shares s_i of the key with $i \in S$ and $\#S \geq k$, any party can then compute the shared key as

$$g^s = \prod_{i \in S} (g^{s_i})^{L^S_{0,i}},$$

where the exponents

$$L^S_{l,i} = \prod_{\substack{j \in S \\ j \neq i}} \frac{j-l}{j-i} \mod q \tag{1}$$

can be precomputed from public information.

If broadcasting the shares g^{s_i} to all participants is too expensive, an alternative is to send them to a central *combiner*, who is then in charge of computing g^s and finalizing the protocol. As we shall see later, this flexibility will be lost in our setting.

Secret sharing in rings. The proof of perfect security of Shamir's secret sharing scheme fundamentally relies on $\mathbb{Z}/q\mathbb{Z}$ being a field. For reasons that will become apparent later, we shall need to adapt the scheme to non-prime q, and thus to general rings of modular integers. This presents two problems: ensuring that no impossible inversions happen when computing the coefficients $L^S_{l,i}$ in Eq. (1), and proving security in the more general setting. These obstacles are not difficult to overcome, as already highlighted in, e.g., RSA-based threshold schemes [53]; we briefly explain how this is done.

Impossible inversions arise during the reconstruction of the shared secret whenever one of the denominators $(j - i)$ in Lagrange's formula is not coprime to q. If q_1 is the smallest prime factor of q, then there can be at most q_1 distinct values modulo q_1; however, any identifier i congruent to 0 modulo q_1 must be prescribed, since otherwise $f(i) \bmod q_1$ would leak information on $s \bmod q_1$. Hence, at most $q_1 - 1$ participants can take part to Shamir's scheme in $\mathbb{Z}/q\mathbb{Z}$; for example, using $1, 2, \ldots, q_1 - 1$ as identifiers ensures that no difference of two of them shares a common factor with q.

Perfect security of the scheme is also achieved by restricting the identifiers to $1, 2, \ldots, q_1 - 1$, or any other set of integers distinct and non-zero modulo all divisors of q, thus restricting the number of participants to $n < q_1$. We formally prove this below.

Proposition 1. *Let q be an integer with prime factorization $q = \prod q_i^{e_i}$. Assume q_1 is the smallest of the prime factors, let $k \leq n < q_1$, and sample $s, c_1, \ldots, c_{k-1} \in \mathbb{Z}/q\mathbb{Z}$ uniformly at random. Let*

$$f(x) = s + \sum_{i=1}^{k-1} c_i x^i$$

and let $x_1, \ldots x_{k-1} \in \mathbb{Z}/q\mathbb{Z}$ be distinct and non-zero modulo all q_i. Associate a random variable S to s, and random variables Y_i to each $f(x_i)$.

The random variables $S, Y_1, \ldots Y_{k-1}$ are independent; in particular Shamir's (k, n)-secret sharing scheme over $\mathbb{Z}/q\mathbb{Z}$ is perfectly secure, in the sense that, given the shares $f(x_1), \ldots, f(x_{k-1})$, every secret s is equally likely to have originated them.

Proof. Consider the map

$$\rho \,:\, (s, c_1, \ldots, c_{k-1}) \mapsto \big(f(0), f(x_1), \ldots, f(x_{k-1})\big);$$

since all $x_i \mod q_j$ are distinct and non-zero, its reduction modulo q_j is an isomorphism of $\mathbb{Z}/q_j\mathbb{Z}$-vector spaces; thus, by the Chinese Remainder Theorem, ρ is an isomorphism of $\mathbb{Z}/q\mathbb{Z}$-modules.

Introducing random variables Y_0 for $f(0)$ and C_i for the c_i's, we have that

$$\begin{aligned} P\{Y_0 = f(0), Y_1 = f(x_1), \ldots, Y_{k-1} = f(x_{k-1})\} \\ = P\{S = s, C_1 = c_1, \ldots, C_{k-1} = c_{k-1}\} = q^{-k}, \end{aligned}$$

from which we deduce that $P\{Y_i = f(x_i)\} = q^{-1}$. In particular, since $s = f(0)$,

$$\begin{aligned} P\{S = s, Y_1 = f(x_1), \ldots, Y_{k-1} = f(x_{k-1})\} \\ = P\{S = s\} \cdot P\{Y_1 = f(x_1)\} \cdots P\{Y_{k-1} = f(x_{k-1})\} \end{aligned}$$

for any $s, f(x_1), \ldots, f(x_{k-1})$, implying that S and the Y_i's are independent. □

2.2 Hard homogeneous spaces

Hard Homogeneous Spaces (HHS) were introduced by Couveignes in [13] as a generalization of Diffie-Hellman schemes. A *principal homogeneous space*, or $\mathcal{G}$-*torsor* is a set $\mathcal{E}$ endowed with a faithful and transitive group action by a group $\mathcal{G}$.[4] In other words, it is defined by a mapping

$$\mathcal{G} \times \mathcal{E} \to \mathcal{E},$$
$$\mathfrak{g} * E = E',$$

satisfying the following properties:

- *Compatibility:* $\mathfrak{g}' * (\mathfrak{g} * E) = (\mathfrak{g}'\mathfrak{g}) * E$ for any $\mathfrak{g}, \mathfrak{g}' \in \mathcal{G}$ and $E \in \mathcal{E}$;
- *Identity:* $\mathfrak{e} * E = E$ if and only if $\mathfrak{e} \in \mathcal{G}$ is the identity element;
- *Transitivity:* for any $E, E' \in \mathcal{E}$ there exists a unique $\mathfrak{g} \in \mathcal{G}$ such that $\mathfrak{g} * E = E'$;

In particular, if $\mathcal{G}$ is finite, these axioms imply that $\#\mathcal{G} = \#\mathcal{E}$.

Couveignes defines a HHS as a finite principal homogeneous space with some additional algorithmic properties. He requires that the following problems can be solved efficiently (e.g., in polynomial time):

- *Group operations:* decide whether a string $\mathfrak{g}$ represents an element of $\mathcal{G}$, decide whether $\mathfrak{g} = \mathfrak{g}'$, compute $\mathfrak{g}^{-1}$ and $\mathfrak{g}\mathfrak{g}'$;
- *Sampling:* sample uniformly random elements from $\mathcal{G}$;
- *Membership:* decide whether a string E represents an element of $\mathcal{E}$, decide whether $E = E'$;
- *Action:* Given $\mathfrak{g}$ and E, compute $\mathfrak{g} * E$.

Furthermore, the following problems should be hard (e.g., not known to be solvable in polynomial time):

- *Vectorization:* Given $E, E' \in \mathcal{E}$, find $\mathfrak{g} \in \mathcal{G}$ such that $\mathfrak{g} * E = E'$;
- *Parallelization:* Given $E, E', F \in \mathcal{E}$, such that $E' = \mathfrak{g} * E$, find $F' = \mathfrak{g} * F$.

As a simple example, let $\mathcal{E}$ be a group of prime order q, then $\mathcal{G} = (\mathbb{Z}/q\mathbb{Z})^\times$ acts on $\mathcal{E}\backslash\{1\}$ by $a * g = g^a$. In this case, the Vectorization problem is the discrete logarithm problem in $\mathcal{E}$, and the Parallelization problem is the Computational Diffie–Hellman problem. Hence any discrete logarithm group is also a HHS.

Couveignes' original proposal used as HHS sets of ordinary elliptic curves over finite fields, with complex multiplication by a quadratic imaginary order $\mathcal{O}$; indeed, these are torsors for the class group $\mathrm{cl}(\mathcal{O})$, and the Vectorization and Parallelization problems are not known to be easily solvable. Based on this HHS, he defined key exchange as a straightforward generalization of the Diffie–Hellman protocol, and he also sketched an interactive identification scheme.

[4] The reader will excuse our extravagant notation for set and group elements: our goal is to be consistent with the notation used in Section 4 for isogeny-based HHS.

However, Couveignes' proposal presents several difficulties, as neither the group action nor random sampling are known to be easily computable. Independently from Couveignes, Rostovtsev and Stolbunov [49,55] proposed a key-exchange scheme based on the same group action, but with a different representation of elements of $\mathrm{cl}(\mathcal{O})$. This proposal had the benefit of making key-exchange feasible, if not practical, and subsequent research [17] eventually led to the development of CSIDH [10], an efficient key exchange scheme based on the action of a quadratic class group on a set of supersingular curves.

Nevertheless, none of these constructions satisfies exactly the axioms of a HHS, since, for example, the cost of evaluating $\mathfrak{g} * E$ in CSIDH is in the worst case exponential in the size of $\mathfrak{g}$. While every group element has an equivalent representation that permits to efficiently evaluate the action, computing such representation is difficult in general. This is not a problem for key-exchange schemes based on CSIDH, but, for example, it makes identification and signature schemes more involved and less efficient than what Couveignes had originally envisioned [16,19].

The roadblock in all these constructions is the fact that the structure of the class group $\mathrm{cl}(\mathcal{O})$ is unknown, and it is thus impossible to have a unique representation for its elements. The best algorithm for computing the class group structure runs in sub-exponential time, and is thus neither practical nor scalable; nevertheless the application to isogeny-based signatures motivated Beullens, Kleinjung and Vercauteren [4] to run an intensive computation for the CSIDH-512 parameter set, which allowed them to construct CSI-FiSh, the most efficient isogeny-based signature to date.

Currently, CSI-FiSh is the only known instance of HHS based on isogenies: group elements have unique representation, the group action can be evaluated efficiently, and the Vectorization and Parallelization problems are believed to be hard, both classically and quantumly. Unfortunately, parameter generation requires exponential time in the security parameter, thus CSI-FiSh is a HHS only in a practical sense for a specific security level, but not in the asymptotic sense.

In the next sections we are going to introduce threshold schemes based on HHS; then we will give more details on CSI-FiSh, and look at how the threshold schemes can be instantiated with it.

3 Threshold schemes from HHS

We now present threshold schemes based on Hard Homogeneous Spaces.

Let a group $\mathcal{G}$ and a set $\mathcal{E}$ be given, such that $\mathcal{G}$ acts faithfully and transitively on $\mathcal{E}$ and the HHS axioms are satisfied. We are going to require an additional property: that an element $\mathfrak{g} \in \mathcal{G}$ of order q is known, and we shall write q_1 for the smallest prime divisor of q. In particular, these hypotheses imply that there is an efficiently computable embedding $\mathbb{Z}/q\mathbb{Z} \hookrightarrow \mathcal{G}$ defined by $a \mapsto \mathfrak{g}^a$, which we are going to exploit to embed Shamir's secret sharing in the HHS.

Notation. From now on we will use capital letters $E, F, \ldots$ to denote elements of the HHS $\mathcal{E}$, and gothic letters $\mathfrak{a}, \mathfrak{b}, \mathfrak{g}, \ldots$ to denote elements of the group $\mathcal{G}$. Following [4], it will be convenient to see $\mathbb{Z}/q\mathbb{Z}$ as acting directly on $\mathcal{E}$: we will write $[a]$ for $\mathfrak{g}^a$, and $[a]E$ for $\mathfrak{g}^a * E$, where $\mathfrak{g}$ is the distinguished element of order q in $\mathcal{G}$.[5] Be wary that under this notation $[a][b]E = [a+b]E$.

Remark 1. The additional hypothesis excludes, in particular, HHS of unknown order, such as CSIDH (outside of the parameter set shared with CSI-FiSh).

Note that, assuming the factorization of q is known, given any element of $\mathcal{G}$ it is easy to test whether it is of order q. Nevertheless, in some instances it may be difficult to decide whether an element $\mathfrak{g}' \in \mathcal{G}$ belongs to $\langle \mathfrak{g} \rangle$; this may happen, for example, if $\mathcal{G} \simeq (\mathbb{Z}/q\mathbb{Z})^2$. This will not impact the protocols we define here, but is an important property to consider when designing threshold protocols in the general HHS setting. At any rate, for instantiations based on CSI-FiSh it is always easy to test membership of $\langle \mathfrak{g} \rangle$.

On the other hand, unless $\mathcal{G} = \langle \mathfrak{g} \rangle$, it is a well known hard problem (exponential in $\log q$) to decide whether given $E, E' \in \mathcal{E}$ there exists $a \in \mathbb{Z}/q\mathbb{Z}$ such that $E' = [a]E$. Indeed, a generic solution to this problem would imply an efficient generic algorithm for solving many instances of discrete logarithms [10].

We now describe a distributed algorithm to compute the group action of $\langle \mathfrak{g} \rangle$ on $\mathcal{E}$ in a threshold manner, and explain how it impacts the communication structure of threshold protocols. Then we present two simple threshold protocols, a KEM and a signature, directly adapted from their non-threshold counterparts.

3.1 Threshold group action

Like in Section 2, we assume that the participants $\mathcal{P}_1, \mathcal{P}_2, \ldots$ possess shares $s_i = f(i)$ of a secret $s \in \mathbb{Z}/q\mathbb{Z}$; their goal is to evaluate the group action $[s]E_0$ for any given $E_0 \in \mathcal{E}$, without communicating their shares s_i.

Let $S \subset \{1, \ldots, n\}$ be a set of cardinality at least k, and recall the definition of the Lagrange coefficients in Eq. (1):

$$L_{l,i}^S = \prod_{\substack{j \in S \\ j \neq i}} \frac{j-l}{j-i} \mod q.$$

Then the participants $\mathcal{P}_i$ for $i \in S$ determine the shared secret by $s = \sum_{i \in S} s_i \cdot L_{0,i}^S$. For the sake of simplicity, we will assume that $S = \{1, ..., k\}$.

The participants coordinate as follows. First, E_0 is sent to $\mathcal{P}_1$, who starts by computing

$$E_1 = \left[s_1 \cdot L_{0,1}^S\right] E_0.$$

The resulting E_1 is passed on to $\mathcal{P}_2$, who continues by computing

$$E_2 = \left[s_2 \cdot L_{0,2}^S\right] E_1 = \left[s_2 \cdot L_{0,2}^S + s_1 \cdot L_{0,1}^S\right] E_0.$$

[5] Note that this action is only transitive if $\mathfrak{g}$ generates $\mathcal{G}$.

This procedure repeats analogously for the parties $\mathcal{P}_3, ..., \mathcal{P}_{k-1}$, and at last $\mathcal{P}_k$ can compute

$$E_k = \left[s_k \cdot L^S_{0,k}\right] E_{k-1} = \left[\sum_{i \in S} s_i \cdot L^S_{0,i}\right] E_0 = [s]E_0.$$

Communication structure. Comparing the algorithm to classical threshold Diffie-Hellman protocols as in Section 2.1, it is obvious that there are differences in their structures. There, each party $\mathcal{P}_i$ computes $g_i = g^{s_i}$ from its secret share s_i and a common generator g. Anyone can then compute $g_i^{L^S_{0,i}}$ for each $i \in S$, and multiply the results to obtain g^s.

In our HHS setting, the situation is different. First, $\left[s_i \cdot L^S_{0,i}\right] E$ cannot be computed from the knowledge of $[s_i]E$ and $L^S_{0,i}$, thus only $\mathcal{P}_i$ can compute it. Consequently, each participant has to know in advance the set S of parties taking part to the computation, in order to apply $L^S_{0,i}$.

Further, it is not possible to introduce a combiner, who could proceed as in the classical case by receiving the different $\left[s_i \cdot L^S_{0,i}\right] E_0$ and combining them to obtain $[s]E_0$, since in general the set $\mathcal{E}$ is not equipped with a compatible group operation $\mathcal{E} \times \mathcal{E} \to \mathcal{E}$. Therefore, it is necessary to adopt a sequential round-robin communication structure:

$$\xrightarrow{E_0,S} \mathcal{P}_1 \xrightarrow{E_1,S} \mathcal{P}_2 \xrightarrow{E_2,S} ... \xrightarrow{E_{k-1},S} \mathcal{P}_k \xrightarrow{[s]E_0} .$$

Note that the order of the $\mathcal{P}_i$ can be changed without affecting the final result.

However, this means that $\mathcal{P}_k$ is the only party who ends up knowing the result of the group action. If a cryptographic protocol needs to handle this element secretly, our algorithm is only suitable for situations where only one participant is required to know the secret result. Algorithm 1 summarizes the described approach in the general case.

Algorithm 1: Threshold variant of the group action computation.

Input : $E_0 \in \mathcal{E}$, set of participants S.
Output: $[s]E_0$.

Set $E \leftarrow E_0$.
foreach $i \in S$ **do**
    If $E \notin \mathcal{E}$, participant $\mathcal{P}_i$ outputs $\perp$ and the algorithm stops.
    Participant $\mathcal{P}_i$ outputs $E \leftarrow \left[s_i \cdot L^S_{0,i}\right] E$.
return E.

In a different setting where all participants are required to secretly know the final result, several modifications are possible. For example, when encrypted channels between the participants exist, the last participant can simply distribute through them the resulting $[s]E_0$.

Alternatively, k parallel executions of Algorithm 1, each arranging the participants in a different order, let all participants know the final result. The cost of this modification is rather high: $O(k^2)$ elements of $\mathcal{E}$ need to be transmitted, and $O(k^2)$ group actions evaluated. This can be improved to $O(k \log k)$ transmitted elements of $\mathcal{E}$ (but still $O(k^2)$ group actions) using a binary splitting strategy.

Remark 2. Algorithm 1 does nothing to prevent corrupted participants from leading to an incorrect output. While threshold schemes based on discrete logarithms can often detect and correct malicious behavior (using, e.g., error correcting codes [32]), this is more difficult for HHS. Indeed, there seems to be no way for a participant to verify the previous participant's output in Algorithm 1, outside of generic zero-knowledge techniques.

3.2 Threshold HHS ElGamal decryption

The first application we present for our threshold group action is *threshold decryption*, a direct adaptation of [21].

Inspired by the classical ElGamal encryption scheme [26], a PKE protocol in the HHS settings was first introduced by Stolbunov [49,55,56]. We briefly recall it here, using the terminology of KEMs.

Public parameters: A HHS $(\mathcal{E}, \mathcal{G})$, a *starting element* $E_0 \in \mathcal{E}$, and a hash function H from $\mathcal{E}$ to $\{0,1\}^\lambda$.
Keygen: Sample a secret key $\mathfrak{a} \in \mathcal{G}$, output $\mathfrak{a}$ and the public key $E_a = \mathfrak{a} * E_0$.
Encaps: Sample $\mathfrak{b} \in \mathcal{G}$, output $K = H(\mathfrak{b} * E_a)$ and $E_b = \mathfrak{b} * E_0$.
Decaps: Given E_b, if $E_b \in \mathcal{E}$ output $K = H(\mathfrak{a} * E_b)$, otherwise output $\perp$.

The **Decaps** routine is easily adapted into a threshold algorithm requiring k participants to collaborate in order to recover the decryption key K. This also requires modifying **Keygen**, which must now be executed by a trusted dealer and integrate Shamir's secret sharing.

Public parameters: A HHS $(\mathcal{E}, \mathcal{G})$ with a distinguished element $\mathfrak{g} \in \mathcal{G}$ of order q, a *starting element* $E_0 \in \mathcal{E}$, and a hash function H from $\mathcal{E}$ to $\{0,1\}^\lambda$.
Keygen:
- Sample a secret $s \in \mathbb{Z}/q\mathbb{Z}$ and generate shares $s_i \in \mathbb{Z}/q\mathbb{Z}$ using Shamir's secret sharing;
- Distribute privately s_i to participant $\mathcal{P}_i$;
- Output public key $E_a = [s]E_0$.

Encaps: Sample $\mathfrak{b} \in \mathcal{G}$, output $K = H(\mathfrak{b} * E_a)$ and $E_b = \mathfrak{b} * E_0$.
Decaps: Given E_b and a set S of participants, $\#S \geq k$, run Algorithm 1 to compute $E = [s]E_b$; output $\perp$ if the algorithm returns $\perp$, otherwise output $K = H(E)$.

The asymmetry of the scheme will not be lost on the reader: while the shared secret for the threshold group is restricted to be in $\langle \mathfrak{g} \rangle$, there are no restrictions for **Encaps**. Although it would be completely possible (maybe even desirable

for practical reasons) to restrict secrets to $\langle \mathfrak{g} \rangle$ also in the encapsulation, we do not do so because there is no known way for decapsulation to test whether E_b has been generated this way.

It is clear that this scheme achieves the stated goal of threshold decryption: upon receiving a ciphertext, at least k participants must agree to decrypt in order to recover the key K; only the last participant in the chain learns K. If less than k participants agree to decrypt, the key K cannot be recovered; however this security property is only guaranteed when all participants behave honestly.

When allowing for corruptions, the scheme immediately becomes broken. Indeed in Algorithm 1, when a participant beyond the first receives an input, they are unable to link it to the ciphertext E_b. This makes it possible to trick an unwilling participant $\mathcal{P}$ into helping decrypt a message: let c be such a message, a group of $k-1$ participants only has to wait for a message c' that $\mathcal{P}$ is willing to decrypt; when $\mathcal{P}$ agrees, they submit to it an intermediate value of a computation for c, which $\mathcal{P}$ is unable to distinguish from one for c'. Contrast this to the original El Gamal threshold decryption of Desmedt and Frankel [21], where each participant performs its computation directly on the input.

Because of this, the security of the protocol can only be proven in a *honest-but-curious* model. We skip the easy security proof, and leave the search for more refined threshold decryption protocols for future work.

3.3 Threshold signatures

An identification scheme in the HHS framework was first sketched by Couveignes [13]; in his PhD thesis [56] Stolbunov also suggested applying the Fiat-Shamir transform [29] to it to obtain a signature scheme. Nevertheless these schemes stood out of reach until recently, when the class group computation for CSIDH-512 was completed [4]; CSI-FiSh is effectively Stolbunov's scheme, combined with optimizations introduced in SeaSign [16].

CSI-FiSh and its ancestors can be easily adapted into threshold protocols. We start by recalling the basic interactive zero-knowledge identification scheme: a prover Peggy wants to convince a verifier Vic that she knows a secret element $\mathfrak{a} \in \mathcal{G}$ such that $E_a = \mathfrak{a} * E_0$. They proceed as follows:

- Peggy samples a random $\mathfrak{b} \in \mathcal{G}$ and commits to $E_b = \mathfrak{b} * E_0$.
- Vic challenges with a random bit $c \in \{0, 1\}$.
- If $c = 0$, Peggy replies with $\mathfrak{z} = \mathfrak{b}$; otherwise she replies with $\mathfrak{z} = \mathfrak{b}\mathfrak{a}^{-1}$.
- If $c = 0$, Vic verifies that $\mathfrak{z} * E_0 = E_b$; otherwise, he verifies that $\mathfrak{z} * E_a = E_b$.

It is immediately seen that the scheme is correct, thanks to the properties of homogeneous spaces, and that it has soundness $1/2$. For the zero-knowledge property, it is crucial that elements in $\mathcal{G}$ can be sampled uniformly, and that they have unique representation. See [56,16,4] for detailed proofs.

We now adapt this scheme into a threshold signature by applying the Fiat-Shamir transform and Shamir's secret sharing as before.

We let again $(\mathcal{E}, \mathcal{G})$ be a HHS with a distinguished element $\mathfrak{g}$ of order q, we fix a starting element $E_0 \in \mathcal{E}$, and a hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$. We assume that a trusted dealer has sampled a random secret $s \in \mathbb{Z}/q\mathbb{Z}$, securely distributed shares s_i to the participants $\mathcal{P}_i$, and published the public key $E_s = [s]E_0$.

Here is a sketch of how a participants $\mathcal{P}_1, \ldots, \mathcal{P}_k$ can cooperate to sign a message m:

- In the commitment phase, the participants collaborate to produce a random element $[b]E_0$ in a way similar to Algorithm 1, by producing each a random value $b_i \in \mathbb{Z}/q\mathbb{Z}$ and evaluating $E_i = [b_i]E_{i-1}$.
- Once $E_k = [b]E_0$ is computed, the challenge bit c is obtained from the hash $H(E_k, m)$.
- If $c = 0$, each $\mathcal{P}_i$ outputs $z_i = b_i$, else each $\mathcal{P}_i$ outputs $z_i = b_i - s_i \cdot L_{0,i}^S$.
- The signature is $(c, z = \sum z_i)$.

To verify the signature it suffices to check that $H([z]E_0, m) = 0 \ldots$, if $c = 0$, or that $H([z]E_s, m) = 1 \ldots$, if $c = 1$. Of course, this sketch must be repeated λ times, in order to ensure the appropriate level of security.

The complete signing algorithm is summarized in Algorithm 2. As presented there, it is rather inefficient in terms of signature size and signing/verification time. All the key/signature size compromises presented in CSI-FiSh [4] are compatible with our threshold adaptation, and would produce a more efficient signature scheme. The details are left to the reader.

Security analysis. We conclude with a study of the security of the threshold signature scheme. We define the following existential forgery game (UF-CMA) with *static corruptions*:

- The trusted dealer draws the secret s, distributes the shares s_i to the participants, and publishes the public key $[s]E_0$.
- The adversary chooses $k-1$ participants to corrupt, and learns their shares s_i.
- After that, the adversary can query a threshold signing oracle, which receives a set S of at least k participants and a message m, and outputs a signature on m by the set S, and all the messages sent by the participants in S during the execution of the protocol.
- The adversary wins if they produce a valid signature for a message m that was not submitted to the oracle.

We will prove the security of the signature scheme under a new assumption, that we call *Power-DDHA*. This decision version of the *Scalar-HHS* problem of Felderhoff [28] is a generalization of the Decision Diffie–Hellman Group Action (DDHA) introduced by Stolbunov [55], and is related to the P-DDH assumption introduced by Kiltz for discrete logarithm groups [38].

Problem 1 (Power-DDHA problem). Let $(\mathcal{E}, \mathcal{G})$ be a HHS. Let $E \in \mathcal{E}$ and $1 < a < \#\mathcal{G}$ an integer; let $\mathfrak{s}$ be a uniformly random element in $\mathcal{G}$. The *a-Power-DDHA problem* is: given $(a, E, \mathfrak{s} * E, F)$, where $F \in \mathcal{E}$ is an element, either

Algorithm 2: Threshold HHS signature.

Input : Message m, participant set S.
Output: A signature on m.

Set $(E_1^0, \ldots, E_\lambda^0) \leftarrow (E_0, \ldots, E_0)$.
Let $k \leftarrow 0$.
foreach $i \in S$ **do**
  Let $k \leftarrow k + 1$.
  foreach $1 \leq j \leq \lambda$ **do**
    If $E_j \notin \mathcal{E}$, participant $\mathcal{P}_i$ outputs $\perp$ and aborts the protocol.
    $\mathcal{P}_i$ samples $b_{i,j} \in \mathbb{Z}/q\mathbb{Z}$ uniformly at random.
    $\mathcal{P}_i$ outputs $E_j^k \leftarrow [b_{i,j}]E_j^{k-1}$.
Let $c_1 \cdots c_\lambda \leftarrow H(E_1^k, \ldots, E_\lambda^k, m)$.
foreach $i \in S$ **do**
  foreach $1 \leq j \leq \lambda$ **do**
    if $c_j = 0$ **then**
      $\mathcal{P}_i$ outputs $z_{i,j} = b_{i,j}$.
    else
      $\mathcal{P}_i$ outputs $z_{i,j} = b_{i,j} - s_i \cdot L_{0,i}^S$.
foreach $1 \leq j \leq \lambda$ **do**
  Let $z_j = \sum_{i \in S} z_{i,j}$.
return the signature $(c_1 \cdots c_\lambda, z_1, \ldots, z_\lambda)$.

sampled from the uniform distribution on $\mathcal{E}$, or $F = \mathfrak{s}^a * E$, decide from which distribution F is drawn.

Remark 3. The special case of (-1)-Power-DDHA where the HHS is instantiated with a graph of $\mathbb{F}_p$-isomorphism classes of supersingular curves, and E is the special curve $E : y^2 = x^3 + x$, is known to be solvable efficiently. Other "special" curves in the graph also enjoy this property, see [11].

This obstacle is easy, but tedious, to circumvent in the proof of the next theorem. We leave the details to the reader.

Felderhoff proved that the search version of Power-DDHA (Scalar-HHS) is equivalent to Parallelization whenever the order of $\mathcal{G}$ is known and odd [28]. We also recall the formal definition of the Vectorization problem, also known as *Group Action Inverse Problem* [55].

Problem 2 (GAIP). Let $(\mathcal{E}, \mathcal{G})$ be a HHS, let E, F be uniformly random elements of $\mathcal{E}$. The *Group Action Inverse Problem* asks to compute $\mathfrak{a} \in \mathcal{G}$ such that $E = \mathfrak{a} * F$.

It is clear that GAIP is harder than Power-DDHA: given a GAIP solver one can simply apply it to $(E, \mathfrak{s} * E)$, and then use the answer to solve Power-DDHA.

Theorem 1. *Under the Power-DDHA assumption, the signature scheme of Algorithm 2 is UF-CMA secure, when the hash function H is modeled as a random oracle.*

Proof. We will reduce the security of the threshold scheme to that of the original non-threshold scheme of Stolbunov, which is known to be secure under the GAIP assumption in the ROM [56,4]. Because the reduction relies on Power-DDHA, which is easier than GAIP, the statement follows. The reduction consists of a sequence of games, the first one being the original UF-CMA game. We only sketch the proof.

In the second game we replace the threshold signing oracle as follows. When the adversary asks for the shares of the $k-1$ corrupted participants, we sample $s_i \in \mathbb{Z}/q\mathbb{Z}$ at random and send them back. We also sample the shared secret $s \in \mathbb{Z}/q\mathbb{Z}$, which uniquely determines a polynomial f of degree $k-1$ such that $f(i) = s_i$. When the adversary asks for a signature, we simply use the signing algorithm 2. This game is thus indistinguishable from the original one for the adversary.

The third game is identical to the second, however we change the way the replies to the adversary are computed.

Along with the signature $(c_1 \cdots c_\lambda, z_1, \ldots, z_\lambda)$, we compute the sequence

$$\begin{aligned} E_s^0 &= E_0, \\ E_s^{k_i} &= \left[s_i \cdot L_{0,i}^S\right] E_s^{k_i - 1}, \end{aligned}$$

for $i \in S$, where k_i is the position of i in S, so that $E_s = E_s^k$.

Then, for each $1 \leq j \leq \lambda$, the participant messages $z_{i,j}$ are sent to the adversary normally. The messages $E_j^{k_i}$ are recomputed as follows: if $c_j = 0$ we set $E_j^{k_i} = [b_{k_1,j} + b_{k_2,j} + \cdots + b_{k_i,j}]E_0$, as in Algorithm 2. If $c_j = 1$ we set $E_j^{k_i} = [z_{k_1,j} + z_{k_2,j} + \cdots + z_{k_i,j}]E_s^{k_i}$, which is immediately seen as being the same as in Algorithm 2, thanks to $b_{i,j} = z_{i,j} + s_i \cdot L_{0,i}^S$. An example of this computation with three participants is pictured in Figure 1.

Since the values sent to the adversary are the same as in the previous game, this game is still indistinguishable from the original one.

In the fourth game we generate the signature as in the previous game, however we avoid using the knowledge of the uncorrupted shares to compute the elements $E_s^{k_i}$. For simplicity, we will assume that querying sets S are always sorted in increasing order, so that the relative order of the participants' actions does not change between queries.

If S contains only one uncorrupted participant $\mathcal{P}_i$, it is easy to perfectly simulate the messages: assume $\mathcal{P}_i$ is in position k_i in S, for any $k' < k_i$ we can compute $E_s^{k'}$ *directly*:

$$E_s^{k'} = \left[\sum_{i \in S, k_i \leq k'} s_i \cdot L_{0,i}^S\right] E_0,$$

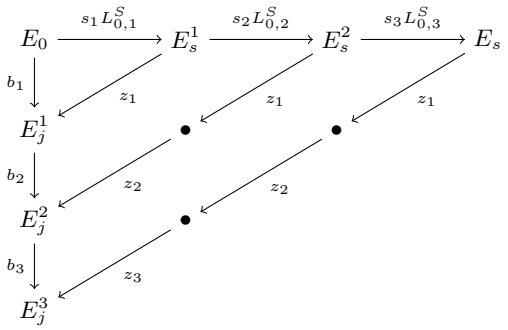

Fig. 1. Recomputation of $E_j^{k_i}$ given $z_{i,j}$.

whereas for all $k' \geq k_i$ we can compute it *backwards*:

$$E_s^{k'} = \left[\sum_{i \in S, k_i > k'} -s_i \cdot L^S_{0,i} \right] E_s.$$

When S contains more than one uncorrupted participant, we will resort to random sampling. Like above, we start one *direct* chain from E_0, and one *backwards* from E_s; both chains stop when they encounter an uncorrupted participant $\mathcal{P}_i$. Now, let $E_s^{k_i - 1}$ be the last curve in the *direct* chain, we set the next curve $E_s^{k_i} = [r_i]E_0$, where r_i is sampled uniformly from $\mathbb{Z}/q\mathbb{Z}$. We also store r_i in association with S, and keep it for reuse the next time the adversary queries for the set S.

We continue the *direct* chain from $E_s^{k_i}$, either using the knowledge of $s_i \cdot L^S_{0,i}$ for corrupted participants, or sampling a random r_i for uncorrupted ones; we stop when we meet the *backwards* chain. An example of this process is pictured below:

$$E_0 \xrightarrow{\mathbf{r_1}} \mathbf{E_s^1} \xrightarrow{s_2 L^S_{0,2}} E_s^2 \xrightarrow{\mathbf{r_3}} \mathbf{E_s^3} \xrightarrow{r_4} E_s^4 \xrightarrow{s_4 L^S_{0,4}} E_s$$

we write in bold data that is obtained through random sampling; the value r_4 is implicitly determined by the other four values.

After we have determined this data, we answer all signing queries for S as in the previous game (see Figure 1). Now, the adversary's view is no longer indistinguishable from the original game, however we argue that it still is computationally indistinguishable assuming Power-DDHA. Indeed, when $c_j = 1$, the adversary is able to recover $E_s^{k_i}$ from $E_j^{k_i}$ as $E_s^{k_i} = [-z_{k_1,j} - z_{k_2,j} - \cdots - z_{k_i,j}]E_j^{k_i}$. This means that the adversary will collect many pairs of the form $\left(E, \left[s_i \cdot L^S_{0,i}\right]E\right)$ (when $\mathcal{P}_i$ is the only uncorrupted participant in S), and many others of the form $(E', [r_i]E')$ (where the expected relation would be $\left(E', \left[s_i \cdot L^{S'}_{0,i}\right]E'\right)$ instead).

In general, it will be the case that $E' = [b]E$ for some $b \in \mathbb{Z}/q\mathbb{Z}$ not necessarily known to the adversary; however, by subtracting known factors coming from corrupted players, the adversary can reduce to a distinguishing problem between $([\sum s'_i]E_0, [\sum s'_i a_i]E_0)$ and $([\sum s'_i]E_0, [r]E_0)$, where the s'_i are unknowns related to uncorrupted shares s_i, the a_i are known (and possibly 0), and r is random. This is an instance of a problem more general than Power-DDHA, and is thus at least as hard as Power-DDHA.

Hence, with non-negligible probability, either the adversary succeeds in distinguishing (generalized) Power-DDHA, or it produces a forged signature.

In the final game we replace the signature generation procedure with a signing oracle for the non-threshold scheme.

We still generate $k-1$ shares $s_i \in \mathbb{Z}/q\mathbb{Z}$ for the corrupted participants, however now the shared secret $s \in \mathbb{Z}/q\mathbb{Z}$ is only implicitly determined by the public key $[s]E_0$. This data fully determines the shares of the uncorrupted participants.

When the adversary sends a query for a signature, we pass it to the signing oracle, which returns a valid signature $(c_1 \cdots c_\lambda, z_1, \ldots, z_\lambda)$. From this signature, for any j, we draw $z_{i_1,j}, \ldots, z_{i_{k-1},j}$ at random, and set $z_{i_k,j} = z_j - z_{i_1,j} - \cdots - z_{i_{k-1},j}$.

We then proceed as in the previous game, thus producing exactly the same distribution. If the adversary produces a forgery, it is also a valid forgery for the non-threshold oracle. This proves that, assuming Power-DDHA, the threshold UF-CMA game is at least as hard as the non-threshold one. □

Remark 4. It is evident from the proof that the security of the (n, n)-threshold signature scheme can be proven without assuming Power-DDHA. The appearance of this surprising assumption seems an artifact related to the limitations of the HHS framework; indeed, the analogous scheme based on discrete logarithms can be proven as hard as standard Schnorr signatures without additional assumptions [54]. We hope that further research will improve the state of security proofs for HHS threshold schemes.

Remark 5. Although our scheme is unforgeable under *static* corruptions, it is obviously *non-robust*: any participant can lead to an invalid signature without being detected. Robustness can be added using generic zero-knowledge techniques, however it would be interesting to achieve it in a more efficient bespoke fashion.

Another desirable improvement would be to prove security in a stronger *adaptive* corruptions model, where the adversary can query the signing oracle before choosing which participants to corrupt.

4 Instantiations based on isogeny graphs

We now describe an instantiation of the previous schemes based on a principal homogeneous space of supersingular elliptic curves defined over a finite field $\mathbb{F}_p$.

It was first observed by Delfs and Galbraith [20] that the set of all supersingular curves defined over a prime field $\mathbb{F}_p$ partitions into one or two *levels*, each level being a principal homogeneous space for the class group of an order of the quadratic imaginary field $\mathbb{Q}(\sqrt{-p})$, in a way analogous to the well known theory of *complex multiplication*.

These principal homogeneous spaces were first used for a cryptographic purpose in the key-exchange scheme CSIDH [10], however only the precomputation performed recently by Beullens *et al.* for the signature scheme CSI-FiSh [4] permits to turn one of these into a true HHS.

We now briefly recall some key facts on CSIDH and CSI-FiSh, before turning to the instantiation of our threshold schemes. More details on the mathematical background of isogeny-based cryptography can be found in [15].

4.1 Supersingular complex multiplication

From now on we let p be a prime, $\mathbb{F}_p$ the field with p elements, and $\bar{\mathbb{F}}_p$ an algebraic closure. An elliptic curve E defined over $\mathbb{F}_p$ is said to be *supersingular* if and only if $\#E(\mathbb{F}_p) = p + 1$. It is well known that there are approximately $p/12$ isomorphism classes of supersingular curves, all defined over $\mathbb{F}_{p^2}$; of these, $O(\sqrt{p})$ are defined over $\mathbb{F}_p$.

Let E be a supersingular curve defined over $\mathbb{F}_p$, an *endomorphism* is an isogeny from E to itself, and it is said to be *defined over* $\mathbb{F}_p$ (or $\mathbb{F}_p$-rational) if it commutes with the *Frobenius endomorphism* π. The $\mathbb{F}_p$-rational endomorphisms of E form a ring, denoted by $\mathrm{End}_{\mathbb{F}_p}(E)$, isomorphic to an order[6] of $\mathbb{Q}(\sqrt{-p})$; more precisely, it is isomorphic to either $\mathbb{Z}[\sqrt{-p}]$ or $\mathbb{Z}[(\sqrt{-p}+1)/2]$. Let $\mathcal{O}$ be such an order, the *class group* $\mathrm{cl}(\mathcal{O})$ is the quotient of the group of invertible ideals of $\mathcal{O}$ by the group of its principal ideals; it is a finite abelian group.

The set of all supersingular curves with $\mathrm{End}_{\mathbb{F}_p}(E)$ isomorphic to a given order $\mathcal{O} \subset \mathbb{Q}(\sqrt{-p})$ is called the *horizontal isogeny class* associated to $\mathcal{O}$. A straightforward extension to the theory of complex multiplication states that the horizontal isogeny class of $\mathcal{O}$, up to $\mathbb{F}_p$-isomorphism, is a principal homogeneous space for $\mathrm{cl}(\mathcal{O})$. To make this into a HHS, an efficient (e.g., polynomial in $\log(p)$) algorithm to evaluate the action of $\mathrm{cl}(\mathcal{O})$ is needed. This is where isogenies play an important role. Fix an isomorphism $\mathrm{End}_{\mathbb{F}_p}(E) \simeq \mathcal{O}$, for any invertible ideal $\mathfrak{a}$, the action $\mathfrak{a} * E$ can be computed as follows: first define the $\mathfrak{a}$-*torsion* subgroup of E as

$$E[\mathfrak{a}] = \{P \in E(\bar{\mathbb{F}}_p) \ \mid \ \alpha(P) = 0 \text{ for all } \alpha \in \mathfrak{a}\},$$

this is a finite subgroup of E, and it is stabilized by the Frobenius endomorphism π; then the unique isogeny $\phi : E \to E/\langle E[\mathfrak{a}]\rangle$ with kernel $E[\mathfrak{a}]$ is such that $\mathfrak{a} * E = E/\langle E[\mathfrak{a}]\rangle$. It follows that, if $\mathfrak{a}$ and $\mathfrak{b}$ are two ideals in the same class, i.e., such that $\mathfrak{a} = (\alpha) \cdot \mathfrak{b}$ for some element $\alpha \in \mathcal{O}$, then $E/\langle E[\mathfrak{a}]\rangle \simeq E/\langle E[\mathfrak{b}]\rangle$.

The curve $E/\langle E[\mathfrak{a}]\rangle$ can be efficiently computed using an isogeny evaluation algorithm [57,27], however the complexity of this operation is polynomial in

[6] In this context, an order is a $\mathbb{Z}$-module isomorphic to $\mathbb{Z} \oplus \omega\mathbb{Z} \simeq \mathbb{Z}[\omega]$ for some $\omega \notin \mathbb{Q}$.

the degree of the isogeny, or, equivalently, in the norm $N(\mathfrak{a}) = \#(\mathcal{O}/\mathfrak{a})$. This implies that the action of an element $\mathfrak{a} \in \mathrm{cl}(\mathcal{O})$ can only be efficiently computed when a representative of small norm of $\mathfrak{a}$ is known, or, more generally, when a decomposition

$$\mathfrak{a} = \prod_i \mathfrak{l}_i$$

with all $\mathfrak{l}_i$ of small norm is known.

Now, for any prime ℓ, the ideal $(\ell) \subset \mathcal{O}$ is either prime, or it splits into a product of two (possibly equal) conjugate prime ideals $\mathfrak{l}\bar{\mathfrak{l}} = (\ell)$ of norm ℓ. In the former case, there are no invertible ideals of norm ℓ in $\mathcal{O}$; in the latter, $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ are the only ideals of norm ℓ, and they are the inverse of one another in $\mathrm{cl}(\mathcal{O})$. Asymptotically, about 50% of the primes ℓ split, thus we may hope to form a basis of generators of $\mathrm{cl}(\mathcal{O})$ of norms bounded by $\mathrm{polylog}(p)$, such that any element of $\mathrm{cl}(\mathcal{O})$ can be represented as a product of $\mathrm{polylog}(p)$ elements of the basis.[7]

This representation for the elements of $\mathrm{cl}(\mathcal{O})$ using a *smooth basis* is at the heart of the Couveignes–Rostovtsev–Stolbunov key exchange scheme, and of CSIDH. However, having a smooth basis may not be enough: to have a HHS, one still needs to be able to rewrite any element of $\mathrm{cl}(\mathcal{O})$ as a compact product of smooth elements. This is the key difference between CSIDH and CSI-FiSh, as we shall see next.

4.2 CSIDH and CSI-FiSh

CSIDH was designed to make evaluating the group action of $\mathrm{cl}(\mathcal{O})$ as efficient as possible. To this end, a prime p of the form

$$p + 1 = 4 \prod_{i=1}^{n} \ell_i$$

is selected, where $\ell_1, \ldots, \ell_{n-1}$ are the first $n-1$ odd primes, and ℓ_n is chosen so to make p prime. This choice guarantees several desirable properties:

- The curve $E : y^2 = x^3 + x$ has $\mathbb{F}_p$-rational endomorphism ring isomorphic to $\mathbb{Z}[\pi]$, where $\pi = \sqrt{-p}$ is the image of the Frobenius endomorphism of E;
- All curves in the horizontal isogeny class of $\mathbb{Z}[\pi]$ can be written in the form $y^2 = x^3 + Ax^2 + x$, and the coefficient A uniquely characterizes the $\mathbb{F}_p$-isomorphism class;
- All ℓ_i split in $\mathbb{Z}[\pi]$ as $(\ell_i) = \mathfrak{l}_i\bar{\mathfrak{l}}_i = \langle \ell_i, \pi - 1\rangle \cdot \langle \ell_i, \pi + 1\rangle$;
- For any curve E, the $\mathfrak{l}_i$-torsion subgroup is easily found as $E[\mathfrak{l}_i] = E[\ell_i] \cap E(\mathbb{F}_p)$.

[7] Jao, Miller and Venkatesan [37] showed that it is indeed possible to bound the norms by $O(\log^2(p))$, assuming the Generalized Riemann Hypothesis.

The first two properties ensure that supersingular isomorphism classes are easy to construct and represent uniquely, the third guarantees[8] that a number exponential in n of ideal classes of $\mathbb{Z}[\pi]$ can be efficiently represented and its action evaluated, the fourth enables some important optimizations for computing isogenies of degree ℓ_i.

In CSIDH and optimized variants [42,41,44,12], all ideal classes are implicitly represented as products

$$\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i},$$

with the exponents e_i in some box $[-B_i, B_i]$ (negative exponents are interpreted as powers of $\bar{\mathfrak{l}}_i$). Explicitly, the representation of an ideal class $\mathfrak{a}$ is simply the vector of exponents $(e_1, \ldots, e_n)$. The action of such ideals can be evaluated in time $\mathrm{poly}(B_i, e_i, n)$ using isogeny formulas.

In practice, a single parameter set has been fully specified for CSIDH, corresponding to the NIST post-quantum level 1.[9] The set has $n = 74$, $\ell_{73} = 373$, and $\ell_{74} = 587$, yielding a prime p of approximately 512 bits; we shall refer to it as CSIDH-512. Protocols based on CSIDH-512 usually sample exponents in a box $[-5, 5]$, which heuristically covers almost all the class group, and which permits to evaluate one class group action in under 30ms [42].

However, based on this data only, CSIDH is *not* a HHS. Indeed, all axioms of an HHS are satisfied but two: it is not possible to efficiently evaluate the action of *any* element of $\mathrm{cl}(\mathbb{Z}[\pi])$, and it is not always possible to test equality of two elements of $\mathrm{cl}(\mathbb{Z}[\pi])$. Take for example the exponent vector $(2^{128}, 0, \ldots, 0)$, corresponding to the ideal $\mathfrak{a} = \langle 3, \pi - 1 \rangle^{2^{128}}$; this is a valid element of $\mathrm{cl}(\mathbb{Z}[\pi])$, however without further knowledge its action can only be evaluated through 2^{128} isogeny evaluations. Hopefully, $\mathfrak{a}$ has an equivalent representation on the basis $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ with much smaller exponents, however we have no way to compute it and, even if we were given it, we could not test their equality.

These problems go away once we have computed the group structure of $\mathrm{cl}(\mathbb{Z}[\pi])$. More precisely, we need to know the *relation lattice* of $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$, i.e., the lattice

$$\Lambda = \left\{ (e_1, \ldots, e_n) \;\middle|\; \prod_{i=0}^{n} \mathfrak{l}_i^{e_i} = 1 \right\},$$

which yields a representation of the class group as $\mathrm{cl}(\mathbb{Z}[\pi]) \simeq \mathbb{Z}^n / \Lambda$. Now, equality of two exponent vectors $\boldsymbol{e}, \boldsymbol{f}$ can be tested by checking that $\boldsymbol{e} - \boldsymbol{f} \in \Lambda$, and any exponent vector $\boldsymbol{e}$ can be evaluated efficiently by finding an (approximate) closest vector $\boldsymbol{f} \in \Lambda$ and evaluating $\boldsymbol{e} - \boldsymbol{f}$ instead.

Neither computing the relation lattice, nor computing a good reduced basis for it are easy tasks: the former requires subexponential time in $\log(p)$, and the

[8] This guarantee is only heuristic: it is possible, although unlikely, that all $\mathfrak{l}_i$ have small order in $\mathrm{cl}(\mathbb{Z}[\pi])$, and thus generate a small subgroup.

[9] NIST defines the security of level 1 as being equivalent to AES-128.

latter exponential time in n.[10] Nevertheless, the computation for the CSIDH-512 parameter set happens to be just within reach of contemporary computers, as proven by Beullens *et al.* [4]: they managed to compute the structure of the class group, which happens to be cyclic of order

$$\begin{aligned}\#\mathrm{cl}(\mathbb{Z}[\pi]) = 3 \cdot 37 \cdot 1407181 \cdot 51593604295295867744293584889 \cdot \\ 3159941450468199585300827874558783220490 9 \approx 2^{257.136},\end{aligned} \tag{2}$$

and a BKZ-reduced basis for the relation lattice. In particular, they found out that the ideal $\mathfrak{l}_1 = \langle 3, \pi - 1 \rangle$ generates $\mathrm{cl}(\mathbb{Z}[\pi])$.

Thanks to CSI-FiSh, we thus dispose of a HHS with quantum security estimated at the NIST-1 security level, although scaling to higher security levels currently looks problematic.

4.3 Instantiation of the threshold schemes

Given the CSI-FiSh data, we can now instantiate our threshold schemes. However, it is evident by Eq. (2) that the full group $\langle \mathfrak{l}_1 \rangle = \mathrm{cl}(\mathbb{Z}[\pi])$ is not suitable for them, because the smallest prime factor of its order is 3, thus limiting the schemes to just 2 participants. We may instead choose as generator $\mathfrak{l}_1^3$, which limits the schemes to 36 participants, or $\mathfrak{l}_1^{111}$, allowing more than a million participants.[11]

Efficiency. The performance of our schemes can be readily inferred from that of the CSI-FiSh signature scheme.

To evaluate the action of an ideal in $\mathrm{cl}(\mathbb{Z}[\pi])$, CSI-FiSh first solves an approximate closest vector problem using Babai's nearest plane algorithm [1], and an algorithm by Doulgerakis, Laarhoven and de Weger [25]; then uses the isogeny evaluation algorithm of CSIDH. The average cost for one evaluation is reported to be $135.3 \cdot 10^6$ cycles (40–50ms on a commercial CPU), which is only 15% slower than the original CSIDH evaluation.[12]

In the encryption scheme, each participant computes exactly one class group action. Since the participants must do their computations sequentially, the total time for decryption is multiplied by the number of participants; the time for encryption, on the other hand, is unaffected by the number of participants, indeed the threshold nature of the protocol is transparent to the user.

[10] Using a quantum computer, the relation lattice can be computed in polynomial time, however lattice reduction still requires exponential time.

[11] An alternative way to allow up to 36 participants is to use the action of $\mathrm{cl}(\mathbb{Z}[(\pi + 1)/2])$ on the horizontal isogeny class of $y^2 = x^3 - x$: the class group is 3 times smaller than $\mathrm{cl}(\mathbb{Z}[\pi])$, and still generated by $\langle 3, \pi - 1 \rangle$. Because the two class group actions are compatible, the CSI-FiSh data can easily be repurposed for this variant without additional computations. This approach is detailed in [18].

[12] Benchmarks in [4] are based on the original CSIDH implementation [10]. A speed-up of roughly 30% is to be expected using the techniques in [42].

In the signature scheme, using the optimization described in [4], depending on the choice of parameters each participant computes between 6 and 56 group actions. Since the group action largely dominates the cost of the whole signing algorithm, we can expect to complete a (k, n)-threshold signature in approximately $k \cdot t \cdot 135.3 \cdot 10^6$ cycles, where $6 \leq t \leq 56$. However, the t group actions by a each participant are independent and can be computed in parallel; since the round-robin evaluation in the threshold scheme leaves plenty of idle cycles for participants while they wait for other participants' results, by carefully staggering the threshold group evaluations the k participants can evaluate the t group actions with the same efficiency as the non-threshold scheme, as long as $k \leq t$. According to [4, Tables 3,4], this would provide, for example, quantum-resistant threshold signatures for up to 16 participants in under 1 second, with public keys of 4 KB and signature size of only 560 B. Another example are 1880 B signatures with public key size of 128 B and k up to 56 in under 3 seconds; other interesting compromises are possible. These numbers compare favorably to other post-quantum threshold signatures that are expected to run in seconds [14], and may be especially interesting for side-channel protected implementations of CSI-FiSh.

Attacks. The security of the threshold schemes is essentially the same as that of the original single-participant signature and encryption schemes.

The fact that secrets are sampled in a subgroup of $\mathrm{cl}(\mathbb{Z}[\pi])$ of index 3 or 111 has a minor impact on security, as cryptanalyses can exploit this information to speed-up their searches.

In the classical setting, the best algorithm for both Vectorization and Parallelization is a random-walk approach [20] that finds a path between two supersingular curves in $O(\sqrt{\#\mathrm{cl}(\mathbb{Z}[\pi])}) = O(\sqrt[4]{p})$. If, like in our case, we restrict to a vertex set that is x times smaller, the random walk algorithm will find a collision approximately $\sqrt{x}$ times faster. Hence, we expect a loss in classical security of less than 4 bits.[13]

Note that this gain is optimal: if an algorithm could solve the Vectorization problem in a subgroup of size N/x more than $O(\sqrt{x})$ times faster, then by a divide and conquer approach the Vectorization problem in the full group of size N could be solved in less than $O(\sqrt{N})$ operations.

A similar gain can also be obtained in the best quantum algorithm for solving the Vectorization problem [40,39,48]. However, since its complexity is subexponential, the final gain is even less than 4 bits. The exact quantum security of CSIDH and CSI-FiSh is currently debated [10,3,6,8,47], nevertheless whatever the final consensus turns out to be, the quantum security of our threshold schemes will be extremely close to it.

[13] In reality, it is well known that the size of the search space can also be reduced by 3 in the original CSIDH, by *walking to the surface*. Thus, the only reduction in security comes from the factor of 37.

5 Conclusion

We introduced threshold variants of encryption and signature schemes based on Hard Homogeneous Spaces, and efficient quantum-safe instantiations thereof based on isogeny graphs of supersingular curves (CSIDH).

Our schemes are similar to well known Diffie–Hellman-style theshold schemes, however they are sharply different in the communication structure: whereas classical schemes have participants output messages in parallel with little coordination, our schemes impose a strictly sequential round-robin message passing style. Apparently, this limitation trickles down, negatively affecting many aspects: security properties, security proofs, efficiency.

In our El Gamal-style decryption algorithm, only one participant learns the cleartext, and we are only able to prove security in a *honest-but-curious* setting. The security of our signature scheme can be proven in a stronger security model with *static corruptions*, however it is not *robust*. Interesting questions for future research are efficient protocols where all participants learn the cleartext, or with stronger security properties, such as the ability to detect malicious participants.

Another topic we did not address in this work are verifiable distributed key generation algorithms, which would allow to run the threshold schemes without resorting to a trusted dealer. As observed by Benaloh [2], Shamir's secret sharing is $(+,+)$-homomorphic: given two secrets s and s' with respective shares s_i and s'_i, the sums of shares $s_i + s'_i$ form valid shares of $s + s'$. Based on this observation, Pedersen [45] constructed a DKG scheme without a trusted dealer, by having each party set up its own (k, n)-Shamir secret sharing scheme, and then combining these schemes using the homomorphic property.

While the same homomorphic property also applies to HHS threshold schemes, it seems difficult to achieve verifiability of the DKG like in [46,33,30]. An interesting research question is the construction of a verifiable DKG in the general HHS framework, or for specific isogeny-based instantiations.

Finally, the instantiation of our schemes is limited by the feasibility of parameter generation: to the present date the only available parameter set is the CSIDH-512 HHS, as computed by Beullens *et al.*, with security currently estimated at the NIST-1 level. Higher security levels would require extremely intensive computations that are currently out of reach.

Acknowledgment. We thank Gustavo Banegas, Tanja Lange, Chloe Martindale, and Dustin Moody for raising the topic of threshold cryptography at the Oxford PQC workshop. Further, we thank Jörn Steuding and the organizers of the summer school "Cryptography meets Graph Theory" in Würzburg for supporting Luca De Feo's visit, and thereby helping to bootstrap this collaboration.

References

1. Babai, L.: On Lovász lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)

2. Benaloh, J.C.: Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86. pp. 251–260. Springer (1986)
3. Bernstein, D.J., Lange, T., Martindale, C., Panny, L.: Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019. pp. 409–441 (2019)
4. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient Isogeny based Signatures through Class Group Computations. Cryptology ePrint Archive, Report 2019/498 (2019), https://eprint.iacr.org/2019/498
5. Beullens, W., Preneel, B., Szepieniec, A., Vercauteren, F.: LUOV. Round 2 submission, NIST Post-Quantum Cryptography Standardization (2019)
6. Biasse, J.F., Jacobson Jr, M.J., Iezzi, A.: A note on the security of CSIDH. In: Chakraborty, D., Iwata, T. (eds.) Progress in Cryptology - INDOCRYPT 2018. pp. 153–168. Springer (2018)
7. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the National Computer Conference. vol. 48 (1979)
8. Bonnetain, X., Schrottenloher, A.: Submerging CSIDH. Cryptology ePrint Archive, Report 2018/537 (2018), https://eprint.iacr.org/2018/537
9. Brando, L.T.A.N., Mouha, N., Vassilev, A.: Threshold Schemes for Cryptographic Primitives: Challenges and Opportunities in Standardization and Validation of Threshold Cryptography. NISTIR 8214 (2018), https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8214.pdf
10. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology - ASIACRYPT 2018. pp. 395–427. Springer (2018)
11. Castryck, W., Panny, L., Vercauteren, F.: Rational isogenies from irrational endomorphisms. Cryptology ePrint Archive, Report 2019/1202 (2019), https://eprint.iacr.org/2019/1202
12. Cervantes-Vázquez, D., Chenu, M., Chi-Domínguez, J.J., De Feo, L., Rodríguez-Henríquez, F., Smith, B.: Stronger and Faster Side-Channel Protections for CSIDH. To appear at LATINCRYPT 2019 (2019), https://eprint.iacr.org/2019/837
13. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006), https://eprint.iacr.org/2006/291
14. Cozzo, D., Smart, N.P.: Sharing the LUOV: Threshold Post-Quantum Signatures. Second PQC Standardization Conference (2019), https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/cozzo-luov-paper.pdf
15. De Feo, L.: Mathematics of isogeny based cryptography (2017), http://arxiv.org/abs/1711.04062
16. De Feo, L., Galbraith, S.D.: SeaSign: Compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019. pp. 759–789 (2019)
17. De Feo, L., Kieffer, J., Smith, B.: Towards practical key exchange from ordinary isogeny graphs. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology - ASIACRYPT 2018. pp. 365–394. Springer (2018)
18. Decru, T., Castryck, W.: CSIDH on the surface (2019), in preparation.
19. Decru, T., Panny, L., Vercauteren, F.: Faster seasign signatures through improved rejection sampling. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 271–285. Springer (2019)

20. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Designs, Codes and Cryptography **78**(2), 425–440 (Feb 2016). https://doi.org/10.1007/s10623-014-0010-1
21. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89. pp. 307–315. Springer (1990)
22. Desmedt, Y., Frankel, Y.: Shared generation of authenticators and signatures. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. pp. 457–469. Springer (1991)
23. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow. Round 2 submission, NIST Post-Quantum Cryptography Standardization (2019)
24. Doerner, J., Kondi, Y., Lee, E., Shelat, A.: Secure two-party threshold ECDSA from ECDSA assumptions. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 980–997. IEEE (2018)
25. Doulgerakis, E., Laarhoven, T., de Weger, B.: Finding closest lattice vectors using approximate Voronoi cells. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 3–22. Springer (2019)
26. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory **31**(4), 469–472 (1985)
27. Elkies, N.D.: Elliptic and modular curves over finite fields and related computational issues. In: Computational perspectives on number theory (Chicago, IL, 1995). Studies in Advanced Mathematics, vol. 7, pp. 21–76. AMS International Press, Providence, RI (1998)
28. Felderhoff, J.: Hard homogenous spaces and commutative supersingular isogeny based Diffie–Hellman (2019), https://perso.ens-lyon.fr/joel.felderhoff/work/M2_report.pdf
29. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86. pp. 186–194. Springer (1987)
30. Fouque, P.A., Stern, J.: One Round Threshold Discrete-Log Key Generation without Private Channels. In: Kim, K. (ed.) Public Key Cryptography - 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001. pp. 300–316. Springer (2001)
31. Gennaro, R., Goldfeder, S.: Fast Multiparty Threshold ECDSA with Fast Trustless Setup. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1179–1194. ACM (2018)
32. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust threshold DSS signatures. In: Maurer, U. (ed.) Advances in Cryptology - EUROCRYPT '96. pp. 354–371. Springer (1996)
33. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure Distributed Key Generation for Discrete-Log Cryptosystems. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99. pp. 295–310. Springer (1999)
34. Harn, L.: Group-oriented (t, n) threshold digital signature scheme and digital multisignature. IEE Proceedings-Computers and Digital Techniques **141**(5), 307–313 (1994)
35. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Renes, J., Soukharev, V., Urbanik, D., Pereira, G.: SIKE. Round 2 submission, NIST Post-Quantum Cryptography Standardization (2019)

36. Jao, D., De Feo, L.: Towards Quantum-Resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. Lecture Notes in Computer Science, vol. 7071, pp. 19–34. Springer Berlin / Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
37. Jao, D., Miller, S.D., Venkatesan, R.: Expander graphs based on GRH with an application to elliptic curve cryptography. Journal of Number Theory **129**(6), 1491–1504 (Jun 2009). https://doi.org/10.1016/j.jnt.2008.11.006
38. Kiltz, E.: A tool box of cryptographic functions related to the diffie-hellman function. In: Rangan, C.P., Ding, C. (eds.) Progress in Cryptology — INDOCRYPT 2001. pp. 339–349. Springer Berlin Heidelberg, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45311-3_32
39. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. TQC, Volume 22 of *LIPIcs*, pages 22-34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2013)
40. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM Journal on Computing **35**(1), 170–188 (2005)
41. Meyer, M., Campos, F., Reith, S.: On Lions and Elligators: An efficient constant-time implementation of CSIDH. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 307–325. Springer (2019)
42. Meyer, M., Reith, S.: A faster way to the CSIDH. In: Chakraborty, D., Iwata, T. (eds.) Progress in Cryptology - INDOCRYPT 2018. pp. 137–152. Springer (2018)
43. National Institute of Standards and Technology (NIST): Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016)
44. Onuki, H., Aikawa, Y., Yamazaki, T., Takagi, T.: (Short Paper) A Faster Constant-Time Algorithm of CSIDH Keeping Two Points. In: Advances in Information and Computer Security - 14th International Workshop on Security, IWSEC 2019. pp. 23–33. Springer (2019)
45. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT '91. pp. 522–526. Springer (1991)
46. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. pp. 129–140. Springer (1991)
47. Peikert, C.: He Gives C-Sieves on the CSIDH. Cryptology ePrint Archive, Report 2019/725 (2019), https://eprint.iacr.org/2019/725
48. Regev, O.: A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. arXiv preprint quant-ph/0406151 (2004), https://arxiv.org/abs/quant-ph/0406151
49. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145 (2006), http://eprint.iacr.org/2006/145
50. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89. pp. 239–252. Springer (1989)
51. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
52. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)

53. Shoup, V.: Practical threshold signatures. In: Preneel, B. (ed.) Advances in Cryptology — EUROCRYPT 2000. pp. 207–220. Springer Berlin Heidelberg, Berlin, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_15
54. Stinson, D.R., Strobl, R.: Provably secure distributed schnorr signatures and a (t, n) threshold scheme for implicit certificates. In: Varadharajan, V., Mu, Y. (eds.) Australasian Conference on Information Security and Privacy - ACISP 2001. pp. 417–434. Springer (2001)
55. Stolbunov, A.: 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)
56. Stolbunov, A.: Cryptographic schemes based on isogenies (2012)
57. Vélu, J.: Isogénies entre courbes elliptiques. C.R. Acad. Sc. Paris, Série A. **271**, 238–241 (1971)