CSI-RAShi: Distributed key generation for CSIDH

Ward Beullens[1], Lucas Disson[2], Robi Pedersen[1], and Frederik Vercauteren[1]

ward.beullens@esat.kuleuven.be, lucas.disson@ens-lyon.fr,
robi.pedersen@esat.kuleuven.be, frederik.vercauteren@esat.kuleuven.be

[1] imec-COSIC, ESAT, KU Leuven, Belgium
[2] ENS, Lyon, France

Abstract We present an honest-majority Distributed Key Generation protocol (DKG) based on Shamir's (k, n)-threshold secret sharing in the setting of Very Hard Homogenous Spaces (VHHS). DKG's in the DLOG setting use Pedersen commitments, for which there is no known analogue in the VHHS setting. As a replacement, we introduce a new primitive called *piecewise verifiable proofs*, which allow a prover to prove that a list of NP-statements is valid with respect to a common witness, and such that the different statements can be verified individually. Our protocol is robust and actively secure in the Quantum Random Oracle Model. For n participants, the total runtime of our protocol is $2 + \lambda + n(1 + 4\lambda)$ group action evaluations, where λ is the underlying security parameter, and is thus independent of the threshold k. When instantiated with CSIDH-512, this amounts to approximately $4.5 + 18n$ seconds.

Keywords: Isogeny-based cryptography, distributed key generation, secret sharing, class group action, CSIDH, QROM.

1 Introduction

Isogeny-based cryptography, proposed by Couveignes [7] and rediscovered by Rostovtsev and Stolbunov [23], is a very promising approach to post-quantum cryptography. Two different types of isogeny-based Diffie-Hellman key agreement schemes exist: Supersingular Isogeny Diffie-Hellman or SIDH [9] and its "commutative" variant called CSIDH [5]. Whereas SIDH relies on random walks in isogeny graphs over $\mathbb{F}_{p^2}$, CSIDH closely follows Couveignes' approach and constructs a so-called very hard homogeneous space (VHHS) based on supersingular curves over $\mathbb{F}_p$.

* This work was supported in part by the Research Council KU Leuven grants C14/18/067 and STG/17/019, and by CyberSecurity Research Flanders with reference number VR20192203. Date of this document: 22nd October 2020.

A VHHS is a natural generalisation of a group for which the decisional Diffie-Hellman problem is hard; in particular, exponentiation in the group is now replaced by a group action on a set. For CSIDH, the group action corresponds to the action of the ideal class group $\mathrm{cl}(\mathcal{O})$ on the set of supersingular elliptic curves over $\mathbb{F}_p$ whose $\mathbb{F}_p$-endomorphism ring is precisely $\mathcal{O}$.

In 2019, Beullens, Kleinjung and Vercauteren [3] computed the class group structure of the CSIDH-512 parameter set. Knowledge of the class group structure, for CSIDH-512 it is cyclic of order $N \approx 2^{256}$, allows to identify the ideal classes with integers mod N, which makes it possible to sample uniformly from the class group and represent the elements uniquely. This allowed Beullens *et al.* to instantiate a simple identification scheme that goes back to Couveignes, Rostovtsev and Stolbunov and combined with the Fiat-Shamir transform resulted in CSI-FiSh [3], which was the first practical post-quantum isogeny based signature scheme.

With the class group structure known, more cryptographic applications, including threshold signatures, threshold PKE and ring signatures are suddenly within reach [2,8,10]. This paper focuses on the threshold schemes. The idea of threshold schemes is that n participants are each given a share of a secret s, in such a way that any qualified subset of participants can reconstruct the secret or perform an action requiring the knowledge of s, such as signing a message or decrypting a ciphertext.

Threshold schemes have seen a surge of interest in recent years [19], due to their usage in voting schemes and blockchain applications among others [1,17]. Secret sharing schemes were initially introduced by Shamir in the late '70s [24] and first turned into a threshold ElGamal encryption scheme by Desmedt and Frankel [11]. Later, threshold signature schemes were proposed in the discrete-logarithm (DLOG) [14,15,16] and in the RSA setting [12,25]. A key question in these schemes is how to generate and share the secret s among all parties without s being revealed. While initial schemes relied on a trusted party called the dealer, in the early '90s, Pedersen [20] introduced the first distributed key generation (DKG) protocol in an honest majority k-out-of-n threshold, i.e. where at least k players are honest and at most $k-1$ malicious and each subset of k players is qualified. Pedersen's protocol was improved by Gennaro et al. [15] to a robust DKG scheme, i.e. where the reconstruction of s is possible, even if malicious players try to sabotage the computation.

Motivation and related work. De Feo and Meyer [10] introduced threshold variants of encryption and signature schemes in the VHHS setting and instantiated their protocols using CSIDH-512. Their approach is similar to DLOG schemes and use Shamir secret sharing. However, since in the VHHS setting, players can not individually combine partial signatures into the final signature, players have to compute their parts subsequently in a round-robin fashion. While efficient, the protocol by De Feo and Meyer is only passively secure and the key distribution is done by a trusted dealer. An alternative VHHS-based threshold

signing protocol called Sashimi was presented by Cozzo and Smart [8] based on replicated secret sharing. In contrast to [10], Sashimi is actively secure. This is achieved using zero-knowledge proofs, which have the downside of being computationally quite expensive: a signature in a $(2, 3)$-threshold takes around 5 minutes per party. Cozzo and Smart give a protocol to generate keys, but this protocol is not robust (an attacker can sabotage the computation), and the adversary can also influence the distribution of the public keys by selectively sabotaging the DKG protocol if it doesn't like the outcome. The question of how to robustly and securely perform a distributed key generation in the VHHS setting without the need for a trusted dealer has been left open by [10] and [8]. In this paper, we provide a robust, actively secure solution that is proven secure in the Quantum Random Oracle Model (QROM).

Our contributions. In this paper, we present CSI-RAShi,[3] a distributed key generation protocol (DKG) based on Shamir secret sharing in the VHHS setting. Our result is an honest-majority (k, n)-threshold scheme, that is also robust and actively secure. We achieve robustness by basing our protocol on the blueprint set out by Gennaro et al. [15], however the translation from the DLOG setting to VHHS presents several challenges: most importantly, there is no analogue of Pedersen commitments in the VHHS setting, and so verification of the shares is not possible in this way. We solve this problem by introducing a new primitive called *Piecewise Verifiable Proofs* (PVP). PVPs are zero-knowledge proofs of a list of NP-statements sharing a common witness, where individual relations (pieces) can be verified independently. In our DKG scheme, this allows parties to verify the correctness of individual shares at a low cost. As a result, in the isogeny setting, our protocol is very efficient in comparison to current isogeny-based distributed signature schemes, such as Sashimi [8]. Using recent results on the post-quantum security of the Fiat-Shamir transform [13,27], we prove security of the proposed PVPs, and consequently of our DKG scheme in the QROM.

Outline. In Section 2, we revisit the notions of very hard homogeneous spaces, Shamir secret sharing, and zero-knowledge proofs. In Section 3, we introduce security definitions for DKG schemes in the VHHS-setting and compare them to the DLOG setting. The following two sections present our main results: first we introduce the concept of piecewise verifiable proofs in Section 4, then we present our robust and actively secure protocol, CSI-RAShi, in Section 5. In Section 6, we instantiate this protocol in the isogeny setting and discuss the computational complexity. We conclude our results in Section 7.

[3] "Commutative Supersingular Isogeny Robust and Actively secure distributed Shamir secret sharing", pronounced *chirashi*, in reference to the Japanese dish *chirashi sushi*, translated as "scattered sushi".

2 Background

In this section we introduce notation and revisit some of the background needed in later sections. We denote by $\mathbb{Z}_N := \mathbb{Z}/N\mathbb{Z}$ the ring of integers modulo N, where N is a composite number with prime factorization $N = \prod_{i=1}^{s} q_i^{e_i}$, such that $q_1 < \cdots < q_s$.

2.1 Very hard homogeneous spaces

Hard homogeneous spaces were introduced by Couveignes [7] in order to generalize the notion of cyclic groups in which the computational Diffie-Hellman problem is hard. By adding a hard problem related to the decisional Diffie-Hellman problem, Couveignes further denotes them as very hard homogeneous spaces. We give a similar definition to the original one here using the notation common in the isogeny setting.

Definition 1 (Very hard homogeneous spaces [6,7,10]). *A* very hard homogeneous space *is a pair $(\mathcal{E}, \mathcal{G})$, where $\mathcal{G}$ is a finite Abelian group acting freely and transitively on a finite set $\mathcal{E}$ by the map $* : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$, for which there are easy (i.e. efficiently computable) and hard algorithmic problems. The easy problems are*

- Group operations*: Given $\mathfrak{a}$, decide whether it represents an element of $\mathcal{G}$. Given $\mathfrak{a}_1, \mathfrak{a}_2 \in \mathcal{G}$, compute a unique representation of $\mathfrak{a}_1$, compute $\mathfrak{a}_1^{-1}$, $\mathfrak{a}_1\mathfrak{a}_2$ or decide if $\mathfrak{a}_1 = \mathfrak{a}_2$.*
- Sampling*: Sample random elements uniformly from $\mathcal{G}$.*
- Membership *and* equality*: Given E, decide whether it is an element of $\mathcal{E}$. Given $E, E' \in \mathcal{E}$, decide if $E = E'$.*
- Action*: Given $\mathfrak{a} \in \mathcal{G}$, $E \in \mathcal{E}$, compute $\mathfrak{a} * E$.*

while the hard problems include

- Vectorization*: Given $E_1, E_2 \in \mathcal{E}$, find $\mathfrak{a} \in \mathcal{G}$, such that $\mathfrak{a} * E = E'$.*
- Parallelization*: Given $E_1, E_2, F_1 \in \mathcal{E}$ with $E_2 = \mathfrak{a} * E_1$, compute the unique $F_2 = \mathfrak{a} * F_1$.*
- Decisional Parallelization*: Distinguish with non-negligible advantage between the distributions $(\mathfrak{a} * E, \mathfrak{b} * E, \mathfrak{ab} * E)$ and $(\mathfrak{a} * E, \mathfrak{b} * E, \mathfrak{c} * E)$ where $E \in \mathcal{E}$ and $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ are chosen at random from $\mathcal{G}$.*

Notation. In the case where $\mathcal{G}$ is cyclic of order N, and $\mathfrak{g}$ is a generator of $\mathcal{G}$, we can also define the group action $[\,] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$, such that, for $a \in \mathbb{Z}_N$, $E \in \mathcal{E}$, we have $[a]E = \mathfrak{g}^a * E$. It then holds that $[a][b]E = [a + b]E$.

2.2 Shamir secret sharing modulo N

In this section, we revisit Shamir secret sharing [24] based on a (k, n)-threshold over the ring $\mathbb{Z}_N$. In these schemes, n mutually untrusted parties called players each hold a share of a common secret s, such that any subset of at least k players can efficiently reconstruct s, while any subset of $k-1$ or fewer players is unable to gain any information at all about s in an information-theoretic sense.

This is achieved via polynomial interpolation over the ring $\mathbb{Z}_N$, i.e. each player holds as a share the evaluation of a common polynomial $f(x) \in \mathbb{Z}_N[x]$ of degree $k-1$ at a specific position uniquely associated to them, e.g. player $\mathcal{P}_i$ for $i \in \{1, \ldots, n\}$ holds $s_i = f(i)$. We then call $\{s_i = f(i)\}_{i \in \{1,\ldots,n\}}$ a Shamir secret sharing of $s = f(0)$ with a (k, n)-threshold. Any subset S of at least k players can reconstruct s via Lagrange interpolation at $f(0)$ by computing

$$s = f(0) = \sum_{i \in S} s_i L_i^S, \tag{1}$$

where

$$L_i^S := L_{0,i}^S = \prod_{j \in S \setminus \{i\}} \frac{j}{j-i} \mod N \tag{2}$$

are the Lagrange basis polynomials evaluated at 0. If there are less than k players, the reconstruction will not succeed, since $f(0)$ is information-theoretically hidden. Shamir secret sharing is well known modulo a prime, but in this paper we consider it modulo a composite number N. To make this possible, we need $k \leq n < q_1$ [10], where q_1 is the smallest prime factor of N. This restriction ensures that the denominators in the Lagrange basis polynomials are coprime to N and thus invertible. It further guarantees perfect security of the secret sharing scheme over $\mathbb{Z}_N$ by Proposition 1 in [10].

2.3 Non-Interactive Commitment schemes

Our protocol makes use of secure non-interactive commitment schemes. In the remainder of the paper we assume a non-interactive commitment function $\mathcal{C} : \{0,1\}^\star \times \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, that takes as input a message $m \in \{0,1\}^\star$ and λ uniformly random bits bits, where λ is the security parameter, and outputs a 2λ-bit long commitment $\mathcal{C}(m, \mathsf{bits})$.

Intuitively, the commitment scheme should not reveal anything about the message it commits to, and it should not be possible to open the commitment to a different message. Instead of just assuming that the commitment is binding, we assume the stronger property of collapsingness as defined by Unruh in [26]. We also assume that $\mathcal{C}$ is quantum computationally hiding, which is formalized as follows:

Definition 2 (Quantum computational hiding). *For a quantum adversary $\mathcal{A}$ we define its advantage for the commitment hiding game for a pair of messages m, m' as*

$$\mathsf{Adv}^{\mathsf{Hiding}}_{\mathcal{C},\mathcal{A},m,m'} = \left| \Pr_{\mathsf{bits}\leftarrow\{0,1\}^\lambda}[\mathcal{A}(\mathcal{C}(m,\mathsf{bits}))=1] - \Pr_{\mathsf{bits}\leftarrow\{0,1\}^\lambda}[\mathcal{A}(\mathcal{C}(m',\mathsf{bits}))=1] \right|.$$

We say that $\mathcal{C}$ is quantum computationally hiding *if for all polynomial time quantum algorithms $\mathcal{A}$, and every pair of messages m, m' the advantage $\mathsf{Adv}^{\mathsf{Hiding}}_{\mathcal{C},\mathcal{A},m,m'}$ is a negligible function of the security parameter λ.*

2.4 Zero-Knowledge Proofs

In this section, we revisit the non-interactive version of the zero-knowledge proofs for simultaneous instances of the vectorization problem introduced in [8]. Let $s \in \mathbb{Z}_N$ be the unique witness, such that for a given indexed set $E_i, E_i' \in \mathcal{E}$ for $i \in \{1,\dots,m\}$, the following equations hold:

$$\forall\ i \in \{1,\dots,m\}: \quad E_i' = [s]E_i\,. \tag{3}$$

To prove knowledge of s, Algorithm 1 allows the prover to publish a proof π of statement (3), which can then be verified using Algorithm 2. Here, $\mathcal{H} : \{0,1\}^\star \to \{0,1\}^\lambda$ denotes a hash function (modelled as a quantum random oracle) used to generate the challenge. Both the prover and the verifier need to compute $m\lambda$ group actions.

Input : m tuples $\mathbf{X} = (E_i, E_i')_{i\in\{1,\dots,m\}}$ with $E_i, E_i' \in \mathcal{E}$, the secret s.
Output: A non-interactive proof π of relation (3).

for $j = 1,\dots,\lambda$ **do**
  $b_j \leftarrow \mathbb{Z}_N$ uniformly at random
  for $i = 1,\dots,m$ **do**
    $\hat{E}_{i,j} \leftarrow [b_j]E_i$
$\mathbf{c} = c_1 \dots c_\lambda \leftarrow \mathcal{H}(\mathbf{X} \parallel \hat{E}_{1,1} \parallel \dots \parallel \hat{E}_{m,1} \parallel \dots \parallel \hat{E}_{1,\lambda} \parallel \dots \parallel \hat{E}_{m,t})$
for $j = 1,\dots,\lambda$ **do**
  $r_j \leftarrow b_j - c_j s \mod N$
return $\pi = (\boldsymbol{c}, \boldsymbol{r})$, *where* $\boldsymbol{r} = (r_1,\dots,r_\lambda)$.

Algorithm 1: Non-interactive zero-knowledge proof $\mathsf{ZK}.P$

Input : m tuples $\mathbf{X} = (E_i, E_i')_{i\in\{1,\dots,m\}}$ with $E_i, E_i' \in \mathcal{E}$, a non-interactive proof $\pi = (\mathbf{c}, \mathbf{r})$.
Output: A boolean value signaling if the proof is deemed correct.

for $j = 1, \dots, \lambda$ **do**
  if $c_j = 0$ then let $\widetilde{E}_{i,j} \leftarrow [r_j]E_i$ for $i = 1, \dots, m$
  if $c_j = 1$ then let $\widetilde{E}_{i,j} \leftarrow [r_j]E_i'$ for $i = 1, \dots, m$
$\widetilde{c}_1 \dots \widetilde{c}_\lambda \leftarrow \mathcal{H}(\mathbf{X} \parallel \widetilde{E}_{1,1} \parallel \dots \parallel \widetilde{E}_{m,1} \parallel \dots \parallel \widetilde{E}_{1,\lambda} \parallel \dots \parallel \widetilde{E}_{m,\lambda})$
return $\widetilde{c}_1 \dots \widetilde{c}_\lambda == \mathbf{c}$

Algorithm 2: Non-interactive zero-knowledge verification $\mathsf{ZK}.V$

Theorem 1. *The algorithms 1 and 2 consitute a non-interactive zero-knowledge quantum proof of knowledge in the QROM for the relation (3).*

Proof. The work of Cozzo et al. [8] proves that the sigma protocol that underlies ZK has special soundness and honest verifier zero-knowledge (HVZK). Moreover, since the group action is free, it is clear that the sigma protocol has perfect unique responses. Therefore, the work of Don et al. [13] shows that the protocol is a quantum proof of knowledge. The work of Unruh [27] shows that because the sigma protocol has completeness, unpredictable commitments and HVZK, the protocol is zero-knowledge against quantum adversaries. □

3 Distributed key generation in the VHHS-setting

In this section, we introduce the security definitions for distributed protocols for generating Shamir secred shared keys for a HHS. We base our definitions on those introduced by Gennaro *et al.* [15], yet we have to use slightly weaker definitons, due to the difference of HHS and the DLOG setting.

3.1 Communication model

Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ denote the n players of the secret generation scheme, each being a probabilistic polynomial-time (PPT) algorithm. Analogous to [22], we assume that there are pairwise secure communication channels between the players, i.e. that can not be read or used except for the two concerned players. We also assume the existence of a reliable broadcast channel that identifies the sender and broadcasts the same message to all other players. Similar to [15] we assume these channels to be partially synchronous (as opposed to perfectly synchronous), meaning that sent messages on either channel are received within some fixed time bound.

3.2 Security definitions

In this section we give our security definitions for a Shamir secret sharing-based DKG protocol. We require correctness and secrecy.

The *correctness* requirement says that if there are at least k honest parties and at most $k-1$ malicious parties, the protocol will end with each honest party $\mathcal{P}_i$ holding a tuple (E, s_i), where all the honest parties agree on the same E. Moreover, the correctness requirement says that there exists a polynomial $f(x) \in \mathbb{Z}_N[x]_{\leq k-1}$ (i.e. a polynomial in $\mathbb{Z}_N[x]$ of degree $\leq k-1$), such that $E = [f(0)]E_0$ and $s_i = f(i)$, except with negligible probability. Our definition implies *robustness* as defined in [15], i.e. that the reconstruction of the secret should also be possible if malicious parties try to subvert the computation.

To formally state our security definitions we introduce the following notation: for two interacting (groups of) oracle algorithms B_1 and B_2 we denote by $b_1, b_2 \leftarrow \langle B_1^{\mathcal{O}}(x) | B_2^{\mathcal{O}}(y) \rangle$ the joint distribution of local outputs of B_1 and B_2 after running together on inputs x and y respectively. In the definitions below, we simply refer to the adversary's local output as A.

Definition 3 (Robust correctness). *We say a Shamir DKG protocol $\Pi = \{\mathcal{P}_i\}_{i\in\{1,\ldots,n\}}$ is* correct*, if for any PPT adversary $\mathcal{A}$, any positive integers $k \leq n$, and any subset $I \subseteq \{1,\ldots,n\}$ with $|I| \geq k$ and $n - |I| < k$ we have that*

$$\Pr\left[\begin{array}{c} \nexists f \in \mathbb{Z}_N[x]_{\leq k-1} : \\ E_1 = \cdots = E_n = [f(0)]E_0\,, \\ \text{and } \forall i \in I : f(i) = s_i \end{array} \middle| A, \{(E_i, s_i)\}_{i\in I} \leftarrow \langle \mathcal{A}^{\mathcal{O}}(1^\lambda) | \{\mathcal{P}_i^{\mathcal{O}}(1^\lambda)\}_{i\in I} \rangle \right]$$

is a negligible function of the security parameter.

For *secrecy*, we require that the protocol does not reveal anything about the secret key s beyond what can be learned from the value of the public key $E = [s]E_0$. This is formalized with a simulator-based definition. For every adversary $\mathcal{A}$, we require a simulator that, given a public key E chosen uniformly at random from $\mathcal{E}$, simulates honest parties, such that an execution where $\mathcal{A}$ interacts with the simulator results in E as the public key. The existence of such a simulator shows that the execution of the transcript can be generated from $E = [s]E_0$ alone, which means that the protocol does not reveal any information beyond what can be learned from E itself. More formally we have the following definition:

Definition 4 (Secrecy). *We say a Shamir DKG protocol $\Pi = \{\mathcal{P}_i\}_{i\in\{1,\ldots,n\}}$ has* secrecy*, if for any PPT aversary $\mathcal{A}$, and any index set of honest users $I \subseteq \{1,\ldots,n\}$ with $|I| \geq k$ and $n - |I| < k$, there exists a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that for any $i_0 \in I$, the following distributions are computa-*

tionally indistinguishable

$$\left\{(A, E_{i_0})\middle|A, \{(E_i, s_i)\}_{i\in I} \leftarrow \langle \mathcal{A}^{\mathcal{O}}(1^\lambda)|\{\mathcal{P}_i^{\mathcal{O}}(1^\lambda)\}_{i\in I}\rangle\right\} \approx_c \left\{(A, E)\middle|\begin{array}{c} E \leftarrow \mathcal{E} \\ A \leftarrow \langle \mathcal{A}^{\mathsf{Sim}_2}(1^\lambda)|\mathsf{Sim}_1(E, 1^\lambda)\rangle \end{array}\right\}.$$

Remark 1. The secrecy definition implies that, even in the presence of at most $k-1$ corrupted parties, the distribution of public keys is computationally indistinguishable from the uniform distribution, because the first distribution contains the common public key as computed by the protocol and the second distribution contains a uniformly random element of $\mathcal{E}$ instead.

3.3 Comparison to security definitions in DLOG setting

Our security definition is slightly weaker than the standard security definition for DLOG-based DKG protocols introduced by Gennaro et al. [15], because there is a subtle difference in the definition of the secrecy property. Both definitions require a simulator that, given a public key E, outputs a transcript of an execution of the protocol that results in E as a public key. The difference is that we only require the transcript to be indistinguishable from real transcripts that result in E as public key *if E is chosen uniformly at random*, whereas the standard definition in the DLOG setting requires the transcripts to be indistinguishable for every choice of E. In the DLOG setting this slightly stronger notion can be achieved using Pedersen commitments. This technique does not seem possible in the VHHS setting, so we have to rely on the parallelization problem, which requires E to be chosen uniformly at random. The property of Gennaro et al. is used to prove that the distribution of the public key is perfectly uniform, even in the presence of up to $k-1$ adversaries. In contrast, our property only implies that the distribution of the public key is indistinguishable from the uniform distribution.

4 Piecewise verifiable proofs

In this section we introduce zero-knowledge proofs that are piecewise verifiable. Given a list of NP relations $R_0, \ldots, R_n$ that share the same witness space and a list of statements $x_0, \ldots, x_n$, a piecewise verifiable proof (PVP) allows a prover to prove the existence of a witness w such that $(x_i, w) \in R_i$ for all $i \in \{0, \ldots, n\}$. The proof is of the form $\pi = (\tilde{\pi}, \{\pi_i\}_{i\in\{0,\ldots,n\}})$, where we think of $\tilde{\pi}$ as the central proof and of the π_i as proof pieces that are only relevant for R_i. The piecewise verifiability property says that for any $i \in \{0, \ldots, n\}$, given a statement piece x_i and a proof piece $(\tilde{\pi}, \pi_i)$ the verifier can check the proof with respect to x_i. If these piecewise verifications succeed for all $i \in I \subseteq \{0, \ldots, n\}$, then this

convinces the verifier of the existence of a witness w such that $(x_i, w) \in R_i$ for all $i \in I$. Crucially, we want the proof pieces not to leak information on the statements $\{x_i\}_{i \notin I}$.

In the following definitions, we use $x_I = \{x_i\}_{i \in I}$, $\pi_I = \{\pi_i\}_{i \in I}$ and $x = \{x_i\}_{i \in \{0,\ldots,n\}}$, and we write $(x_I, w) \in R_I$ and $(x, w) \in R$ if $(x_i, w) \in R_i$ for all $i \in I$ or for all $i \in \{0, \ldots, n\}$, respectively. We define a non-interactive piecewise verifiable proof (NIPVP) as follows:

Definition 5 (NIPVP in the QROM). *Let $R = R_0, \ldots, R_n$ be a list of NP relations that share the same witness space. A* non-interactive piecewise verifiable proof in the QROM *for R consists of two PPT algorithms $(P^{\mathcal{O}}, V^{\mathcal{O}})$ with quantum access to a random oracle $\mathcal{O}$ such that:*

- *$P^{\mathcal{O}}$ takes as input $x = (x_0, \cdots, x_n)$ and w such that $(x, w) \in R$ and outputs a proof $\pi = (\tilde{\pi}, \{\pi_i\}_{i \in \{0,\ldots,n\}})$.*
- *$V^{\mathcal{O}}$ takes as input a statement piece (i, x_i) and a proof piece $(\tilde{\pi}, \pi_i)$ and outputs 1 or 0, signaling that it accepts or rejects the proof, respectively.*

We require three properties of a non-interactive piecewise verifiable proof: completeness, soundness and zero-knowledge. The difference with conventional non-interactive proofs is that the properties need to hold with respect to any choice of $I \subseteq \{0, \ldots, n\}$. For the soundness property we require that if there does not exists a w such that $(x_i, w) \in R_i$ for all $i \in \{0, \ldots, n\}$, then a prover can not output accepting proof pieces $(\tilde{\pi}, \{\pi_i\}_{i \in I})$ (except with negligible probability). For the zero-knowledge property we require a simulator that simulates piecewise proofs given only a partial statement x_I. This implies that a set of proof pieces $(\tilde{\pi}, \{\pi_i\}_{i \in I})$ does not leak information on the witness or on $\{x_i\}_{i \notin I}$.

Definition 6 (completeness). *We say a NIPVP $(P^{\mathcal{O}}, V^{\mathcal{O}})$ for the list of relations R is complete if for any $(x, w) \in R$ and any $i \in \{0, \ldots, n\}$ we have*

$$\Pr[V^{\mathcal{O}}(i, x_i, \widetilde{\pi}, \pi_i) = 1 \,|\, \pi \leftarrow P^{\mathcal{O}}(x, w)] = 1\,.$$

Definition 7 (soundness). *For a NIPVP $(P^{\mathcal{O}}, V^{\mathcal{O}})$ for the list of relations R, a subset $I \subseteq \{0, \cdots, n\}$ and an adversary $\mathcal{A}$, we define the soundness advantage as*

$$\mathsf{Adv}^{\mathsf{sound}}_{\mathcal{A},I} = \Pr\left[\begin{matrix}\forall i \in I : V^{\mathcal{O}}(i, x_i, \widetilde{\pi}, \pi_i) = 1 \\ \nexists w : (x_I, w) \in R_I\end{matrix}\middle| (x_I, \tilde{\pi}, \pi_I) \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda)\right]\,.$$

We say that $(P^{\mathcal{O}}, V^{\mathcal{O}})$ is sound if for every polynomial-time quantum adversary $\mathcal{A}$ and all subsets $I \subseteq \{0, \cdots, n\}$ the advantage $\mathsf{Adv}^{\mathsf{sound}}_{\mathcal{A},I}$ is a negligible function of the security parameter.

Definition 8 (zero-knowledge). *We say a NIPVP $(P^{\mathcal{O}}, V^{\mathcal{O}})$ for the list of relations R is zero-knowledge if for any subset $I \subseteq \{0,\dots,n\}$, there exists a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$, such that for any poly-time quantum distinguisher $\mathcal{A}$ the distinguishing advantage*

$$\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{Sim},\mathcal{A}} = \left| \Pr\left[\mathcal{A}^{P',\mathcal{O}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{S,\mathsf{Sim}_2}(1^\lambda) = 1\right] \right|,$$

is a negligible function of the security parameter, where P' is an oracle that on input $(x,w) \in R$ runs $\pi := P^{\mathcal{O}}(x,w)$ and outputs $(\tilde{\pi}, \{\pi_i\}_{i\in I})$ and S is an oracle that on input $(x,w) \in R$ returns $\mathsf{Sim}_1(x_I)$ (i.e. Sim_1 does not get to see the witness or x_i for $i \notin E$.

4.1 Piecewise verifiable zero-knowledge proof

In this section, we present a piecewise verifiable zero-knowledge proof for the following list of relations $R = (R_0,\dots,R_n)$, whose common witness space is $\mathbb{Z}_N[x]_{\leq k-1}$, the set of polynomials over $\mathbb{Z}_N$ of degree at most $k-1$:

$$\begin{aligned} R_0 &= \{(x_0 = (E_0,E_1), f(x)) | [f(0)]E_0 = E_1\}, \\ \forall i \in \{1,\dots,n\} : R_i &= \{(x_i, f(x)) | f(i) = x_i\}. \end{aligned} \tag{4}$$

A statement for R_0 consists of a pair $(E_0,E_1) \in \mathcal{E}^2$, and a statement for the remaining relations $\{R_i\}_{i\in\{1,\dots,n\}}$ is an element of $\mathbb{Z}_N$.

Algorithm 3 and Algorithm 4 describe the NIPVP for relations of this form. They make use of a random oracle $\mathcal{H} : \{0,1\}^\star \to \{0,1\}^\lambda$, and a non-interactive commitment scheme $\mathcal{C} : \{0,1\}^\star \times \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, where λ is the security parameter.

Algorithm 3 requires the computation of λ group actions, one call to the random oracle $\mathcal{H}$ and $2(n+1)$ calls to the commitment scheme $\mathcal{C}$. Algorithm 4 requires one call to the random oracle $\mathcal{H}$, two calls to the commitment scheme $\mathcal{C}$, and only in the case $i = 0$ it requires the computation of λ group actions.

4.2 Security proof

Theorem 2. *Algorithms 3 and 4 constitute a complete, sound and zero-knowledge NIPVP in the QROM for the list of relations of (4) provided that the commitment scheme $\mathcal{C}$ is collapsing and quantum computationally hiding.*

Input : A witness polynomial $f(x) \in \mathbb{Z}_N[x]_{\leq k-1}$,
a statement $x = ((E_0, E_1), x_1, \cdots, x_n))$.
Output: A non-interactive piecewise proof π of the relations in (4).

for $j = 1, \ldots, \lambda$ **do**
  $b_j \leftarrow \mathbb{Z}_p[x]_{\leq k-1}$ uniformly at random
  $\hat{E}_j \leftarrow [b_j(0)]E_0$
$y_0, y_0' \leftarrow \{0,1\}^\lambda$ uniformly at random
$\mathsf{C}_0 \leftarrow \mathcal{C}(\hat{E}_1 \parallel \cdots \parallel \hat{E}_\lambda, y_0)$
$\mathsf{C}_0' \leftarrow \mathcal{C}(E_0 \parallel E_1, y_0')$
for $i = 1, \ldots, n$ **do**
  $y_i, y_i' \leftarrow \{0,1\}^\lambda$ uniformly at random
  $\mathsf{C}_i \leftarrow \mathcal{C}(b_1(i) \parallel \cdots \parallel b_\lambda(i), y_i)$
  $\mathsf{C}_i' \leftarrow \mathcal{C}(x_i, y_i')$
$\mathbf{c} = c_1 \ldots c_\lambda \leftarrow \mathcal{H}(\mathsf{C}, \mathsf{C}')$, where $\mathsf{C} = (\mathsf{C}_0, \ldots, \mathsf{C}_n), \mathsf{C}' = (\mathsf{C}_0', \ldots, \mathsf{C}_n')$
for $j = 1, \ldots, \lambda$ **do**
  $r_j(x) \leftarrow b_j(x) - c_j f(x) \mod N$
return $\tilde{\pi} = (\mathsf{C}, \mathsf{C}', \mathbf{r})$ *and* $\{\pi_i = y_i\}_{i \in \{0,\ldots,n\}}$, *where* $\mathbf{r} = (r_1, \ldots, r_\lambda)$.

Algorithm 3: NIPVP proof algorithm $\mathsf{PVP}.P$

The proof of this theorem is given in Appendix A. Even though our protocol can be seen as the Fiat-Shamir transformed version of a Sigma protocol with special soundness and quantum computationally unique responses we could not straightforwardly use the results of Don et al. [13]. This is because our protocol uses the "weak" Fiat-Shamir transform, where the challenge is determined by querying the random oracle on the commitment, rather than on the commitment and the statement. We can not feed x to the random oracle, because this would make it impossible to verify a proof piece without knowning the full statement x. Nevertheless, we could bootstrap the techniques from Don et al. to prove that our NIPVP was sound. To prove the zero-knowledge property, we used the result of Unruh [27] on the zero-knowledge of the Fiat-Shamir transform. Proving the completeness of the protocol is straightforward.

5 Distributed key generation

In this section, we describe CSI-RAShi, a DKG protocol based on the non-interactive piecewise verifiable proof of Section 4. The structure of our protocol is similar to the Gennaro protocol in the DLOG setting [15], which consists of 4 phases:

1. **Generating VSS.** In the first phase of the Gennaro protocol each party $\mathcal{P}_i$ performs a Pedersen verifiable secret sharing (VSS) protocol for a random

Input : An index $i \in \{0, \ldots, n\}$, a statement piece x_i of the form $(E_0, E_1) \in \mathcal{E}^2$ if $i = 0$, or $x_i \in \mathbb{Z}_N$ if $i \neq 0$, and a proof piece $(\widetilde{\pi}, \pi_i) = ((\mathsf{C}, \mathsf{C}', \mathbf{r}), (y_i, y_i'))$.
Output: A boolean value signaling if the proof is deemed correct

if $\mathsf{C}_i' \neq \mathcal{C}(x_i, y_i')$ **then**
  return 0
$c_1 \ldots c_\lambda \leftarrow \mathcal{H}(\mathsf{C}, \mathsf{C}')$
if $i == 0$ **then**
  for $j = 1, \ldots, \lambda$ **do**
    $\widetilde{E}_j \leftarrow [r_j(0)]E_{c_j}$
  return $\mathsf{C}_0 == \mathcal{C}(\widetilde{E}_1 \parallel \cdots \parallel \widetilde{E}_\lambda, y_0)$
else
  return $\mathsf{C}_i == \mathcal{C}(r_1(i) + c_1 x_i \parallel \cdots \parallel r_\lambda(i) + c_\lambda x_i \parallel x_i, y_i)$

Algorithm 4: NIPVP piecewise verification algorithm $\mathsf{PVP}.V$

value $z^{(i)} = f^{(i)}(0)$ to send to each player a share $s_{ij} = f^{(i)}(j)$ of a Shamir secret sharing of $z^{(i)}$.

2. **Verifying VSS.** In the second phase each party $\mathcal{P}_i$ uses the verifiability of the Pedersen VSS to check that the share it received from $\mathcal{P}_j$ is consistent. If a verification fails, then party $\mathcal{P}_i$ broadcasts a complaint against $\mathcal{P}_j$, and $\mathcal{P}_j$ gets the chance to clear his name. At the end of this phase, the honest parties will agree on a set $\mathcal{Q}$ of qualified parties who performed their VSS correctly.
3. **Compute shares.** The common secret key is implicitly defined as $\sum_{i \in \mathcal{Q}} z^{(i)}$, and each party can add the Shamir shares contributed by parties in $\mathcal{Q}$ in phase one to get their Shamir share of the common secret key.
4. **Compute common public key.** The common public key is defined as $\prod_{i \in \mathcal{Q}} g^{z^{(i)}}$. To compute this, each party publishes $g^{z^{(i)}}$ and the other parties can use the VSS to check whether this value is consistent with their shares or not. If it is not consistent, then the honest parties will agree on this, and they publish their shares s_{ij} such that $z^{(i)}$ can be publicly reconstructed.

The first problem that arises when trying to adapt this protocol to very hard homogeneous spaces is that there is no analogue of the Pedersen VSS in this setting. We solve this problem by using our piecewise verifiable zero-knowledge proof from Section 4 instead: in the first phase, each party $\mathcal{P}_i$ picks a polynomial $f^{(i)}(x)$ of degree at most $k-1$ and a $R^{(i)} \in \mathcal{E}$, then publishes $(R^{(i)}, R'^{(i)} = [f^{(i)}(0)]R)$ as a commitment to $z^{(i)} = f^{(i)}(0)$ and sends the share $s_{ij} = f^{(i)}(j)$ to party $\mathcal{P}_j$. It also constructs a piecewise verifiable proof $\pi = (\tilde{\pi}, \{\pi_i\}_{i \in \{0,\ldots,n\}})$ using $\mathsf{PVP}.P$ in order to prove that there exists a polynomial $f^{(i)}(x)$ such that:

$$R'^{(i)} = [f^{(i)}(0)]R^{(i)} \quad \text{and} \quad \forall j \in \{0, \ldots, n\} : f^{(i)}(j) = s_{ij}.$$

Using the piecewise verifiability, each party $\mathcal{P}_j$ uses $\mathsf{PVP}.V$ to verify that $R'^{(i)} = [f^{(i)}(0)]R^{(i)}$ and to check that $f^{(i)}(j) = s_{ij}$. The zero-knowledge property of the piecewise verifiable proof guarantees that the proof does not leak any information about s_{ij} to the other parties.

A second problem is that in the last phase of the Gennaro protocol, each party can just publish $g^{z^{(i)}}$, from which the public key $\prod_{i\in\mathcal{Q}} g^{z^{(i)}}$ can be computed. This is not possible in the VHHS setting, because it is not possible to compute $[\sum_{i\in\mathcal{Q}} z^{(i)}]E_0$, given $[z^{(i)}]E_0$ for all $i \in \mathcal{Q}$. To solve this problem we once again turn to zero-knowledge proofs, but in this case the standard (i.e. not piecewise verifiable) zero-knowledge proofs from Section 2.4 are sufficient. We let the first party $\mathcal{P}_1$ publish $F_1 = [z^{(1)}]E_0$, together with a zero-knowledge proof for the existence of a value $a \in \mathbb{Z}_N$ such that simultaneously $[a]E_0 = F_1$ and $[a]R^{(i)} = R'^{(i)}$. This proves that $\mathcal{P}_1$ honestly added his share of the secret key to E_0. Then, the remaining parties compute one-by-one, in a round-robin fashion, $F_i = [z^{(i)}]F_{i-1}$ and publish this value together with a proof that there exists an $a \in \mathbb{Z}_N$ such that $F_i = [a]F_{i-1}$ and $R'^{(i)} = [a]R^{(i)}$.

5.1 Our protocol.

Figure 1 presents $\mathsf{CSI\text{-}RAShi}$, a robust DKG protocol based on Shamir secret sharing and non-interactive piecewise verifiable zero-knowledge proofs.

Theorem 3. *If the proof* ZK *and the piecewise verifiable proof* PVP *are sound, then the distributed key generation protocol of Figure 1 satisfies the correctness requirement of Definition 3. Moreover, if additionally* $(\mathcal{E}, \mathbb{Z}_N)$ *constitutes a very hard homogeneous space with map* $\mathbb{Z}_N \times \mathcal{E} \to \mathcal{E} : (a, E) \mapsto [a]E$ *and if* ZK *and* PVP *are zero-knowledge, then the DKG protocol satisfies the secrecy requirement of Definition 4.*

Proof. **Consistency.** Suppose $\mathcal{A}$ is an adversary against the correctness property of the DKG protocol for an index set I of honest parties, then we construct adversaries $\mathcal{B}^{\mathcal{A}}_{\mathsf{ZK}}$ and $\mathcal{B}^{\mathcal{A}}_{\mathsf{PVP}}$ against the soundness of the ZK and PVP systems respectively and that have black box access to $\mathcal{A}$, such that if $\mathcal{A}$ breaks the correctness of the DKG protocol, then $\mathcal{B}_{\mathsf{ZK}}$ breaks the soundness of the ZK protocol or $\mathcal{B}_{\mathsf{PVP}}$ breaks the soundness of the PVP protocol.

The adversary $\mathcal{B}^{\mathcal{A}}_{\mathsf{ZK}}$ works as follows: he simulates the set of honest parties $\{\mathcal{P}_i\}_{i\in I}$ and engages in the DKG protocol with adversary $\mathcal{A}$. Then, at the end of the protocol, $\mathcal{B}_{\mathsf{ZK}}$ picks at random an index i in $\mathcal{Q} \setminus I$ and outputs the statement $(R^{(i)}, R'^{(i)}, F_{i-1}, F_i)$ and the corresponding proof $\pi'^{(i)}$. The adversary $\mathcal{B}^{\mathcal{A}}_{\mathsf{PVP}}$ works very similarly: it simulates honest parties and executed the DKG protocol with $\mathcal{A}$. Then, at the end of the protocol it again picks a random index i in $\mathcal{Q} \setminus I$ and outputs the statement piece $x_I^{(i)} = (x_0^{(i)} = (R^{(i)}, R'^{(i)}), \{x_j^{(i)}\}_{j\in I})$ as well as the proof piece $(\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in I})$.

CSI-RAShi

Generating the VSS. Each $\mathcal{P}_i$ samples $f^{(i)}(x)$ and $R^{(i)}$ uniformly from $\mathbb{Z}_N[x]_{\leq k-1}$ and $\mathcal{E}$, respectively, then determines $R'^{(i)} = [f^{(i)}(0)]R^{(i)}$ and computes the full statement

$$x^{(i)} = (x_0^{(i)} = (R^{(i)}, R'^{(i)}), \{x_j^{(i)} = f^{(i)}(j)\}_{j\in\{1,\ldots,n\}}) .$$

Then, it constructs a piecewise verifiable proof

$$\pi^{(i)} = (\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in\{0,\ldots,n\}}) \leftarrow \mathsf{PVP}.P^{\mathcal{O}}(x^{(i)}, f^{(i)}(x))$$

and publishes $(x_0^{(i)}, \tilde{\pi}^{(i)}, \pi_0^{(i)})$ and sends $(x_j^{(i)}, \pi_j^{(i)})$ privately to $\mathcal{P}_j$.

Verifying the VSS. Each $\mathcal{P}_j$ verifies all the proof pieces with respect to the $R'^{(i)} = [f^{(i)}(0)]R^{(i)}$ and the $f^{(i)}(j) = s_{ij}$ part of the statement: For each $i \neq j$ it runs $\mathsf{PVP}.V^{\mathcal{O}}(0, x_0^{(i)}, \tilde{\pi}^{(i)}, \pi_0^{(i)})$ and $\mathsf{PVP}.V^{\mathcal{O}}(j, x_j^{(i)}, \tilde{\pi}^{(i)}, \pi_j^{(i)})$. If at least one of these checks fails $\mathcal{P}_j$ broadcasts a complaint against $\mathcal{P}_i$.

Any player with at least k complaints is disqualified. If $\mathcal{P}_j$ complains that $\mathcal{P}_i$'s proof does not verify, $\mathcal{P}_i$ responds by broadcasting $(x_j^{(i)}, \pi_j^{(i)})$ so that everyone can verify $\mathsf{PVP}.V^{\mathcal{O}}(j, x_j^{(i)}, \tilde{\pi}^{(i)}, \pi_j^{(i)})$. If this verification succeeds, the protocol continues as normal, otherwise $\mathcal{P}_i$ is disqualified. Since disqualifying players happens on the basis of only broadcasted information, all the honest players will agree on the same set of qualified parties $\mathcal{Q} \subset \{1, \ldots, n\}$.

Compute shares. At this point the joint secret key is implicitly defined as $s = \sum_{i\in Q} f^{(i)}(0)$. Each party $\mathcal{P}_j$ derives their share of s as $s_j = \sum_{i\in\mathcal{Q}} x_j^{(i)}$.

Compute common public key.

1. In a round-robin way, the qualified players compute $F_i = [f^{(i)}(0)]F_{i-1}$, where $F_0 = E_0$. At each step, player $\mathcal{P}_i$ publishes the proof
$$\pi'^{(i)} \leftarrow \mathsf{ZK}.P((R^{(i)}, R'^{(i)}), (F_{i-1}, F_i), f^{(i)}(0)),$$
which is verified by all other parties.
2. If $\mathcal{P}_j$ finds that the proof by player $\mathcal{P}_i$ is wrong, it publishes $(x_j^{(i)}, \pi_j^{(i)})$. Then, every party runs $\mathsf{PVP}.V^{\mathcal{O}}(j, x_j^{(i)}, \tilde{\pi}^{(i)}, \pi_j^{(i)})$ for all the published pairs. If there are k honest players, then at least k parties can publish a tuple $(x_j^{(i)}, \pi_j^{(i)})$ for which the verification will succeed, so the honest parties can all reconstruct $f^{(i)}(0)$, compute F_i, and continue the protocol.
3. The parties return $F_{|\mathcal{Q}|}$ as their public key.

Figure 1: CSI-RAShi, a robust DKG protocol using NIPVPs for the relations $R_0 = \{((R, R'), f(x))|[f(0)]R = R'\}$ and $R_i = \{(x_i, f)|x_i = f(i)\}$ with witness space $\mathbb{Z}_N[x]_{\leq k-1}$.

Let $\mathsf{Break}_{\mathsf{ZK}}$ be the event that at least one of the ZK proofs sent by a party in the qualified set $\mathcal{Q}$ controlled by $\mathcal{A}$ is a valid proof for an invalid statement. If that event occurs, then $\mathcal{B}_{\mathsf{ZK}}$ will pick that proof and output it with probability $1/|\mathcal{Q} \setminus I| > 1/(n - |I|)$, so the advantage of $\mathcal{B}_{\mathsf{ZK}}$ against the soundness property of ZK is at least $\Pr[\mathsf{Break}_{\mathsf{ZK}}]/(n - |I|)$. Similarly, if $\mathsf{Break}_{\mathsf{PVP}}$ is the event that at least one of the PVP proofs sent by a party in the qualified set $\mathcal{Q}$ controlled by $\mathcal{A}$ is a valid proof piece of an invalid statement piece, then the advantage of $\mathcal{B}_{\mathsf{PVP}}$ against the soundness property of PVP is at least $\Pr[\mathsf{Break}_{\mathsf{PVP}}]/(n - |I|)$.

If the event $\mathsf{Break}_{\mathsf{PVP}}$ does not occur, then for all $i \in \mathcal{Q}$, the partial statement $((R^{(i)}, R'^{(i)}), \{x_I^{(i)}\}_{j \in I})$ is valid, meaning that there exists $f^{(i)}(x) \in \mathbb{Z}_N[x]_{\leq k-1}$, such that $R'^{(i)} = [f^{(i)}(0)]R^{(i)}$ and such that $x_j^{(i)} = f^{(i)}(j)$ for all $j \in I$. Therefore, the honest parties $\{\mathcal{P}_j\}_{j \in I}$ will hold consistent Shamir shares $x_j^{(i)}$ of $f^{(i)}(0)$, and they will output a consistent sharing of $f(0)$, where $f(x) = \sum_{i \in \mathcal{Q}} f^{(i)}(x)$. Given that all the Shamir shares are consistent, it is guaranteed that for all the ZK proofs $\pi'^{(i)}$ that fail to verify in the last phase of the protocol the honest parties will successfully reconstruct $f^{(i)}(0)$, so they will be able to compute $F_i = [f^{(i)}(0)]F_{i-1}$. If $\mathsf{Break}_{\mathsf{ZK}}$ does not occur, then all the statements $(R^{(i)}, R'^{(i)}, F_{i-1}, F_i)$ for the remaining (valid) proof are consistent, which also implies that $F_i = [f^{(i)}(0)]F_{i-1}$ in this case. Therefore, the honest parties will obtain the common public key $F_n = [f(0)]E_0$.

We have proven that if neither $\mathsf{Break}_{\mathsf{PVP}}$ nor $\mathsf{Break}_{\mathsf{ZK}}$ occurs, then the honest parties obtain a consistent sharing of the secret key that corresponds to the common public key F_n, so in this case $\mathcal{A}$ does not win the correctness game against the DKG protocol. This means that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{correctness}} \leq \Pr[\mathsf{Break}_{\mathsf{ZK}}] + \Pr[\mathsf{Break}_{\mathsf{PVP}}]$, so

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{correctness}} \leq (n - |I|) \cdot \left(\mathsf{Adv}_{\mathcal{B}_{\mathsf{ZK}}}^{\mathsf{sound}} + \mathsf{Adv}_{\mathcal{B}_{\mathsf{PVP}}}^{\mathsf{sound}} \right) .$$

Therefore, if $\mathcal{A}$ is a PPT adversary with a non-negligible advantage against the correctness property of the DKG protocol, then at least one of $\mathcal{B}_{\mathsf{ZK}}$ or $\mathcal{B}_{\mathsf{PVP}}$ will also be a PPT adversary with non-negligible advantage against the soundness of ZK or PVP respectively.

Secrecy. Let $\mathcal{A}$ be an adversary that statically corrupts at most $k-1$ parties, and let $I \subset \{1, \ldots, n\}$ be the set of uncorrupted parties. We construct a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ as in the secrecy definition 4. Given a random element $E \in \mathcal{E}$, the simulator has to simulate the honest parties $\{\mathcal{P}_i\}_{i \in I}$ and the random oracle such that the simulation is indistinguishable from an execution of the DKG protocol where $\mathcal{A}$ is interacting with honest parties, and where E is the resulting common public key.

To prove this theorem we introduce a sequence of four simulators. The first simulator $\mathsf{Sim}^{(0)}$ faithfully simulates honest parties (and hence does not enforce that

E is the common public key). Then we incrementally modify the simulator to get simulators $\mathsf{Sim}^{(1)}, \mathsf{Sim}^{(2)}, \mathsf{Sim}^{(3)}$, and we will prove that the final simulator $\mathsf{Sim}^{(3)}$ satisfies the requirements of the secrecy definition.

$\mathsf{Sim}^{(0)}$: All the simulators consist of two parts (with a shared state), a part that simulates the honest parties, and a part that simulates the random oracle. The first simulator $\mathsf{Sim}^{(0)}$ consists of $\mathsf{Sim}_1^{(0)}$, which ignores the input element E and just simulates the honest parties faithfully, and $\mathsf{Sim}_2^{(0)}$, which simulates a random oracle by keeping a list of queries. Therefore, the following two distributions are identical:

$$\begin{aligned}\left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathcal{O}}(1^\lambda)|\{\mathcal{P}_i^{\mathcal{O}}(1^\lambda)\}_{i\in I}\rangle\right\}=\\ \left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(0)}}(1^\lambda)|\mathsf{Sim}_1^{(0)}(E,1^\lambda)\rangle\right\},\end{aligned}$$

where E' represents any element from the set $\{E_i\}_{i\in I}$.

$\mathsf{Sim}^{(1)}$: We fix an honest party $\mathcal{P}_s$ for $s \in I$. In this step we will use the zero-knowledge simulator of the PVP protocol $\mathsf{Sim}^{\mathsf{PVP}} = (\mathsf{Sim}_1^{\mathsf{PVP}}, \mathsf{Sim}_2^{\mathsf{PVP}})$ to simulate the piecewise verifiable proof $\pi^{(s)}$ from party $\mathcal{P}_s$. The only difference between $\mathsf{Sim}_1^{(0)}$ and $\mathsf{Sim}_1^{(1)}$ is that $\mathsf{Sim}_1^{(1)}$ does not generate $\pi^{(s)}$ honestly, but instead it calls $(\tilde{\pi}^{(s)}, \{\pi_i^{(s)}\}_{i\notin I}) \leftarrow \mathsf{Sim}_1^{\mathsf{PVP}}(\{x_i^{(s)}\}_{i\notin I})$. The second part $\mathsf{Sim}_2^{(1)}$ still simulates a random oracle by maintaining a list of queries, except that it forwards the queries for the random oracle for the PVP proof to $\mathsf{Sim}_2^{\mathsf{PVP}}$. Note that we assume that the domain of the random oracle queries for the PVP protocol is separated from domain of the queries for the ZK protocol, such that this selective forwarding is possible. A PPT adversary that distinguishes $\mathsf{Sim}^{(0)}$ from $\mathsf{Sim}^{(1)}$ with a non-negligible advantage would break the assumption that the PVP protocol is zero-knowledge, so we know that the following distributions are computationally indistinguishable

$$\begin{aligned}\left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(0)}}(1^\lambda)|\mathsf{Sim}_1^{(0)}(E,1^\lambda)\rangle\right\}\approx_c\\ \left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(1)}}(1^\lambda)|\mathsf{Sim}_1^{(1)}(E,1^\lambda)\rangle\right\}.\end{aligned}$$

$\mathsf{Sim}^{(2)}$: This step is similar to the previous step, but now we use the zero-knowledge simulator $\mathsf{Sim}^{\mathsf{ZK}} = (\mathsf{Sim}_1^{\mathsf{ZK}}, \mathsf{Sim}_2^{\mathsf{ZK}})$ of the ZK protocol. The only difference between $\mathsf{Sim}_1^{(1)}$ and $\mathsf{Sim}_1^{(2)}$ is that $\mathsf{Sim}_1^{(2)}$ does not generate $\pi'^{(s)}$ honestly, but instead it calls $\pi'^{(s)} \leftarrow \mathsf{Sim}_1^{\mathsf{ZK}}(R^{(s)}, R'^{(s)}, F_s, F_{s-1})$. The simulator $\mathsf{Sim}_2^{(1)}$ forwards queries to the ZK random oracle to $\mathsf{Sim}_2^{\mathsf{ZK}}$, instead of answering the queries by itself. (We again assume that the domains for the two different functions of the random oracle are separated). Assuming that the ZK protocol is zero-knowledge,

the following distributions are computationally indistinguishable

$$\left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(1)}}(1^\lambda)|\mathsf{Sim}_1^{(1)}(E,1^\lambda)\rangle\right\}\approx_c$$
$$\left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(2)}}(1^\lambda)|\mathsf{Sim}_1^{(2)}(E,1^\lambda)\rangle\right\}.$$

$\mathsf{Sim}^{(3)}$: The final simulator enforces that E is the common public key. At the beginning of the public key generation phase the simulator computes

$$f'(x)=\sum_{\substack{i\in\mathcal{Q}\\ i>s}} f^{(i)}(x).$$

The simulator knows all the $f^{(i)}(x)$ for $i\in\mathcal{Q}$, because he either chose $f^{(i)}(x)$ himself, or he received $|I|\geq k$ shares from the adversary. The soundness property of the PVP proof implies that if $\mathcal{A}$ runs in polynomial time, then with all but a negligible probability the shares will be consistent shares on a polynomial $f^{(i)}(x)$, which the simulator can reconstruct. Then, instead of computing $F_s=[f^{(s)}(0)]F_{s-1}$, the simulator computes $F_s=[-f'(0)]E$. With this modification the common public key will result in $[f'(0)]F_s=[f'(0)-f'(0)]E=E$, because the soundness of the ZK protocol guarantees that if the proof produced by party $\mathcal{P}_i$ is valid, then $F_i=[f^{(i)}(0)]F_{i-1}$, and if the proof is not correct, then the soundness of the PVP protocol guarantees that the honest parties will reconstruct $f^{(i)}(0)$ and compute $F_i=[f^{(i)}(0)]F_{i-1}$. What remains to prove is that

$$\left\{(A,E')\middle|A,\{(E_i,s_i)\}_{i\in I}\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(2)}}(1^\lambda)|\mathsf{Sim}_1^{(2)}(E,1^\lambda)\rangle\right\}\approx_c$$
$$\left\{(A,E)\middle|\begin{array}{c}E\leftarrow\mathcal{E}\\ A\leftarrow\langle\mathcal{A}^{\mathsf{Sim}_2^{(3)}}(1^\lambda)|\mathsf{Sim}_1^{(3)}(E,1^\lambda)\rangle\end{array}\right\}.$$

We prove this with a reduction to the parallelization problem. Suppose $\mathcal{A}'$ is a PPT algorithm that distinguishes the two distributions with non-negligible probability, then we construct a PPT algorithm $\mathcal{B}'^{\mathcal{A}'}$ that makes black box access to $\mathcal{A}'$ and that solves the parallelization problem with the same advantage. The adversary $\mathcal{B}'$ works as follows: given input (E_a,E_b,E_c) it runs $\mathsf{Sim}^{(2)}$ except that instead of picking $R^{(s)}$ at random and setting $R'^{(s)}=[f^{(s)}(0)]R$ it now sets $R^{(s)}=E_b$ and $R'^{(s)}=E_c$. Since the simulator only uses $n-|I|<k$ evaluations of the random polynomial $f^{(s)}(x)$, it holds that $f^{(s)}(0)$ is information theoretically hidden to the adversary, so this change does not affect the view of $\mathcal{A}$. Instead of putting $F_s=[f^{(s)}(0)]F_{s-1}$, $\mathcal{B}'$ also computes

$$F_s=\left[\sum_{\substack{i\in Q\\ i<s}} f^{(i)}(0)\right]E_a\,.$$

Then $\mathcal{B}'$ forwards (A,E') to $\mathcal{A}'$, where A is the local output of $\mathcal{A}$, and E' is the public key outputted by any honest party. Finally $\mathcal{B}'$ outputs whatever $\mathcal{A}'$

outputs.

If E_a, E_b, E_c was a random instance of the parallelization problem, with $a \in \mathbb{Z}_N$ such that $E_a = [a]E_0$ and $E_c = [a]E_b$, then $R^{(s)} = [a]R'^{(s)}$ and $F_s = [a]F_{s-1}$, so the input to $\mathcal{A}'$ exactly follows the distribution

$$\left\{ (A, E') \middle| A, \{(E_i, s_i)\}_{i \in I} \leftarrow \langle \mathcal{A}^{\mathsf{Sim}_2^{(2)}}(1^\lambda) | \mathsf{Sim}_1^{(2)}(E, 1^\lambda) \rangle \right\},$$

and if E_a, E_b, E_c was a uniformly random triple, then $R^{(s)}, R'^{(s)}$ and F_s are uniformly random, just like in the case of $\mathsf{Sim}^{(3)}$, so the input to $\mathcal{A}'$ exactly follows the distribution

$$\left\{ (A, E) \middle| \begin{matrix} E \leftarrow \mathcal{E} \\ A \leftarrow \langle \mathcal{A}^{\mathsf{Sim}_2^{(3)}}(1^\lambda) | \mathsf{Sim}_1^{(3)}(E, 1^\lambda) \rangle \end{matrix} \right\},$$

Therefore, the advantage of $\mathcal{B}'$ for solving the decisional parallelization problem is the same as the distinguishing advantage of $\mathcal{A}'$. The decisional parallelization assumption therefore implies that the two distributions are computationally indistinguishable, which concludes the proof. □

5.2 Cost

During an execution of the VSS generation and VSS verification steps, where all the parties behave honestly, each party computes $2 + n\lambda$ isogeny group actions. These computations can be completely done in parallel. Cheating parties can force additional verifications in the VSS verification phase, but these verifications only involve symmetric operations, so this comes at a negligible computational cost: in the worst-case an honest party has to hash $O(n(n-k)\lambda^2)$ bits, which is negligible for all practical values of n, k.
Because of the round-robin, the public key generation step is innately sequential, especially since players need to verify the correctness of F_{i-1} before computing F_i. We note, however that the verifications at a specific step in the round robin can all be done in parallel. Further, in the first round, while $\mathcal{P}_1$ construct its proof, all other players can already start constructing half of their proof $\pi^{(i)}$ by computing the commitments for the relation $(R^{(i)}, R'^{(i)})$. Thus, the first round takes $1 + 4\lambda$ total time to evaluate, while every subsequent round only takes $1 + 3\lambda$. This yields a total time of $\lambda + n(1 + 3\lambda)$ group actions in the public key computation step of the protocol (assuming no player was disqualified). Similarly to the first part of the protocol, in the public key generation phase any additional checks caused by cheating parties do not require group action evaluations. However, if party $\mathcal{P}_i$ is dishonest in the public key generation phase, the honest players will have to compute the F_i themselves. If all of the up to $n - k$ corrupt parties misbehave, this means the honest parties have to compute $n - k$ additional group action evaluations. This is small compared to the total

cost of an honest execution of the protocol.
The total sequential cost of the protocol thus takes $T(n, \lambda) = 2 + \lambda + n(1 + 4\lambda)$ group action evaluations while the actual computational effort per player is $T_{\mathcal{P}}(n, \lambda) = 3(1 + n\lambda)$. Note that both costs are independent of the threshold k.

6 Instantiation based on isogenies

In this section, we look at the VHHS instantiation from supersingular elliptic curve isogeny graphs. In this scenario, the elements in $\mathcal{E}$ are supersingular elliptic curves defined over the finite field $\mathbb{F}_p$ with a certain endomorphism ring $\mathcal{O}$. This endomorphisms ring is isomorphic to an order $\mathcal{O}$ of the quadratic imaginary field $\mathbb{Q}(\sqrt{-p})$. In the CSIDH [5] and CSI-FiSh [3] settings, this order is chosen to be $\mathbb{Z}[\sqrt{-p}]$ and its class group $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$ acts freely and transitively on elements of $\mathcal{E}$ as follows

$$\mathfrak{a} * E = E/E[\mathfrak{a}], \quad \text{where} \quad E[\mathfrak{a}] = \{P \in E(\mathbb{F}_p) \mid \alpha(P) = 0 \ \forall \alpha \in \mathfrak{a}\}$$

for some $\mathfrak{a} \in \mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$. For efficiency reasons p is generally chosen as

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

where $\ell_1 < \cdots < \ell_{n-1}$ are the first $n-1$ odd primes and ℓ_n is chosen, so that p becomes prime. With this choice, the action of an element $\mathfrak{a} \in \mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$ can be efficiently computed as consecutive evaluations of ℓ_i-isogenies by representing them as

$$\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}, \tag{5}$$

where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and where the exponents are bound in some short interval $[-b, b]$, b a small integer. The negative exponents correspond to the action by $\bar{\mathfrak{l}}_i = (\ell_i, \pi + 1)$.

In order to be able to represent an arbitrary ideal class with a smooth ideal as in equation (5), we need to know the relation lattice and the group structure of $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$. This has been computed for the CSIDH-512 parameter set ($n = 74$ and $\ell_n = 587$) in [3]. This makes it possible, for the CSIDH-512 parameter set, to efficiently evaluate the action of arbitrary ideal classes. With current optimizations [18], these group actions can be evaluated in about 35 ms on a commercial CPU.

Security. While the CSIDH-512 parameter set is believed to provide 128 bits of classical security, it only offers at most 60 bits of security against quantum adversaries [4,21]. If in the future, due to progress in quantum computing technology, this is no longer sufficient, it is necessary to move to larger CSIDH parameter sets. Currently this is difficult, because for larger CSIDH parameter sets

it computationally expensive to compute the class group structure. Luckily, the ideal class group can be computed in quantum polynomial time, so switching to larger parameter sets should be possible well before the CSIDH-512 parameters are broken.

Protocol cost. As mentioned in Section 5, each of the n participating players has to evaluate a total of $T_{\mathcal{P}}(n, \lambda) = 3(1 + n\lambda)$ isogenies during the execution of CSI-RAShi, while the full runtime of the protocol, $T(n, \lambda) = 2 + \lambda + n(1 + 4\lambda)$, is slightly larger due to the sequentiality of the public key generation step. Taking the standard security parameter of $\lambda = 128$ and using the estimate of 35 ms to compute one isogeny, we find the following estimated runtimes for different numbers of players n.

n	2	4	8	16	128	1024
$T_{\mathcal{P}}(n)$	27 sec	54 sec	108 sec	3.6 min	29 min	3.8 hours
$T(n)$	40 sec	76 sec	148 sec	4.9 min	38 min	5.1 hours

We note an increase in cost of the full protocol of just below 18 seconds per player, the offset being 4.55 seconds. This makes the cost of the key generation step considerably lower than e.g. the cost of the distributed signature computation in the Sashimi protocol [8], which takes about five minutes per participating player.

7 Conclusion

In this work, we presented CSI-RAShi, a distributed key generation protocol based on Shamir secret sharing in the very hard homogeneous spaces setting. We introduced a primitive called *piecewise verifiable proof*, which allows parties to prove the existence of a single witness for multiple NP-statements, while allowing the verification of individual statements separately. We proved the security of this new primitive in the Quantum Random Oracle Model by using recent results on the quantum security of the Fiat-Shamir transform [13,27]. By basing our main protocol on a blueprint proposed by Gennaro et al. [15] and using standard zero-knowledge and piecewise verifiable proofs as subroutines, CSI-RAShi achieves robustness (i.e. the distributed key can be reconstructed even in the presence of malicious adversaries), and is actively secure. Since we can not benefit from Pedersen commitments in the very hard homogeneous spaces setting, we have to concede to a slightly weaker definition of the secrecy property, where the input to the simulator has to be chosen uniformly at random, instead of arbitrarily. Further, the computation of the public key has to be done in a round-robin way, where standard zero-knowledge proofs guarantee that the correct witness is used. The time complexity of the complete protocol scales linearly with the number of participants n and is independent of the threshold k. We instantiated the very hard homogeneous space with isogenies in the CSIDH-512 setting, using

the knowledge of the recently determined relation lattice for this parameter set [3]. In this setting, the total runtime of the protocol is approximately $4.5 + 18n$ seconds, where n is the number of participants in the protocol.

It is interesting to see if this cost can be further reduced, while keeping the protocol actively secure. Especially the public key computation step currently takes about twice as long as the secret sharing step and relies on a sequential round-robin structure with expensive zero-knowledge proofs. We also leave it as an open problem to prove that our protocol, or an adaptation thereof, is secure against adaptive corruptions.

In analogy to [10], the instantiation of our protocol with isogenies relies on the knowledge of the underlying class group and relation lattice. So far, this has only been computed for the CSIDH-512 parameter set, whose current security estimate is assumed to be below the NIST-1 level [4,21]. Computing these elements for higher level parameter sets currently seems out of reach.

Finally, we hope that adaptations of our piecewise verifiable proofs primitive will prove to be useful as building blocks in other cryptographic protocols.

References

1. Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.
2. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2020.
3. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019.
4. Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 493–522. Springer, 2020.
5. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.
6. Wouter Castryck, Jana Sotáková, and Frederik Vercauteren. Breaking the decisional Diffie-Hellman problem for class group actions using genus theory. *IACR Cryptol. ePrint Arch.*, 2020:151, 2020.
7. Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006:291, 2006.
8. Daniele Cozzo and Nigel P Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol. In *International Conference on Post-Quantum Cryptography*, pages 169–186. Springer, 2020.

9. 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.
10. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In *IACR International Conference on Public-Key Cryptography*, pages 187–212. Springer, 2020.
11. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *Conference on the Theory and Application of Cryptology*, pages 307–315. Springer, 1989.
12. Yvo Desmedt and Yair Frankel. Shared generation of authenticators and signatures. In *Annual International Cryptology Conference*, pages 457–469. Springer, 1991.
13. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *Annual International Cryptology Conference*, pages 356–383. Springer, 2019.
14. Rosario Gennaro, Stanisław Jarecki, Hugo Krawczyk, and Tal Rabin. Robust threshold DSS signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 354–371. Springer, 1996.
15. Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, 2007.
16. Lein Harn. Group-oriented (t, n) threshold digital signature scheme and digital multisignature. *IEE Proceedings-Computers and Digital Techniques*, 141(5):307–313, 1994.
17. Aniket Kate. Distributed key generation and its applications. 2010.
18. Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *International Conference on Cryptology in India*, pages 137–152. Springer, 2018.
19. National Institute of Standards and Technology. Threshold cryptography. 2020.
20. Torben Pryds Pedersen. A threshold cryptosystem without a trusted party. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 522–526. Springer, 1991.
21. Chris Peikert. He gives C-sieves on the CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 463–492. Springer, 2020.
22. Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85, 1989.
23. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptol. ePrint Arch.*, 2006:145, 2006.
24. Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
25. Victor Shoup. Practical threshold signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 207–220. Springer, 2000.
26. Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 166–195. Springer, 2016.
27. Dominique Unruh. Post-quantum security of Fiat-Shamir. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 65–95. Springer, 2017.

A Security proof of NIPVP

A.1 Completeness

Lemma 1. *Algorithms 3 and 4 constitute a complete NIPVP in the QROM for the list of relations of (4) if the used commitment scheme is collapsing and quantum computationally hiding.*

Proof. If the protocol is followed correctly and if the input was a valid statement-witness pair $(x, w) \in R$, then the verifier will accept the proof piece with probability 1.

- In the case $i = 0$ the verifier will accept the proofs because the curves $\widetilde{E}_j$ recomputed by the verifier match the curves $\hat{E}_j$ computed by the prover: for each $j \in \{1, \ldots, \lambda\}$, if $c_j = 0$, then $r_j = b_j$ and hence $\widetilde{E}_j = [r_j(0)]E_0 = [b_j(0)]E_0 = \hat{E}_j$. If $c_j = 1$, then $r_j(0) = b_j(0) - f(0)$, so again we have $\widetilde{E}_j = [r_j(0)]E_1 = [b_j(0) - f(0)][f(0)]E_0 = [b_j(0)]E_0 = \hat{E}_j$. Thus both C_0 are equal and the verifier will accept.
- In the case $i > 0$ for each $j \in \{1, \ldots, \lambda\}$, the prover computes $b_j(i)$, and the verifier computes $r_j(i) + c_j x_j = b_j(i) - c_j f(i) + c_j x_j = b_j(i)$, if $x_i = f(i)$. So if the witness is valid, then the C_i match and the verifier will accept. □

A.2 Soundness

Our protocol can be seen as a "weak" Fiat-Shamir transformed version of a sigma protocol, where by "weak" we mean that, to obtain a challenge we only hash the commitment (instead of hashing both the commitment and the statement). The known results on the security of the FS transform in the QROM are about the strong FS transform. Therefore, before we can prove the soundness of our protocol we first prove the following lemma, which allows us to prove the security of the weak FS transform. This lemma bootstraps the known results for the strong FS transform to prove the soundness of the weak FS transform of a sigma protocol where the first message of the prover commits to the statement.

Lemma 2. *Suppose $\Sigma = (P_1, V_1, P_2, V_2)$ is a sigma protocol for the relation R with superpolynomially sized challenge space $\mathcal{C}h$, special soundness and quantum computationally unique responses. Let $\Sigma' = (P', V')$ be the following sigma protocol:*

$$\begin{aligned} P'_1(x, w) &: y \leftarrow \{0,1\}^\lambda, \mathsf{C}_x \leftarrow \mathcal{C}(x, y), com \leftarrow P_1(x, w), \\ &\quad com' = (\mathsf{C}_x, com) \\ V'_1(com') &: ch \leftarrow \mathcal{C}h \\ P'_2(ch) &: rsp \leftarrow P_2(ch), rsp' \leftarrow (x, y, rsp) \\ V'_2(x, com', ch, rsp') &: \textit{accept if } \mathsf{C}_x = \mathcal{C}(x, y) \textit{ and } V_2(x, com, ch, rsp) = 1 \end{aligned}$$

Then the weak Fiat-Shamir transformed version of Σ' is a quantum proof of knowledge for the same relation R, assuming that $\mathcal{C}$ is collapsing.

Proof. The strategy of the proof is to interpret the weak FS transform for the relation R as the strong FS transformed protocol for a different relation R'. We can then use the techinques of Don et al. [13] on the security of the strong FS stransform in the QROM.

We define the following relation

$$R' = \{(\mathsf{C}_x, (x, y, w)) \,|\, \mathsf{C}_x = \mathcal{C}(x, y) \text{ and } (x, w) \in R\}\,,$$

and the following sigma protocol $\Sigma'' = (P'', V'')$:

$$\begin{aligned} P_1''(\mathsf{C}_x, (x, y, w)) &: com \leftarrow P_1(x, w) \\ V_1''(com) &: ch \leftarrow \mathcal{C}h \\ P_2''(ch) &: rsp'' \leftarrow (x, y, P_2(ch)) \\ V_2''(\mathsf{C}_x, com, ch, rsp'') &: \text{accept if } \mathsf{C}_x = \mathcal{C}(x, y) \text{ and } V_2(x, com, ch, rsp) = 1 \end{aligned}$$

Observe that the adaptive proof of knowledge game against the weak FS transform of Σ' is identical to the adaptive proof of knowledge game against the strong FS transform of Σ'', so it suffices to prove that $FS(\Sigma'')$ is a quantum proof of knowledge to finish the proof. We will do this by invoking the theorems of Don et al., which say that if Σ'' has special soundness, quantum computationally unique responses and a superpolynomial challenge space, then $FS(\Sigma'')$ is a quantum proof of knowledge (Combination of Theorem 25 and Corollary 16 of [13]).

Superpolynomial challenge space. The challenge space $\mathcal{C}h$ is superpolynomial by assumption.

Special soundness.[4] Suppose we are given two accepting transcripts $\mathsf{C}_x, com, ch, (x, y, rsp)$ and $\mathsf{C}_x, com, ch', (x', y', rsp')$ with $ch \neq ch'$. This means that

$$\begin{gathered} \mathsf{C}_x = \mathcal{C}(x, y) = \mathcal{C}(x', y')\,, \text{ and} \\ V_2(x, com, ch, rsp) = V_2(x', com, ch', rsp') = \text{accept}\,. \end{gathered}$$

[4] Our extractor is not guaranteed to output a witness, instead it is allowed to output a collision in $\mathcal{C}$. This means that the extractor for the FS transformed protocol could also output a collision for $\mathcal{C}$ instead of outputting a witness. This is not a problem, because $\mathcal{C}$ is assumed to be collapsing, which implies that an efficient adversary can only output collisions with negligible probability. [26]

Then, we can either extract a collision $\mathcal{C}(x,y) = \mathcal{C}(x',y')$ for $\mathcal{C}$ in the case $x \neq x'$, or otherwise we can invoke the special soundness of Σ to obtain a witness w such that $(x,w) \in R$, which means we can construct a witness $w' = (x,y,w)$ such that $(\mathsf{C}_x, w') \in R'$.

Quantum computationally unique responses (Definition 24 of [13]). We define 3 games Game_i for $i \in \{1,2,3\}$, played by a two-stage poly-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

$$\begin{aligned}
\mathsf{Game}_i^{\mathcal{A}}() : \quad & (x,y,rsp), com, ch \leftarrow \mathcal{A}_1() \\
& z \leftarrow V_2(x, com, ch, rsp) \text{ and } \mathsf{C}_x = \mathcal{C}(x,y) \\
\text{if } i \in \{1,2\} \quad & rsp \leftarrow \mathcal{M}(rsp) \\
\text{if } i \in \{1\} \quad & (x,y) \leftarrow \mathcal{M}(x,y) \\
& (com, ch) \leftarrow \mathcal{M}(com, ch) \\
& b \leftarrow \mathcal{A}_2(x,y,rsp,com,ch)
\end{aligned}$$

Then the sigma protocol has quantum computationally unique responses if for any adversary $\mathcal{A}$, the following advantage is a negligible function of the security parameter:

$$Adv = \left| \Pr_{\mathsf{Game}_1^{\mathcal{A}}}[z = b = 1] - \Pr_{\mathsf{Game}_3^{\mathcal{A}}}[z = b = 1] \right| . \tag{6}$$

This follows immediately from the assumptions, because the assumption that $\mathcal{C}$ is collapsing implies that $\left|\Pr_{\mathsf{Game}_1^{\mathcal{A}}}[z = b = 1] - \Pr_{\mathsf{Game}_2^{\mathcal{A}}}[z = b = 1]\right|$ is negligible, and the assumption that Σ has quantum computationally unique responses implies that $\left|\Pr_{\mathsf{Game}_2^{\mathcal{A}}}[z = b = 1] - \Pr_{\mathsf{Game}_3^{\mathcal{A}}}[z = b = 1]\right|$ is negligible. □

Lemma 3. *Algorithms 3 and 4 constitute a sound NIPVP in the QROM for the list of relations of (4) if the used commitment scheme is collapsing.*

Proof. We need to prove that for any $I \subset \{0,\ldots,n\}$ and any poly-time quantum adversary $\mathcal{A}^{\mathcal{O}}$, the following advantage is negligible:

$$\mathsf{Adv}^{\mathsf{sound}}_{\mathcal{A},I}(\lambda) = \Pr\left[\begin{matrix} \forall i \in I : V^{\mathcal{O}}(i, x_i, \widetilde{\pi}, \pi_i) = 1 \\ \nexists w : (x_I, w) \in R_I \end{matrix} \middle| (x_I, \pi_I) \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda) \right] .$$

If $|I| < k$, then $\mathsf{Adv}^{\mathsf{sound}}_{\mathcal{A},I} = 0$ for any $\mathcal{A}$, simply because for every x_I, there exists a $w \in \mathbb{Z}_N[x]_{\leq k-1}$ such that $(x_I, w) \in R_I$. Therefore, we can focus on the case $|I| \geq k$ for the remainder of the proof. We fix $I \subset \{0,\ldots,n\}$ with $|I| \geq k$.

We define the function F as follows

$$\begin{aligned} F : \{0,1\}^\lambda \times \mathcal{X}_I \times (\{0,1\}^\lambda)^I \times (\mathbb{Z}_N[x]_{\leq k-1})^\lambda &\to (\{0,1\}^{2\lambda})^I \\ (\mathbf{c}, x_I, y_I, \mathbf{r}) &\mapsto \{\mathsf{C}'_i\}_{i\in I}\,, \end{aligned}$$

where $\mathsf{C}'_0 = \mathcal{C}([r_1(0)]E_{c_1}||\cdots|||[r_\lambda(0)]E_{c_\lambda}||x_0, y_0)$ (if $0 \in I$), and where $\mathsf{C}'_i = \mathcal{C}(r_1(i) + c_1 x_i||\cdots||r_\lambda(i) + c_\lambda x_i||x_i, y_i)$.

With this notation we have $V^{\mathcal{O}}(i, x_i, \widetilde{\pi}, \pi_i) = 1$ for all $i \in I$ if and only if $F(\mathcal{O}(\mathsf{C}), x_I, y_I, \mathbf{r}) = \mathsf{C}_I$ and $\mathsf{C}'_i = \mathcal{C}(x_i, y_i)$ for all $i \in I$. So the claim that the advantage $\mathsf{Adv}^{\mathsf{sound}}_{\mathcal{A},I}(\lambda)$ is negligible for every efficient adversary $\mathcal{A}$ is equivalent to the claim that the "weak" FS transform of the following sigma protocol $\Sigma' = (P'_1, V'_1, P'_2, V'_2)$ is a Quantum computationally sound proof for R_I (Definition 9 of [13]):

$$\begin{aligned} P'_1(x_I, w) &: y_I, y'_I \leftarrow (\{0,1\}^\lambda)^I, \mathsf{C}'_i \leftarrow \mathcal{C}(x_i, y'_i) \text{ for all } i \in I, \\ &\quad \mathbf{b} \leftarrow (\mathbb{Z}_N[x]_{\leq k-1})^\lambda, \mathsf{C}_I = F(0, x_I, y_I, \mathbf{b}) \\ V'_1(\mathsf{C}_I, \mathsf{C}'_I) &: \mathbf{c} \leftarrow \{0,1\}^\lambda \\ P'_2(\mathbf{c}) &: \mathbf{r} \leftarrow \mathbf{b} - \mathbf{c}\cdot w, rsp \leftarrow (y_I, y'_I, \mathbf{r}) \\ V'_2(rsp) &: \text{ accept if } \mathsf{C}_I = F(\mathbf{c}, x_I, y_I, \mathbf{r}) \text{ and } \mathsf{C}'_i = \mathcal{C}(x_i, y_i) \text{ for all } i \in I \end{aligned}$$

Since quantum computational soundness is implied by the quantum proof of knowledge property, it suffices to prove that the weak FS transform of Σ' is a quantum proof of knowledge. This sigma protocol takes the form of Σ' in the statment of Lemma 2, so we can conclude that our NIPVP is sound if the sigma protocol $\Sigma = (P_1, V_1, P_2, V_2)$ with

$$\begin{aligned} P_1(x_I, w) &: y_I \leftarrow (\{0,1\}^\lambda)^I, \mathbf{b} \leftarrow (\mathbb{Z}_N[x]_{\leq k-1})^\lambda, \mathsf{C}_I = F(0, x_I, y_I, \mathbf{b}) \\ V_1(\mathsf{C}_I) &: \mathbf{c} \leftarrow \{0,1\}^\lambda \\ P_2(\mathbf{c}) &: \mathbf{r} \leftarrow \mathbf{b} - \mathbf{c}\cdot w, rsp \leftarrow (\mathbf{r}, y_I) \\ V_2(rsp) &: \text{ accept if } \mathsf{C}_I = F(\mathbf{c}, x_I, y_I, \mathbf{r}) \end{aligned}$$

has a superpolynomial challenge space, special soundness and quantum computationally unique responses.

Superpolynomial challenge space. The size of the challenge space is 2^λ, which is superpolynomial in λ.

Special soundness[5]**.** Let $x_I, \mathsf{C}_I, \mathbf{c}, \mathbf{r}$ and $x_I, \mathsf{C}_I, \mathbf{c}'', \mathbf{r}'$ be two accepting transcripts with $\mathbf{c} \neq \mathbf{c}'$. Take $j \in \{1, \ldots, \lambda\}$ such that $c_j \neq c'_j$, without loss of gener-

[5] Similar to the proof of Lemma 2, our special soundness extractor outputs either a witness w such that $(x, w) \in R$, or a collision for $\mathcal{C}$. Since $\mathcal{C}$ is collision resistant,

ality we can assume $c_j = 0$ and $c'_j = 1$. Then, if $0 \in I$ and $[r_j(0)]E_0 \neq [r'(0)]E_1$ then we found a collision in $\mathcal{C}$. Similarly, if for some non-zero $i \in I$ we have $r_j(i) \neq r'_j(i) + x_i$ then we also have a collosion for $\mathcal{C}$. If there is no collision, it means that

$$\begin{aligned} r_j(i) &= r'_j(i) + x_i \text{ for all } i \in I, i > 0\,, \text{ and} \\ [r_j(0)]E_0 &= [r'(0)]E_1 \quad (\text{if } 0 \in I)\,, \end{aligned}$$

so $r_j(x) - r'_j(x)$ is a witness for x_I.

Quantum computationally unique responses. Notice that F factors as $F = G \circ H$, where H is the function that given $\mathbf{c}, x_I$ and $\mathbf{r}$ computes the input to the commitment function $\mathcal{C}$, and where G takes the output of H and the commitment randomness y_I and outputs C_I. We define 3 games Game_i for $i \in \{1, 2, 3\}$, played by a two-stage poly-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

$$\begin{aligned} \mathsf{Game}_i^{\mathcal{A}}() : \quad & (y_I, \mathbf{r}), (\mathsf{C}_i, \mathbf{c}) \leftarrow \mathcal{A}_1() \\ & z \leftarrow 1 \text{ if } \mathsf{C}_I = F(\mathbf{c}, x_I, y_I, \mathbf{r}) \text{ and } 0 \text{ otherwise.} \\ \text{if } i = 3 \quad & \mathbf{r} \leftarrow \mathcal{M}(\mathbf{r}) \\ \text{if } i \in \{2, 3\} \quad & (h, y_I) \leftarrow \mathcal{M}(H(\mathbf{c}, x_I, \mathbf{r}), y_I) \\ & (\mathsf{C}_I, \mathbf{c}) \leftarrow \mathcal{M}(\mathsf{C}_I, \mathbf{c}) \\ & b \leftarrow \mathcal{A}_2(x, y, rsp, com, ch) \end{aligned}$$

We need to prove that for any efficient $\mathcal{A}$ the following is a negligible function:

$$\left| \Pr_{\mathsf{Game}_1^{\mathcal{A}}}[z = b = 1] - \Pr_{\mathsf{Game}_3^{\mathcal{A}}}[z = b = 1] \right| .$$

Since G is just the parallel composition of $|I|$ instances of $\mathcal{C}$, and since we assumed that $\mathcal{C}$ is collapsing it follows that G is collapsing. Therefore we have that $\left|\Pr_{\mathsf{Game}_1^{\mathcal{A}}}[z = b = 1] - \Pr_{\mathsf{Game}_2^{\mathcal{A}}}[z = b = 1]\right|$ is negligible. Since for a fixed value of x_I and $\mathbf{c}$, the function $H(\mathbf{c}, x_I, \cdot)$ is injective (here we use that $|I| \geq k$), we get that after measuring h and $\mathbf{c}$, the register $\mathbf{r}$ is not in a superposition of basis vectors. Therefore, the measurement $\mathbf{r} \leftarrow \mathcal{M}(\mathbf{r})$ does not affect the state of the system and we have $\Pr_{\mathsf{Game}_2^{\mathcal{A}}}[z = b = 1] = \Pr_{\mathsf{Game}_3^{\mathcal{A}}}[z = b = 1]$. □

A.3 Zero-knowledge

For a fixed $I \subset \{0, \ldots, n\}$, our security definition of zero-knowledge for NIPVP is similar to the standard definition of non-interactive zero-knowledge in the

this is not a problem, because the PoK extractor can only output a collision with negligible probability.

QROM (e.g. definition 6 of [27]), except that the simulator is only given a partial statement x_I, instead of the full statement $\mathbf{x}$. Our proof strategy is to first reduce the NIPVP soundness to the standard zero-knowledge property of a standard sigma protocol. Then, we can use the results of Unruh [27] to finish the proof.

Lemma 4. *Fix $I \subset \{0,\dots,n\}$ and suppose $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ is a zero-knowledge simulator for the "weak" FS transform of the sigma protocol $\Sigma = (P_1, V_1, P_2, V_2)$ for the relation R_I.*

$$\begin{aligned}
P_1(x_I, w) &: y_I, y'_I \leftarrow (\{0,1\}^\lambda)^I, \mathsf{C}'_i \leftarrow \mathcal{C}(x_i, y'_i) \text{ for all } i \in I, \\
&\quad \mathbf{b} \leftarrow (\mathbb{Z}_N[x]_{\leq k-1})^\lambda, \mathsf{C}_I = F(0, x_I, y_I, \mathbf{b}) \\
V_1(\mathsf{C}_I, \mathsf{C}'_I) &: \mathbf{c} \leftarrow \{0,1\}^\lambda \\
P_2(\mathbf{c}) &: \mathbf{r} \leftarrow \mathbf{b} - \mathbf{c} \cdot w, rsp \leftarrow (y_I, y'_I, \mathbf{r}) \\
V_2(rsp) &: \text{ accept if } \mathsf{C}_I = F(\mathbf{c}, x_I, y_I, \mathbf{r}) \text{ and } \mathsf{C}'_i = \mathcal{C}(x_i, y_i) \text{ for all } i \in I\,.
\end{aligned}$$

Then there exists a simulator $\mathsf{Sim}' = (\mathsf{Sim}'_1, \mathsf{Sim}'_2)$ such that for any poly-time quantum distinguisher $\mathcal{A}$ the distinguishing advantage

$$\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{Sim},\mathcal{A}} = \left| \Pr\left[\mathcal{A}^{P',\mathcal{O}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{S',\mathsf{Sim}'_2}(1^\lambda) = 1\right] \right|,$$

is a negligible function of the security parameter, where P' is an oracle that on input $(\mathbf{x}, w) \in R$ runs $\pi := P^{\mathcal{O}}(\mathbf{x}, w)$ and outputs $\pi_I = (\tilde{\pi}, \{\pi_i\}_{i\in I})$ and S' is an oracle that on input $(\mathbf{x}, w) \in R$ returns $\mathsf{Sim}'_1(\{x_i\}_{i\in I})$.

Proof. The simulator Sim'_2 simply forwards all its queries to Sim_2, and Sim'_1 forwards his queries to Sim_1 to obtain $\mathsf{C}_I, y_I, y'_I, \mathbf{r}$. Then, for all $i \notin I$ the simulator Sim'_1 commits to dummy values to produce C_i and C'_i. Then Sim'_1 outputs $\tilde{\pi} = (\mathsf{C}, \mathsf{C}', \mathbf{r}), \{\pi_i = (y_i, y'_i)\}_{i\in I}$.

We prove the lemma with a simple hybrid argument: Let Sim'' be identical to Sim' except that it interacts with a real prover for Σ, instead of with Sim. Then, because Sim is supposed to be computationally indistinguishable from a real prover, we have that $\left|\Pr\left[\mathcal{A}^{S',\mathsf{Sim}'_2}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{S'',\mathsf{Sim}''_2}(1^\lambda) = 1\right]\right|$ is negligible. Secondly, since the only difference between an honest prover for the NIPVP protocol and Sim'' is that Sim'' commits to dummy values instead of real values for $i \notin I$, it follows from the quantum computationally hiding property of the commitment scheme that $\left|\Pr\left[\mathcal{A}^{P',\mathcal{O}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{S'',\mathsf{Sim}''_2}(1^\lambda) = 1\right]\right|$ is negligible. □

Lemma 5. *Algorithms 3 and 4 form a zero-knowledge NIPVP in the QROM for the list of relations of (4) if the used commitment scheme is quantum computationally hiding and collapsing.*

Proof. In light of Lemma 4, if suffices to prove that for every $I \subset \{0, \ldots, n\}$, the "weak" FS transform of the sigma protocol Σ_I is zero-knowledge. Unruh proved (Theorem 20 of [27]) that if a sigma protocol has HVZK, completeness and unpredictable commitments, then the "strong" FS transform of that sigma protocol is zero-knowledge, but the proof goes through without problems in case of a "weak" FS transform also. Therefore, it suffices to prove for each $I \subset \{0, \ldots, n\}$, that Σ_I has HVZK, completeness and unpredictable commitments.

Completeness. The protocol has perfect completeness. The proof is similar to the proof of Lemma 1.

Unpredictable commitments. We say the sigma protocol has unpredictable commitments if the commitments have superlogarithmic collision entropy. More concretely, if there exists a negligible function $\mu(\lambda)$, such that for every $(x_I, w) \in R_I$ we have

$$\Pr[(\mathsf{C}_I, \mathsf{C}'_I) = (\mathsf{C}''_I, \mathsf{C}'''_I) | (\mathsf{C}_I, \mathsf{C}'_I) \leftarrow P_1(x_I, w), (\mathsf{C}''_I, \mathsf{C}'''_I) \leftarrow P_1(x_I, w)] \leq \mu(\lambda)\,.$$

Let $i \in I$, then since C_i and C''_i are commitments, there are two possible ways to get a collision:

- The first possibility is that both commit to the same value. But since C_i commits to λ uniformly random elements of $\mathcal{G}$ (or $\mathcal{E}$ in case $i = 0$), the probability that this happens is negligible.
- The second possibility is that both commitments commit to different values. Since we assume that $\mathcal{C}$ is collapsing (which implies collision resitance), this can also only happen with negligible probability.

Honest Verifier Zero Knowledge. The protocol has perfect HVZK. Consider the simulator that picks $y_I, y'_I, \mathbf{r}$ and $\mathbf{c}$ uniformly at random and sets $\mathsf{C}_I = F(\mathbf{c}, x_I, y_I, \mathbf{r})$ and $\mathsf{C}'_i = \mathcal{C}(x_i, y'_i)$ for all $i \in I$.

This produces the same distribution of transcripts as honest executions of the protocol, because in both cases $y_I.y'_I, \mathbf{r}$ and $\mathbf{c}$ are uniformly random, and the rest of the transcript is a function of $y_I, y'_I, \mathbf{r}$ and $\mathbf{c}$.

□