Compressing Proofs of k-Out-Of-n Partial Knowledge

Thomas Attema[1,2,3,⋆], Ronald Cramer[1,2,⋆⋆], and Serge Fehr[1,2,⋆⋆⋆]

[1] CWI, Cryptology Group, Amsterdam, The Netherlands
[2] Leiden University, Mathematical Institute, Leiden, The Netherlands
[3] TNO, Cyber Security and Robustness, The Hague, The Netherlands

Version 2 - July 16, 2020[4]

Abstract. In a zero-knowledge (ZK) proof of partial knowledge, introduced by Cramer, Damgård and Schoenmakers (CRYPTO 1994), a prover claiming knowledge of witnesses for some k-subset of n given public statements can convince the verifier without revealing which k-subset. The accompanying dedicated solution based on secret sharing achieves linear communication complexity for general k, n and for many natural classes of statements. Especially the case $k = 1$ and $n = 2$ ("one-out-of-two") has seen myriad applications during the last decades, e.g., in electronic voting, ring signatures, and confidential transaction systems in general.

In this paper we focus on the discrete logarithm (DL) setting; the prover's claim pertains to knowledge of discrete logarithms of k-out-of-n given elements from a group supporting DL-based cryptography. Groth and Kohlweiss (EUROCRYPT 2015) have shown how to solve the special case $k = 1$ and general n with *logarithmic* communication instead of linear. However, their method, which is original, takes explicit advantage of $k = 1$ and does not generalize to $k > 1$ without losing all advantage over prior work.

Our contributions are as follows. We show a solution with logarithmic communication *for general* k, n instead of just $k = 1$ and general n from prior work. Applying the Fiat-Shamir transform renders a non-interactive logarithmic-size zero-knowledge proof. Our approach deploys a novel twist on a basic primitive from Compressed Σ-Protocol Theory (CRYPTO 2020) that we then utilize to compress a carefully chosen adaptation of the CRYPTO 1994 approach down to logarithmic size. Interestingly, *even for* $k = 1$ *and general* n our approach improves prior work as it reduces communication up to almost a factor 1/2.

We also generalize this to proofs of partial knowledge about compact commitments of long vectors. Optionally, the prover may at the same time demonstrate his secret to satisfy some arbitrary given constraint. Finally, we also generalize from threshold to arbitrary access structures.

Keywords: Proofs of Partial Knowledge, One-out-of-Many, Compressed Σ-Protocol Theory, Zero-Knowledge, Secure Algorithmics, Ring-Signatures.

1 Introduction

Recently, compressed Σ-protocol theory [AC20b] was introduced as a strengthening of Σ-protocol theory. It inherits the flexibility and versatility of Σ-protocols while compressing their communication complexity from linear to logarithmic. The main pivot of this theory is a standard Σ-protocol for opening linear forms on Pedersen vector commitments, i.e., a Σ-protocol for proving that a committed vector $\mathbf{x}$ satisfies $L(\mathbf{x}) = y$ for a public linear form L and a public scalar y. By an appropriate adaptation of the techniques from [BCC$^+$16, BBB$^+$18] this pivotal Σ-protocol is compressed to achieve communication complexity that is logarithmic in the dimension of $\mathbf{x}$. Additionally a linearization approach to handle non-linearities is described. As one of

⋆ thomas.attema@tno.nl
⋆⋆ cramer@cwi.nl, cramer@math.leidenuniv.nl
⋆⋆⋆ serge.fehr@cwi.nl

[4] **Change log** w.r.t. Version 1 - June 19, 2020: (a) minor editorial changes throughout, (b) elaborated on the alternative approach of constructing proofs of partial knowledge *indirectly* via circuit ZK protocols, and (c) removed the exact knowledge errors in several theorems (see [AC20a] for a discussion on knowledge errors).

the applications of this theory it was shown how to obtain circuit zero-knowledge protocols with logarithmic communication complexity for arbitrary arithmetic circuits.

In this work, we consider another application, namely *proofs of partial knowledge* [CDS94]. These allow a prover to convince a verifier, in zero-knowledge, to know k-out-of-n secrets, in particular without revealing which secrets the prover knows. Typically, these secrets are solutions to public instances of intractable problems, such as the discrete logarithm problem.

In [CDS94], proofs of partial knowledge were introduced and a generic solution, that fits seamlessly with Σ-protocol theory, was given. Their work requires two main ingredients. First, a Σ-protocol Π for proving knowledge of one secret. Protocol Π is assumed to be perfectly complete, special sound, special honest verifier zero-knowledge (SHVZK) and public-coin. Second, the protocol requires a linear n-party threshold secret sharing scheme with $(n-k)$-privacy, i.e., an $(n-k+1, n)$-threshold secret sharing scheme (see, e.g., [CDN15]). For simplicity we only consider Shamir's secret sharing scheme [Sha79] in the remainder of this section.

We now describe the approach of [CDS94] for proving knowledge of k-out-of-n secrets. More precisely, we consider a public set of n problem instances and a secret subset $S \subset \{1, \ldots, n\}$ with $|S| = k$ such that the prover knows the solutions (secrets) for all problem instances $i \in S$. The prover and verifier run n parallel instantiations of Σ-protocol Π, while the secret sharing scheme allows the prover to use simulated transcripts for at most $(n-k)$-out-of-n problem instances. More precisely, the protocol goes as follows.

1. First, the prover simulates $n-k$ accepting transcripts (a_i, c_i, r_i) for $i \notin S$. Note that a simulator exists since Σ-protocol Π is assumed to be SHVZK. Second, the prover computes first messages a_i of honest executions of Σ-protocol Π for problem instances $i \in S$. The prover sends $(a_1, \ldots, a_n)$ to the verifier.
2. The verifier sends a challenge c, sampled uniformly at random, to the prover.
3. First, the prover computes c_i for $i \in S$ such that $(c_1, \ldots, c_n)$ is an $(n-k+1, n)$-secret sharing of c. Note that c_i for $i \notin S$ were already fixed in Step 1. Second, the prover computes, given first messages a_i and challenges c_i, the accepting responses r_i for $i \in S$. The prover sends $(c_1, \ldots, c_n)$ and $(r_1, \ldots, r_n)$ to the verifier.
4. The verifier checks that (a_i, c_i, r_i) is an accepting transcript for all $1 \leq i \leq n$, and that $(c_1, \ldots, c_n)$ is an $(n-k+1, n)$-secret sharing of c.

The k-out-of-n proof of partial knowledge is a Σ-protocol with communication complexity that is linear in the number of statements n. Completeness follows since for any challenge c the elements $(c_i)_{i \notin S}$ can be (uniquely) completed to an $(n-k+1, n)$-secret sharing $(c_1, \ldots, c_n)$ of c.

Special soundness follows by the special soundness of Π and since any two $(n-k+1, n)$-secret sharings $(c_1, \ldots, c_n)$ and $(c'_1, \ldots, c'_n)$ of $c \neq c'$ differ in at least k coefficients. SHVZK follows since Π is public coin, i.e., the verifier's messages are all sampled uniformly at random. Hence, the simulated challenges $(c_i)_{i \notin S}$ are distributed uniformly and the distribution of $(c_1, \ldots, c_n)$ is independent from the set S.

A straightforward generalization shows that this approach applies to the scenario where Π is any public coin SHVZK proof of knowledge (PoK), possibly with more than 3 moves. Moreover, for simplicity we have restricted ourselves to a threshold access structure containing all subsets of $\{1, \ldots, n\}$ with cardinality at least k. This approach generalizes to arbitrary access structures. For details we refer to [CDS94].

Finally we note that, it is possible to formulate a proof of partial knowledge as a circuit satisfiability problem and solve it with known circuit ZK techniques. However, as this requires the commitment scheme to be implemented as an arithmetic circuit, this will lead to a more cumbersome approach and at the very least a considerable overhead. Therefore it is interesting to consider *direct* approaches that avoid these circuit formulations. In contrast, the related problem of proving knowledge of n commitment openings of which k are equal to 0 can easily be solved with known circuit ZK techniques. However, these techniques do not apply here since the prover does not necessarily know all the commitment openings.

1.1 Contributions

In this work, we consider a novel twist on a basic protocol from Compressed Σ-protocol Theory [AC20b]. Namely, we observe that the compressed Σ-protocol for opening linear forms can be adapted to *open arbitrary*

homomorphisms, i.e., proving that a committed vector $\mathbf{x} \in \mathbb{Z}_q^n$ satisfies $f(\mathbf{x}) = y$ for a group homomorphism $f : \mathbb{Z}_q^n \to \mathbb{G}$ and an element $y \in \mathbb{G}$. The loss of efficiency is at most a constant factor and the adapted protocol still achieves a logarithmic communication complexity. This generalized functionality has not been considered before. However, in the present context of proofs of partial knowledge, it turns out to be rather useful.

As a warm-up, we argue that this allows a prover to prove, in zero-knowledge, that it knows of n-out-of-n discrete logarithms, i.e., to prove knowledge of x_i such that $g^{x_i} = P_i$ for all $1 \leq i \leq n$, with *logarithmic communication complexity*. Namely, by having the prover commit to the vector $\mathbf{x} = (x_1, \ldots, x_n)$ of exponents and open the homomorphisms $\phi_i(\mathbf{x}) = g^{x_i}$ for $1 \leq i \leq n$. These homomorphisms evaluate to P_i if and only if the committed coefficients are equal to the DLs of the P_i's. The amortization technique to open multiple linear forms for essentially the price of one [AC20b] directly applies to opening multiple homomorphisms, thereby we achieve a logarithmic communication. We emphasize that this is merely a warm-up. Standard amortization techniques namely solve the n-out-of-n case with a *constant* communication complexity.

Continuing our warm-up, let us now consider the scenario where the prover wants to show, in zero-knowledge, that it knows k-out-of-n DLs, i.e., the prover claims to know a subset $S \subset \{1, \ldots, n\}$ of cardinality k and exponents $x_i \in \mathbb{Z}_q$ such that $g^{x_i} = P_i$ for all $i \in S$. We reduce this k-out-of-n case to the n-out-of-n case by having the prover "eliminate" the exponents that it does not know. To this end the prover uses a vector $(s_1, \ldots, s_n)$ such that $s_i = 0$ for all $i \notin S$ and commits to the vector $(\mathbf{s}, \mathbf{y}) = (s_1, \ldots, s_n, s_1 x_1, \ldots, s_n x_n) \in \mathbb{Z}_q^{2n}$, where $y_i = s_i x_i$ is understood to be equal to 0 for $i \notin S$. Then the prover shows that it can open the homomorphisms $\psi_i(\mathbf{s}, \mathbf{y}) = g^{-y_i} P_i^{s_i}$ to 1 for all $1 \leq i \leq n$. From this it follows that the prover knows the DL of $P_i^{s_i}$ for all i. Moreover, the prover has demonstrated knowledge of the DLs of the P_i's for which $s_i \neq 0$. What remains is for the prover to show that the vector $\mathbf{s}$ contains at most $n - k$ zeros. This can be proven by a direct applications of the circuit ZK techniques from compressed Σ-protocol theory.

However, we follow a slightly different and more efficient approach by applying a carefully chosen adaptation of the proofs of partial knowledge from [CDS94]. Namely, instead of letting $(s_1, \ldots, s_n)$ be any vector with at most $n - k$ zeros we let it be a Shamir secret sharing, with $n - k$ privacy, of 1 of the prover's choice. Note that, in this case, the prover can choose $s_i = 0$ for any set of at most $n - k$ indices. Such a secret sharing is uniquely defined by a polynomial $p(X) = 1 + \sum_{i=1}^{n-k} a_i X^i$ of degree at most $n - k$. Instead of committing to the vector $(\mathbf{s}, \mathbf{y})$ the prover commits to $(a_1, \ldots, a_{n-k}, \mathbf{y}) \in \mathbb{Z}_q^{2n-k}$ and defines the shares s_i as the appropriate affine combinations of the coefficients committed to. Correctness of the vector $(s_1, \ldots, s_n)$ now automatically follows and amortization over the n homomorphisms applies as before, thereby again achieving a logarithmic communication complexity.

Altogether, we construct the first *direct*[5] proof of k-out-of-n partial knowledge with logarithmic communication. Additionally, our protocol improves the communication costs of previous proofs of partial knowledge that exist for the special case of $k = 1$ [GK15, BCC$^+$15, Dia20, JM20]. Furthermore, we would like to emphasize the simplicity of our construction.

Our protocol requires the prover to send exactly $4\lceil \log_2(2n - k + 1) \rceil - 5$ group elements and 4 field elements to the verifier. In contrast, the 1-out-of-n proof of [GK15] requires the prover to send $4\lceil \log_2(n) \rceil$ group elements and $3\lceil \log_2(n) \rceil + 1$ field elements. Hence, besides generalizing their results, we reduce the communication costs from roughly $7 \log_2(n)$ elements to roughly $4 \log_2(n)$ elements. Moreover, the protocol is interactive and can be made non-interactive by applying the Fiat-Shamir transformation [FS86].

Additionally, we show that our protocols generalize to proofs of partial knowledge about "multi-generator discrete logarithms" and corresponding Pedersen vector commitments with a logarithmic communication complexity. Moreover, we show that our proofs of partial knowledge are compatible with circuit ZK protocols of [AC20b], allowing the prover to demonstrate that his secret information satisfies some arbitrary given constraint. Finally, we generalize the results from threshold access structures to arbitrary access structures.

[5] Informally, we say that the approach is *indirect* if it formulates a proof of partial knowledge as a circuit satisfiability problem, thereby implementing the modular exponentiation function as an arithmetic circuit.

1.2 Related Work

The introduction of k-out-of-n proofs of partial knowledge was accompanied by a generic construction resulting in a communication complexity that is linear in n [CDS94]. Their protocol is interactive and relies solely on the existence of a public coin honest-verifier zero-knowledge PoK for proving knowledge of a single secret, i.e., a 1-out-of-1 proof of (partial) knowledge.

The proofs of partial knowledge were shown to be applicable to the construction of group signature schemes [Cam97]. Group signature schemes [CvH91] allow a member of a group to sign a message without revealing which member it is. A designated trusted third party, acting as a group manager, is capable of revoking the anonymity of the signer.

In contrast, ring signature schemes [RST01] do not contain such a revocation mechanism. In a ring signature scheme a group member can select any ad-hoc subset of group members and anonymously sign a message on behalf of this subset. Because of this ad-hoc nature a ring signature must contain a list of the subset's members and, therefore, its size grows linearly in the size of the ring. However, in many practical scenario's the costs of specifying a ring can be amortized over many instances. The proofs of partial knowledge of [CDS94], or more specifically their 1-out-of-n proofs, together with the Fiat-Shamir [FS86] heuristic allow for a straightforward construction of ring-signature schemes. The schemes of [RST01] follow a more efficient approach, which relies on the use of trapdoor one-way functions.

A problem related to proofs of partial knowledge is the set-membership problem, where a prover claims that it can open a commitment P to an element x in a public set S. In [BG13], a zero-knowledge protocol for proving set-membership, with logarithmic communication complexity, was introduced. Their approach is based on the discrete logarithm assumption and uses an efficient zero-knowledge protocol for polynomial evaluation.

Groth and Kohlweiss [GK15] consider a prover who claims to be able to open at least 1-out-of-n public commitments to zero. Their solution is a Σ-protocol that works for any additively homomorphic commitment scheme over $\mathbb{Z}_q$ and it achieves a logarithmic communication complexity. To describe their approach, let $1 \leq \ell \leq n$ be the index of the prover's secret. The prover commits to each bit of ℓ and runs $\lceil \log_2(n) \rceil$ standard Σ-protocols, in parallel and on a common challenge, proving that all these commitments can indeed be opened to a binary value. In addition, the prover shows that the responses of these parallel Σ-protocols satisfy some multiplicative relation, which completes the protocol. This logarithmic 1-out-of-n protocol gives rise to an efficient ring signature scheme. However, it does not have a straightforward generalization to k-out-of-n proofs of partial knowledge.

By considering m-ary decompositions, instead of binary, the communication efficiency can be further improved [BCC$^+$15]. Recently, a generalization from 1-out-of-n proofs to "many-out-of-many" proofs was given [Dia20]. This generalization considers a prover that claims to know the opening of all commitments in one of the orbits of a public permutation of n public commitments. However, the protocol only works for permutations with orbits of equal size. Since the permutation is public and of this specific form, this protocol does not constitute a general k-out-of-n proof of partial knowledge.

Further efficiency improvements to the 1-out-of-n proofs were introduced in [JM20]. Their protocol applies a hierarchical approach containing two layers of 1-out-of-n proofs reducing the prover's computational efforts. Finally, a non-trivial adaptation of the techniques from [GK15, BCC$^+$15] has resulted in a 1-out-of-n proof of partial knowledge based on lattice assumptions [ESS$^+$19].

An application of proofs of partial knowledge, and in particular ring signature schemes, is a confidential decentralized payment system such as Zerocoin [MGGR13]. Zerocoin was proposed as an extension of Bitcoin to provide stronger privacy guarantees. A Zerocoin transaction requires a ZKPoK that the transferred coin is an element of a public set of unspent coins. The application of the protocol of [GK15] was proposed as an improvement of the original Zerocoin protocol. Other decentralized payment systems that rely on 1-out-of-n proofs to provide confidentiality are, e.g., Lelantus [Jiv19] and Zether [BAZB20]. In [Dia20], it is shown how their generalization to many-out-of-many proofs improves the communication complexity of the Zether payment system. They show that many practical scenarios require more general proofs of partial knowledge than only 1-out-of-n proofs.

Alternatively, proofs of partial knowledge can be constructed via circuit ZK protocols. A standard approach is to incorporate the group elements P_i into a Merkle tree [Mer80], and ask the prover to prove knowledge of k exponents x_i such that the group elements g^{x_i} are the leafs of k valid, but secret, Merkle paths. In this case, the arithmetic circuit implements, for all $1 \leq i \leq k$, a composition of the exponentiation g^{x_i} and $\log_2(n)$ hash function evaluations and is therefore of size $|C| = O(k \log(n))$. The application of circuit ZK protocols with logarithmic communication complexity therefore results in proofs of partial knowledge with communication complexity $O(\log(k) + \log(\log(n)))$. Replacing Merkle trees with RSA-accumulators [BdM93] allows for a further reduction in the asymptotic communication complexity [STY00]. However, these approaches do require the arithmetic circuit to implement the exponentiations g^{x_i}. In suitable groups $\mathbb{G}$, still supporting DL-based cryptography, the exponentiation g^{x_i} can be computed by the evaluation of an arithmetic circuit with approximately 1000 multiplication gates [HBHW20]. This *indirect* approach thus leads to sizable arithmetic circuits. In this work, we focus on a *direct* approach that omits the need for arithmetic circuit formulations of exponentiations and hash functions.

1.3 Organization of the Paper

The remainder of paper is organized as follows. In Section 2, we recall the notation and some of the results from compressed Σ-protocol theory [AC20b]. In Section 3, we describe our twist on the pivotal Σ-protocol from [AC20b]. In Section 4, we combine this generalization with an adaptation of the techniques from [CDS94] to construct our proof of partial knowledge. Finally, in Section 5, we discuss a number of extensions and generalizations of our proofs of partial knowledge.

2 Preliminaries

2.1 Interactive Proofs

We briefly introduce the concept of an interactive proof[6] and some of the basic (security) properties. An interactive proof Π for relation R is a protocol between prover $\mathcal{P}$ and a verifier $\mathcal{V}$. It takes as public input the statement x and as prover's private input the witness w, which is written as $\textsc{Input}(x; w)$. As the output of the protocol the verifier either accepts or rejects the prover's claim of knowing a witness w. Π is called (perfectly) *complete* if on any input $(x; w) \in R$ the verifier always accepts. Evaluating Π on input $(x; w)$ is also written as $\Pi(x; w)$.

An interactive proof with μ communication rounds is also called a μ-move protocol. Note that the final message is always sent from the prover to the verifier. The messages communicated in one protocol evaluation are also referred to as a *conversation* or a *transcript*. If all the messages from the verifier to the prover consist of random coins chosen by the verifier, one speaks of a *public-coin* protocol. All our protocols will be public-coin and thereby suitable for the Fiat-Shamir transformation [FS86], which turns public-coin interactive proofs into *non-interactive* protocols.

An interactive proof Π for relation R is said to have *witness extended emulation* [Lin03] if there exists algorithm χ (witness extended emulator) that runs in expected polynomial time and does the following. The algorithm χ, on input x and given rewindable oracle access to a (possibly dishonest) prover $\mathcal{P}^*$, outputs a transcript and a witness w such that: (1) the emulated transcript is statistically indistinguishable from conversations between $\mathcal{P}^*$ and an honest verifier $\mathcal{V}$, and (2) the probability that the emulated transcript is accepting and the witness w is not a valid witness for x is negligible. An interactive protocol that has witness extended emulation is said to be a *proof of knowledge* (PoK).

We also consider the computational version of a PoK, where witness extended emulation is required to hold only for computationally bounded dishonest provers under a computational hardness assumption. In those cases, the relation R typically depends on a (possibly implicit) security parameter, as well as on some additional public parameters that are assumed to be chosen according to a specific probability distribution,

[6] In contrast to the original definition [GMR85], we do not require an interactive proof to be complete and sound by definition; instead, we consider those (and other) properties as desirable security properties.

and the success probability of the prover is then understood to be on average over the choice of these public parameters. These computational variants of proofs of knowledge are also called arguments of knowledge.

Protocol Π is called *honest verifier zero-knowledge* (HVZK) if there exists an efficient simulator that, on input a statement x that admits a witness w, outputs an accepting transcript, such that the simulated transcripts follow exactly the same distribution as transcripts between an honest prover and an honest verifier.

A 3-move public-coin interactive proof is called a Σ-protocol. The 3 messages are then typically denoted (a, c, z) where c is called the *challenge*. For a HVZK Σ-protocol the simulator often proceeds by first selecting a random challenge c and then preparing the messages a and z; in this case, we speak of *special honest verifier zero-knowledge* (SHVZK).

A Σ-protocol is called k-*special sound* if there exists an efficient algorithm that, on input any statement x and k accepting transcripts $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with common first message a and pairwise distinct challenges c_i, outputs a witness w for x.

More generally, we consider $(2\mu + 1)$-move public-coin protocols, in which all the verifier's messages are uniformly random challenges. These protocols are called $(k_1, \ldots, k_\mu)$-*special sound* if there exists an efficient algorithm that, on input any statement x and a $(k_1, k_2, \ldots, k_\mu)$-tree of accepting transcripts, outputs a witness w for x. A $(k_1, k_2, \ldots, k_\mu)$-tree of accepting transcripts is a set of $\prod_{i=1}^{\mu} k_i$ accepting transcripts that are arranged in the following tree structure. The nodes in this tree correspond to the prover's messages and the edges correspond to the verifier's challenges. Every node at depth i has precisely k_i children corresponding to k_i pairwise distinct challenges. Every transcript corresponds to exactly one path from the root node to a leaf node.

We note that in some public-coin protocols the verifier sends μ challenges in less than $2\mu+1$ rounds, i.e., some of the verifier's messages contain more than one challenge. For these protocols, we also consider the $(k_1, \ldots, k_\mu)$-special soundness property. In this case, a $(k_1, k_2, \ldots, k_\mu)$-tree of accepting transcripts contains nodes that do not correspond to a message sent from the prover to the verifier.

Let us assume that the challenges are sampled uniformly at random from challenge sets with a cardinality that is exponential in the security parameter. In this work all challenge sets are equal to $\mathbb{Z}_q \cong \mathbb{Z}/(q\mathbb{Z})$ for some prime q that is understood to be exponential in the security parameter. Hence, for the protocols in this work this assumption is satisfied. Then witness extended emulation is known to follow from $(k_1, k_2, \ldots, k_\mu)$-special soundness [BCC+16]. In this work, we will show that all protocols are $(k_1, k_2, \ldots, k_\mu)$-special sound for some μ and some list of k_i's, from which witness extended emulation therefore follows.

2.2 Multi-Exponentiation and The Pedersen Vector Commitment Scheme

We consider statements over the ring $\mathbb{Z}_q \cong \mathbb{Z}/(q\mathbb{Z})$ with q prime. We let $\mathbb{G}$ be an Abelian group of prime order q for which we write its group operation multiplicatively. We write vectors in $\mathbb{Z}_q^n$ or $\mathbb{G}^n$ in boldface, i.e., $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$ and $\mathbf{g} := (g_1, \ldots, g_n) \in \mathbb{G}^n$, and we write $\mathbf{g}^{\mathbf{x}}$ for the multi-exponentiation

$$\mathbf{g}^{\mathbf{x}} := \prod_{i=1}^{n} g_i^{x_i} \in \mathbb{G}\,.$$

Furthermore, for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ and scalar $c \in \mathbb{Z}_q$, we have the following component-wise operations:

$$\mathbf{g} * \mathbf{h} := (g_1 h_1, g_2 h_2, \ldots, g_n h_n) \in \mathbb{G}^n,\ \mathbf{g}^c := (g_1^c, g_2^c, \ldots, g_n^c) \in \mathbb{G}^n \ \text{ and } \ \mathbf{x} * \mathbf{y} := (x_1 y_1, x_2 y_2, \ldots, x_n y_n) \in \mathbb{Z}_q^n.$$

Additionally, assuming n is even, we write $\mathbf{g}_L := (g_1, \ldots, g_{n/2}), \mathbf{g}_R := (g_{n/2+1}, \ldots, g_n) \in \mathbb{G}^{n/2}$ and $\mathbf{x}_L := (x_1, \ldots, x_{n/2}), \mathbf{x}_R := (x_{n/2+1}, \ldots, x_n) \in \mathbb{Z}_q^{n/2}$, for the left and right halves of these vectors.

We let $\mathbb{G}_T$ be another Abelian group and denote the set of all group homomorphisms $f : \mathbb{Z}_q^n \to \mathbb{G}_T$ by $\mathrm{Hom}(\mathbb{Z}_q^n, \mathbb{G}_T)$. Typically $\mathbb{G}_T = \mathbb{G}$ or $\mathbb{G}_T = \mathbb{Z}_q$, in the latter case $\mathrm{Hom}(\mathbb{Z}_q^n, \mathbb{G}_T) = \mathrm{Hom}(\mathbb{Z}_q^n, \mathbb{Z}_q)$ is the set of linear forms on $\mathbb{Z}_q^n$. For any homomorphism $f : \mathbb{Z}_q^n \to \mathbb{G}_T$ it holds that its image $\mathrm{im}(f) \subset \mathbb{G}_T$ is a $\mathbb{Z}_q$-module. For this reason, and without loss of generality, we assume that $\mathbb{G}_T$ is a $\mathbb{Z}_q$-module.

Moreover, we define the left and right part of f as follows:

$$\begin{aligned} f_L &: \mathbb{Z}_q^{n/2} \to \mathbb{G}_T, \quad \mathbf{x} \mapsto f(\mathbf{x}, 0), \\ f_R &: \mathbb{Z}_q^{n/2} \to \mathbb{G}_T, \quad \mathbf{x} \mapsto f(0, \mathbf{x}), \end{aligned} \tag{1}$$

where, e.g., $(\mathbf{x}, 0) \in \mathbb{Z}_q^n$ is the vector $\mathbf{x}$ appended with $n/2$ zeros.

In this work we also consider the Pedersen vector commitment scheme. This commitment scheme allows a prover to (compactly) commit to an n-dimensional vector $\mathbf{x} \in \mathbb{Z}_q^n$ in a single group element $P \in \mathbb{G}$. We recall that a Pedersen vector commitment P is simply a multi-exponentiation, i.e.,

$$P = h^\gamma \mathbf{g}^\mathbf{x},$$

for public parameters $h \in \mathbb{G}$ and $\mathbf{g} \in \mathbb{G}^n$ and for a (private) $\gamma \in \mathbb{Z}_q$ sampled uniformly at random by the prover.

The Pedersen vector commitment scheme is perfectly hiding and computationally binding under the discrete logarithm assumption. More precisely, the commitment scheme is binding under the assumption that a prover does not know a non-zero vector $(\gamma, x_1, \ldots, x_n) \in \mathbb{Z}_q^{n+1}$ such that

$$h^\gamma \prod_{i=1}^{n} g_i^{x_i} = 1.$$

Such a non-zero vector $(\gamma, x_1, \ldots, x_n)$ is also called a non-trivial discrete log relation for group elements $h, g_1, \ldots, g_n$. From here on forward, we assume that these group elements have been sampled uniformly at random in a setup phase and that the prover does not know a non-trivial discrete logarithm (DL) relation. These group elements form the set of public parameters for all our protocols. We say a protocol is computationally $(k_1, \ldots, k_\mu)$-special sound, under the discrete logarithm assumption, if $(k_1, \ldots, k_\mu)$-special soundness holds under the assumption that a prover does not know a non-trivial DL relation between the public parameters.

3 Proving Group Homomorphisms on Multi-Exponentiations

In this section, we construct an interactive proof protocol for proving that a secret multi-exponent $\mathbf{x} \in \mathbb{Z}_q^n$ for a public multi-exponentiation $P = \mathbf{g}^\mathbf{x} \in \mathbb{G}$ is mapped to a given public value y under an arbitrary but given group homomorphism $f : \mathbb{Z}_q^n \to \mathbb{G}_T$. Our new protocol has a communication complexity that is logarithmic in the dimension n. By considering one of the coordinates of $\mathbf{x}$ to be "the randomness", and considering an f that ignores this coordinate, we immediately get a protocol that applies to Pedersen vector commitments and proves that the committed vector satisfied the relation defined by the considered group homomorphism and the target value P.

Our approach for constructing said protocol is as follows. We start with the canonical Σ-protocol for the considered problem of proving $f(\mathbf{x}) = y$ (Section 3.1), and we then adapt the compression mechanism of [AC20b] such that it is applicable to our setting. Indeed, our setting is a generalization of [AC20b], which applies to the special case where f is a linear form $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$. This then results in a compressed Σ-protocol that features the claimed logarithmic complexity (Section 3.3).

Later in the section, we also discuss a couple of (standard) amortization techniques applied to our protocol, for instance for proving $f_i(\mathbf{x}) = y_i$ for *several* group homomorphisms f_i at (essentially) the cost of proving one.

3.1 The Standard Σ-protocol for Opening Homomorphisms

We consider the problem of proving that the multi-exponent $\mathbf{x}$ of a multi-exponentiation $P = \mathbf{g}^\mathbf{x}$ is mapped to a certain value y under a given homomorphism $f \in \mathrm{Hom}(\mathbb{Z}_q, \mathbb{G}_T)$, i.e., that $f(\mathbf{x}) = y$, without revealing $\mathbf{x}$. More concretely, we want to construct PoK protocols for the relation

$$R = \left\{ \left(P \in \mathbb{G}, f \in \mathrm{Hom}(\mathbb{Z}_q^n, \mathbb{G}_T), y \in \mathbb{G}_T; \mathbf{x} \in \mathbb{Z}_q^n\right) : P = \mathbf{g}^\mathbf{x}, y = f(\mathbf{x}) \right\}. \tag{2}$$

Protocol 1, denoted by Π_0, is the canonical Σ-protocol for this relation R, following the generic construction design for q-one-way group homomorphisms[7] [Cra96, CD98]. The properties of Π_0, known to hold for this generic construction, are summarized in Theorem 1. Note that the only difference between this protocol and Protocol 2 of [AC20b] is that here we consider multi-exponentiations and general group homomorphisms instead of Pedersen commitments and linear forms.

Theorem 1 (Homomorphism Evaluation). *Π_0 is a Σ-protocol for relation R. It is perfectly complete, special honest-verifier zero-knowledge and unconditionally special sound. Moreover, the communication costs are:*

- *$\mathcal{P} \to \mathcal{V}$: 1 element of $\mathbb{G}$, 1 element of $\mathbb{G}_T$ and n elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathbb{Z}_q$.*

Protocol 1 Σ-protocol Π_0 for relation R
Opening a homomorphism on a Pedersen vector commitment.

$$\text{Public Parameters} : \mathbf{g} \in \mathbb{G}^n,$$
$$\text{Input}(P, f, y; \mathbf{x})$$

$$P = \mathbf{g}^{\mathbf{x}} \in \mathbb{G}$$
$$y = f(\mathbf{x}) \in \mathbb{G}_T$$

Prover		Verifier
$\mathbf{r} \leftarrow_R \mathbb{Z}_q^n$		
$A = \mathbf{g}^{\mathbf{r}}$		
$t = f(\mathbf{r})$		
	$\xrightarrow{A,t}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = c\mathbf{x} + \mathbf{r}$		
	$\xrightarrow{\mathbf{z}}$	
		$\mathbf{g}^{\mathbf{z}} \stackrel{?}{=} AP^c$
		$f(\mathbf{z}) \stackrel{?}{=} cy + t$

3.2 Compression mechanism

The Σ-protocol Π_0 for opening homomorphisms has a linear communication complexity. We now deploy the techniques from [BCC$^+$16, BBB$^+$18, AC20b] to compress the communication complexity from linear to logarithmic. A first observation is that the verifiers final check verifies that

$$(AP^c, f, cy + t; \mathbf{z}) \in R\,,$$

i.e., that the prover's final message $\mathbf{z}$ is a witness with respect to the same relation R for the statement $(AP^c, f, cy + t)$; which is computed by the verifier. This is no coincidence; this holds generically for this standard construction of Σ-protocols for q-one-way group homomorphisms. The final message of Π_0 can therefore be understood as a trivial PoK for relation R, and replacing this trivial PoK by a more efficient one will reduce the communication complexity with affecting security (significantly). In particular, the alternative PoK does not have to be zero-knowledge since the trivial one obviously is not.

[7] Here, applied to the q-one-way group homomorphisms $\mathbb{Z}_q^n \to \mathbb{G} \times \mathbb{G}_T$, $\mathbf{x} \mapsto (\mathbf{g}^{\mathbf{x}}, f(\mathbf{x}))$.

Our compression mechanism is thus an interactive proof Π_1 for relation R that is not zero-knowledge anymore but has improved efficiency. The compression mechanism is very similar to the one used in [AC20b]. The difference is that we consider the more general case of opening arbitrary group homomorphisms, rather than restricting ourselves to linear forms. This generalization requires a minor adaptation. The first step in the compression of [AC20b] is namely to incorporate the linear form evaluation into the multi-exponentiation as an additional exponent on a new generator $k \in \mathbb{G}$. This reduction step does not apply to the general case of opening arbitrary group homomorphisms, and is therefore omitted in our protocols. For this reason we directly apply (a minor adaptation of) the main compression mechanism of [AC20b]; ultimately this will increase the communication costs of the compressed Σ-protocol by roughly a factor two when compared to opening linear forms. However, in contrast to the compressed Σ-protocol for opening linear forms [AC20b], our protocol is unconditionally sound rather than computationally.

The compression mechanism, i.e., our protocol Π_1 for relation R that has improved efficiency but is not zero-knowledge, is described in Protocol 2 below. Here, n is assumed to be even, which is without loss of generality (if not the witness can be appended with a zero). Also, recall that $\mathbf{x}_L := (x_1, \ldots, x_{n/2})$ equals the left half of vector $\mathbf{x} \in \mathbb{Z}_q^n$ and that $f_R(\mathbf{x}_L) := f(0, \ldots, 0, \mathbf{x}_L)$, etc.

Before discussing the security of Π_1 as a proof of knowledge in Theorem 2, we emphasize the following two important properties of Π_1. The size of the response has halved compared to the original protocol Π_0, and thereby the communication costs are reduced by roughly a factor two, and second, verifying the correctness of the response is again by means of checking whether it is a witness for the same relation R, but now for the group homomorphism $f' := cf_L + f_R \in \mathrm{Hom}\big(\mathbb{Z}_q^{n/2}, \mathbb{G}_T\big)$.

Protocol 2 Compression Mechanism Π_1 for relation R_1.

Public Parameters : $\mathbf{g}$

Input$(P, f, y; \mathbf{x})$

$P = \mathbf{g}^{\mathbf{x}} \in \mathbb{G}$

$y = f(\mathbf{x}) \in \mathbb{G}_T$

Prover		Verifier
$A = \mathbf{g}_R^{\mathbf{x}_L}, a = f_R(\mathbf{x}_L)$ $B = \mathbf{g}_L^{\mathbf{x}_R}, b = f_L(\mathbf{x}_R)$		
	$\xrightarrow{A,B,a,b}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
	$\mathbf{g}' := \mathbf{g}_L^c * \mathbf{g}_R \in \mathbb{G}^{n/2}$ $Q := AP^cB^{c^2}$ $f' := cf_L + f_R$	
$\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R$		
	$\xrightarrow{\mathbf{z}}$	$(\mathbf{g}')^{\mathbf{z}} \stackrel{?}{=} Q$
		$f'(\mathbf{z}) \stackrel{?}{=} a + cy + c^2 b$

Theorem 2 (Compression Mechanism). *Let $n \in \mathbb{Z}_{>0}$ be even. Then Π_1 is a 3-move protocol for relation R. It is perfectly complete and unconditionally 3-special sound. Moreover, the communication costs are:*

- *$\mathcal{P} \to \mathcal{V}$: 2 elements of $\mathbb{G}$, 2 elements of $\mathbb{G}_T$ and $n/2$ elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathbb{Z}_q$.*

Proof. **Completeness** follows directly.

Special Soundness: We show that the protocol is 3-special sound, i.e., there exists an efficient algorithm that on input three accepting transcripts computes a witness for relation R.

Let $(A, B, a, b, c_1, \mathbf{z}_1)$, $(A, B, a, b, c_2, \mathbf{z}_2)$ and $(A, B, a, b, c_3, \mathbf{z}_3)$ be three accepting transcripts for distinct challenges $c_1, c_2, c_3 \in \mathbb{Z}_q$. Let $a_1, a_2, a_3 \in \mathbb{Z}_q$ be such that

$$\begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_2 & c_3 \\ c_1^2 & c_2^2 & c_3^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Note that since the challenges are distinct, this Vandermonde matrix is invertible and a solution to this equation exists. We define $\bar{\mathbf{z}} = \sum_{i=1}^{3} a_i(c_i\mathbf{z}_i, \mathbf{z}_i)$ for which it is easily verified that

$$\mathbf{g}^{\bar{\mathbf{z}}} = P \qquad \text{and} \qquad f(\bar{\mathbf{z}}) = y.$$

Hence, $\bar{\mathbf{z}}$ is a witness for relation R, which completes the proof.

3.3 Compressed Σ-protocol

Finally, we compose Σ-protocol Π_0 and its compression mechanism Π_1 to obtain a compressed Σ-protocol for opening homomorphisms on multi-exponentiations $\mathbf{g}^{\mathbf{x}}$ such as Pedersen vector commitments. We follow the notation of [AC20b] and write $\Pi_b \diamond \Pi_a$ for the composition of two composable interactive proof Π_a and Π_b. Protocols Π_a and Π_b are composable if protocol Π_b is a PoK for the prover's final message of protocol Π_a. Recall that this composition means that the final message of protocol Π_a is replaced by an execution of protocol Π_b.

We assume that n is a power of two, if it is not the witness can be appended with zeros such that its dimension is a power of 2. For $n \geq 2$ it is optimal to omit the compression mechanism, for this reason it is assumed that $n > 2$. To minimize the communication complexity we recursively apply the compression protocol Π_1 until the dimension of the witness is reduced to four, i.e., $\mu = \lceil \log_2(n) \rceil - 2$ times. For this composition we write

$$\Pi_c = \underbrace{\Pi_1 \diamond \cdots \diamond \Pi_1}_{\mu \text{ times}} \diamond \Pi_0. \tag{3}$$

Theorem 3 captures the security and efficiency properties of Protocol Π_c.

Theorem 3 (Compressed Σ-Protocol for Opening Homomorphisms). *Let $n > 2$, then Π_c is a $(2\mu + 3)$-move protocol for relation R, where $\mu = \lceil \log_2(n) \rceil - 2$. It is perfectly complete, special honest-verifier zero-knowledge and unconditionally $(2, k_1, \ldots, k_\mu)$-special sound, where $k_i = 3$ for all $1 \leq i \leq \mu$. Moreover, the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}$, $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}_T$ *and* 4 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n) \rceil - 1$ *elements of* $\mathbb{Z}_q$.

Proof. **Completeness** follows in a straightforward manner.

Special Honest Verifier Zero-Knowledge follows since Π_0 is SHVZK. A simulator for Π_c runs the simulator for Π_0, and replaces the final messages of the simulated transcripts by honest executions of $\Pi_1 \diamond \cdots \diamond \Pi_1$.

Special Soundness: Since the protocol is the composition of protocols that are 2- or 3-special sound, it is easily seen that Π_c is $(2, 3, \ldots, 3)$-special sound, i.e., there exists an efficient algorithm that on input a $(2, 3, \ldots, 3)$-tree (depth $\mu + 1$) of $2 \cdot 3^\mu$ accepting transcripts computes a witness for relation R.

Remark 1. We explicitly emphasize once more that the above and below results on opening homomorphisms $f(\mathbf{x})$ on multi-exponentiations $\mathbf{g}^{\mathbf{x}}$ immediately carry over to opening homomorphisms $f(\mathbf{x})$ on Pedersen vector commitments $\mathbf{g}^{\mathbf{x}}h^{\gamma}$, simply by renaming the involved variables in the obvious way.

3.4 Amortization Techniques

This section describes two standard amortization techniques. First, we consider the scenario where a prover wishes to open *one* homomorphism f on *many* multi-exponentiations $P_1, \ldots, P_s$, i.e., we consider the relation

$$R_{\text{AMORExp}} = \left\{ (P_1, \ldots, P_s, f, y_1, \ldots, y_s; \mathbf{x}_1, \ldots, \mathbf{x}_s) : P_1 = \mathbf{g}^{\mathbf{x}_1}, y_1 = f(\mathbf{x}_1), \ldots, P_1 = \mathbf{g}^{\mathbf{x}_1}, y_s = f(\mathbf{x}_s) \right\}. \quad (4)$$

The standard (amortized) Σ-protocol for relation R_{AMORExp} is similar to Σ-protocol Π_0 for relation R: it has the same first two moves, but then the prover's final response is $\mathbf{z} = \mathbf{r} + \sum_{i=1}^{s} c^i \mathbf{x}_i$ and the verifier checks that $\mathbf{g}^{\mathbf{z}} = AP_1^c \cdots P_s^{c^s}$ and $f(\mathbf{z}) = t + cy + \cdots + c^s y_s$. The communication costs of the amortized Σ-protocol are exactly equal to the communication costs of protocol Π_0 and the compression mechanism applies as before. We denote the compressed amortized Σ-protocol for relation R_{AMORExp} by Π_{AMORExp}. Its main properties are summarized in Theorem 4.

Theorem 4 (Amortization over Many Multi-Exponentiations). *Let $n > 2$, then Π_{AMORExp} is a $(2\mu + 3)$-move protocol for relation R_{AMORExp}, where $\mu = \lceil \log_2(n) \rceil - 2$. It is perfectly complete, special honest-verifier zero-knowledge and unconditionally $(s + 1, k_1, \ldots, k_\mu)$-special sound, where $k_i = 3$ for all $1 \leq i \leq \mu$. Moreover, the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}$, $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}_T$ *and* 4 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n) \rceil - 1$ *elements of* $\mathbb{Z}_q$.

Second, we consider the amortization scenario where a prover wishes to open *many* homomorphisms $f_1, \ldots, f_s$ on *one* multi-exponentiation P, i.e., we consider a compressed Σ-protocol for the following relation

$$R_{\text{AMORHom}} = \left\{ (P, f_1, \ldots, f_s, y_1, \ldots, y_s; \mathbf{x}) : P = \mathbf{g}^{\mathbf{x}}, y_1 = f_1(\mathbf{x}), \ldots y_s = f_s(\mathbf{x}) \right\}. \quad (5)$$

This scenario is reduced to the original scenario of opening one homomorphism on one commitment by means of a standard polynomial amortization trick. In the first move of the protocol, the verifier sends a random challenge $\rho \in \mathbb{Z}_q$ to the prover, and then Π_c is executed on the instance given by $P = \mathbf{g}^{\mathbf{x}}$, $f_\rho = f_1 + \rho f_2 + \cdots + \rho^{s-1} f_s$ and $y_\rho = y_1 + \rho y_2 + \cdots + \rho^{s-1} y_s$.

The core idea behind this construction is the observation that if $\mathbf{x}$ satisfies $f_\rho(\mathbf{x}) = y_\rho$ for s distinct choices of ρ then $f_i(\mathbf{x}) = y_i$ for all $i \in \{1, \ldots, s\}$. A caveat is that when trying to extract such an $\mathbf{x}$ by rewinding $s - 1$ times and choosing different ρ's, one might potentially extract different choices of $\mathbf{x}$'s. However, since $\mathbf{g}^{\mathbf{x}} = P$ must still hold, this would lead to a non-trivial DL relation among the g_i's, and thus cannot happen when the prover is computationally bounded.

The properties of this protocol for relation R_{AMORHom}, denoted by Π_{AMORHom}, are summarized in Theorem 5. Note that the communication from prover to verifier is identical to that of protocol Π_c. However, the polynomial amortization trick degrades the soundness from unconditional to computational because of the above reason.

Theorem 5 (Amortization over Many Homomorphisms). *Let $n > 2$, then Π_{AMORHom} is a $(2\mu + 4)$-move protocol for relation R_{AMORHom}, where $\mu = \lceil \log_2(n) \rceil - 2$. It is perfectly complete, special honest-verifier zero-knowledge and computationally $(s, 2, k_1, \ldots, k_\mu)$-special sound, under the discrete logarithm assumption in $\mathbb{G}$, where $k_i = 3$ for all $1 \leq i \leq \mu$. Moreover, the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}$, $2\lceil \log_2(n) \rceil - 3$ *elements of* $\mathbb{G}_T$ *and* 4 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n) \rceil$ *elements of* $\mathbb{Z}_q$.

In the above claim on the computational special soundness we take it as understood $g_1, \ldots, g_n$ are chosen uniformly at random in $\mathbb{G}$.

Proof. **Completeness** and **SHVZK** follow directly from the corresponding properties of Protocol Π_c.

Special Soundness: From the proof of Theorem 3 we know that for every ρ there exists an efficient algorithm that, from any $(2, 3, \ldots, 3)$-tree (depth $\mu + 1$) of accepting transcripts, extracts a witness $\mathbf{z}$ such that $\mathbf{g}^{\mathbf{z}} = P$ and $f_\rho(\mathbf{z}) = y_1 + \rho y_2 + \cdots + \rho^{s-1} y_s$.

We show that there also exists an efficient algorithm that, from s exponents $\mathbf{z}_1, \ldots, \mathbf{z}_s \in \mathbb{Z}_q^n$ such that $\mathbf{g}^{\mathbf{z}_i} = P$ and $f_{\rho_i}(\mathbf{z}_i) = y_1 + \rho_i y_2 + \cdots + \rho_i^{s-1} y_s$ for all i and for pairwise distinct challenges $\rho_i \in \mathbb{Z}_q$, extracts either a non-trivial DL-relation for the public parameters $\mathbf{g}$ or a witness for relation R_{AMORHOM}. Combining these two results shows that Protocol Π_{AMORHOM} is $(s, 2, 3, \ldots, 3)$-special sound from which knowledge soundness follows from [AC20b].

First suppose that there exist $1 \leq i, j \leq s$ such that $\mathbf{z}_i \neq \mathbf{z}_j$. Then $\mathbf{g}^{\mathbf{z}_i} = P = \mathbf{g}^{\mathbf{z}_j}$ gives a non-trivial DL-relation, which completes the proof for this case.

Now suppose that $\mathbf{z}_i = \mathbf{z}$ for all i. Let $(a_{i,j})_{1 \leq i,j \leq s}$ be the inverse of the Vandermonde matrix generated by the challenges $\rho_1, \ldots, \rho_s$, i.e.,

$$\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ \rho_1^s & \cdots & \rho_s^s \end{pmatrix} \begin{pmatrix} a_{1,1} & \cdots & a_{1,s} \\ \vdots & \ddots & \vdots \\ a_{s,1} & \cdots & a_{s,s} \end{pmatrix} = I_s.$$

Note that this Vandermonde matrix is invertible because the challenges are pairwise distinct. Then for all $1 \leq i \leq s$ it holds that

$$f_i(\mathbf{z}) = a_{1,i} f_{\rho_1}(\mathbf{z}) + \cdots + a_{s,i} f_{\rho_s}(\mathbf{z}) = y_i.$$

Hence $\mathbf{z}$ is a witness for relation R_{AMORHOM} which completes the proof.

4 Proving *Partial* Knowledge

Here, we show our new efficient proofs for partial knowledge, i.e., for proving knowledge of k-out-of-n discrete logarithms (Section 4.1), and for proving knowledge of k-out-of-n commitment openings (Section 4.2). As we will see, these new proofs of partial knowledge follow quite easily by exploiting the core idea of the general construction in [CDS94] and combining it with the techniques and results from the section above. This further demonstrates the strength of combining the compression technique introduced by [BCC+16, BBB+18] with general Σ-protocol theory.

4.1 Partial Knowledge of DL's

In this section we construct a simple SHVZK proof of knowledge for proving knowledge of k-out-of-n discrete logarithms. Our protocol inherits the logarithmic communication from the compressed Σ-protocol(s) from the previous section. More precisely, we give a SHVZK protocol for the following relation

$$\begin{aligned} R_{\text{PARTIAL}} = \big\{ \big(g, P_1, \ldots, P_n \in \mathbb{G}, k \in \{1, \ldots, n\}; S \subset \{1, \ldots, n\}, \mathbf{x} \in \mathbb{Z}_q^n\big) : \\ |S| = k, P_i = g^{x_i} \text{ for all } i \in S \big\}. \end{aligned} \tag{6}$$

Note that, for notational convenience, the witness $\mathbf{x}$ is defined as a vector in $\mathbb{Z}_q^n$ while only the k coefficients $(x_i)_{i \in S}$ are relevant in this relation.

The protocol goes as follows. First, the prover computes the unique polynomial

$$p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j \in \mathbb{Z}_q[X]$$

of degree at most $n - k$ such that $p(0) = 1$ and $p(i) = 0$ for all $i \notin S$.

Second, the prover computes

$$t_i := p(i) x_i$$

for $i \in \{1, \ldots, n\}$ (recall that $p(i)$ vanishes for those i for which the prover does not know x_i), and sends a Pedersen commitment $P \in \mathbb{G}$ to the vector

$$\mathbf{y} = (a_1, \ldots, a_{n-k}, t_1, \ldots, t_n) \in \mathbb{Z}_q^{2n-k}$$

to the verifier. Here, the commitment P is computed as $P = \mathbf{g}^{\mathbf{y}} h^{\gamma}$ with respect to public parameters $\mathbf{g} = (g_1, \ldots, g_{2n-k}) \in \mathbb{G}^{2n-k}$ and $h \in \mathbb{G}$ for which no non-trivial DL-relations are known to the prover, i.e., so that the commitment is indeed binding.

Finally, the prover proves to the verifier that the committed vector $\mathbf{y}$ satisfies

$$g^{t_i} = P_i^{p(i)} \tag{7}$$

for all $i \in \{1, \ldots, n\}$, where the exponent $p(i)$ on the right-hand-side term is understood as the evaluation of the affine function $(w_1, \ldots, w_{n-k}) \mapsto 1 + \sum_{j=1}^{n-k} w_j i^j$ applied to $a_1, \ldots, a_{n-k}$. Thus, rewriting (7) as

$$g^{t_i} P_i^{-\sum_j a_j i^j} = P_i \tag{8}$$

we obtain an expression where the left hand side is a group homomorphism f applied to the committed committed vector $\mathbf{y}$, and thus the prover can prove one instance of (7) by means of the compressed protocol from Theorem 3; respectively, for improved efficiency, it can invoke the amortized protocol Π_{AMORHOM} from Theorem 5 for proving that (7) holds *for all* $i \in \{1, \ldots, n\}$.

The resulting protocol, denoted Π_{PARTIAL}, is summarized below in Protocol 3. We note that, in line with the amortized protocol it uses as a subroutine, it is *computationally* special sound, based on the assumption that the prover does not know any non-trivial DL-relations among the public parameters. Formally, we have the following security and efficiency properties.

Protocol 3 SHVZK Proof of Partial Knowledge Π_{PARTIAL} for Relation R_{PARTIAL}
Proving knowledge of k-out-of-n discrete logarithms.

PUBLIC PARAMETERS : $\mathbf{g} \in \mathbb{G}^{2n-k}, h \in \mathbb{G}$

INPUT $(g, P_1, \ldots, P_n, k; S, \mathbf{x})$

$S \subset \{1, \ldots, n\}, |S| = k$
$g^{x_i} = P_i$ for $i \in S$

Prover		Verifier
$p(X) = 1 + \sum_{i=1}^{n-k} a_i X^i$ s.t. $p(i) = 0 \quad \forall i \notin S$ $\mathbf{y} = (a_1, \ldots, a_{n-k}, p(1)x_1, \ldots, p(n)x_n)$ $\gamma \leftarrow_R \mathbb{Z}_q, P = \mathbf{g}^{\mathbf{y}} h^{\gamma}$	$\xrightarrow{P}$	
		Run Π_{AMORHOM} to prove that $\mathbf{y}$ satisfies $g^{y_{i+n-k}} P_i^{-\sum_j y_j i^j} = P_i \quad \forall i \in \{1, \ldots, n\}$

Theorem 6 (k-out-of-n SHVZK Proof of Partial Knowledge). *Let $n > 1$, then Π_{PARTIAL} is a $(2\mu+5)$-move protocol for relation R_{PARTIAL}, where $\mu = \lceil \log_2(2n - k + 1) \rceil - 2$. It is perfectly complete, special honest-verifier zero-knowledge and computationally $(n, 2, k_1, \ldots, k_\mu)$-special sound, under the discrete logarithm assumption in $\mathbb{G}$, where $k_i = 3$ for all $1 \leq i \leq \mu$. Moreover, the communication costs are:*

– $\mathcal{P} \to \mathcal{V}$: $4\lceil \log_2(2n-k+1)\rceil - 5$ *elements of* $\mathbb{G}$ *and* 4 *elements of* $\mathbb{Z}_q$.
– $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(2n-k+1)\rceil$ *elements of* $\mathbb{Z}_q$.

Proof. **Completeness** follows in a straightforward manner.

Special Honest Verifier Zero-Knowledge follows immediately from the fact that P is uniformly random and from the corresponding zero-knowledge property of Π_{AMORHOM}.

Special Soundness: The computational special soundness of Π_{AMORHOM} guarantees existence of an extractor that extracts, from any computationally-bounded successful prover, an opening $\mathbf{y} = (a_1, \ldots, a_{n-k}, t_1, \ldots, t_n)$ of the commitment P for which (8) holds for all $i \in \{1, \ldots, n\}$, and thus, considering the corresponding polynomial $p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j$, for which (7) holds for all $i \in \{1, \ldots, n\}$. Given the bounded degree of p and the non-zero constant coefficient, $p(i) = 0$ for at most $n-k$ choices of $i \in \{1, \ldots, n\}$. Thus, setting $S := \{i : p(i) \neq 0\}$, we have $|S| \geq k$, and for any $i \in S$ we can set $x_i := t_i / p(i)$ and (7) then implies that $g^{x_i} = P_i$.

4.2 Partial Knowledge of Commitment Openings

In the previous section we constructed a protocol for proving knowledge of k-out-of-n discrete logarithms or, equivalently, a protocol for showing that a prover can open k-out-of-n Pedersen commitments to 0. This protocol can easily be adapted to accommodate, for example, the following variation of this zero-knowledge scenario.

In this variation we let $P_1, \ldots, P_n$ be Pedersen commitments, for which the prover claims to know k-out-of-n openings, not necessarily to 0. More precisely, the prover claims to know a witness for the following relation:

$$R_{\text{PARTIALCOM}} = \big\{ \big(g, h, P_1, \ldots, P_n \in \mathbb{G}, k \in \{1, \ldots, n\}; S \subset \{1, \ldots, n\}, x_1, \ldots, x_n \in \mathbb{Z}_q, \\ \gamma_1, \ldots, \gamma_n \in \mathbb{Z}_q\big) : |S| = k, P_i = g^{x_i} h^{\gamma_i} \text{ for all } i \in S \big\}.^{8} \tag{9}$$

A proof of knowledge for relation $R_{\text{PARTIALCOM}}$ is obtained by applying the following adaptations. After defining the the polynomial $p(X)$ as before, the prover computes

$$t_i := p(i) x_i \in \mathbb{Z}_q \quad \text{and} \quad s_i := p(i)\gamma_i \in \mathbb{Z}_q,$$

for $i \in \{1, \ldots, n\}$ and sends a Pedersen commitment $P \in \mathbb{G}$ to the vector

$$\mathbf{y} = (a_1, \ldots, a_{n-k}, t_1, \ldots, t_n, s_1, \ldots, s_n) \in \mathbb{Z}_q^{3n-k},$$

to the verifier. Finally, by invoking Protocol Π_{AMORHOM}, the prover shows that

$$g^{t_i} h^{s_i} P_i^{-\sum_j a_j i^j} = P_i$$

for all $i \in \{1, \ldots, n\}$. Formally, we have the following security and efficiency properties.

Theorem 7 (k-out-of-n SHVZK Proof of Partial Knowledge for Commitment Openings). *$\Pi_{\text{PARTIALCOM}}$ is a $(2\mu+5)$-move protocol for relation $R_{\text{PARTIALCOM}}$, where $\mu = \lceil \log_2(3n-k+1)\rceil - 2$. It is perfectly complete, special honest-verifier zero-knowledge and computationally $(n, 2, k_1, \ldots, k_\mu)$-special sound, under the discrete logarithm assumption in $\mathbb{G}$, where $k_i = 3$ for all $1 \leq i \leq \mu$. Moreover, the communication costs are:*

– $\mathcal{P} \to \mathcal{V}$: $4\lceil \log_2(3n-k+1)\rceil - 5$ *elements of* $\mathbb{G}$ *and* 4 *elements of* $\mathbb{Z}_q$.
– $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(3n-k+1)\rceil$ *elements of* $\mathbb{Z}_q$.

[8] The element $h \in \mathbb{G}$, used in the commitments P_i, is not necessarily the same element as the element $h \in \mathbb{G}$ used in the Pedersen vector commitment P of Protocol Π_{PARTIAL}.

Remark 2. We emphasize that $\Pi_{\text{PartialCom}}$ is only special sound under the assumption that the prover does not know a non-trivial DL relation between the public parameters $\mathbf{g} \in \mathbb{G}^{3n-k}$ and $h \in G$ for the Pedersen commitment P to the vector $\mathbf{y}$, i.e., it is crucial that the commitment P is binding. In contrast, the special soundness of $\Pi_{\text{PartialCom}}$ does not depend on a computational assumption regarding the public parameters $g, h \in \mathbb{G}$ for the Pedersen commitments P_i, i.e., the commitments P_i are not required to be binding for Protocol $\Pi_{\text{PartialCom}}$ to be special sound.

5 Extensions and Generalizations

Our techniques from Section 4 for proofs of partial knowledge can be extended and generalized in various directions. We discuss some examples here.

5.1 Multi-Exponentiations and Vector Commitments

A straightforward generalization of Protocol Π_{Partial} shows that, instead of the DL problem for standard exponentiations, we can also consider multi-exponentiations. More concretely, this generalization gives a protocol for the following relation

$$\begin{aligned} R' = \big\{ \big(\mathbf{h} \in \mathbb{G}^m, P_1, \ldots, P_n \in \mathbb{G}, k \in \{1, \ldots, n\}; S \subset \{1, \ldots, n\}, \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m\big) : \\ |S| = k, P_i = \mathbf{h}^{\mathbf{x}_i} \text{ for all } i \in S \big\}. \end{aligned} \tag{10}$$

The only adaptation of protocol Π_{Partial} that is required is the replacement of the scalars $x_i \in \mathbb{Z}_q$ by vectors $\mathbf{x}_i \in \mathbb{Z}_q^m$. The communication complexity of the resulting protocol grows logarithmically in the dimension m of the multi-exponentiations. In a completely analogous manner, protocol $\Pi_{\text{PartialCom}}$ from Section 4.2 can be generalized to proving partial knowledge of Pedersen vector commitment openings.

5.2 Plug and Play with Circuit Zero-Knowledge

In many practical scenarios, one wishes to prove not only partial knowledge of commitment openings, but also that the committed values satisfy some additional constraints. Typically these constraints are defined by an arithmetic circuit $C : \mathbb{Z}_q^n \to \mathbb{Z}_q$ and the committed values $x_1, \ldots, x_n \in \mathbb{Z}_q$ are claimed to satisfy $C(x_1, \ldots, x_n) = 0$. More concretely, we consider a prover that claims to know a witness for the following relation

$$\begin{aligned} R_{\text{PartialCirc}} = \big\{ \big(g, h, P_1, \ldots, P_n \in \mathbb{G}, k \in \{1, \ldots, n\}; S \subset \{1, \ldots, n\}, x_1, \ldots, x_n \in \mathbb{Z}_q, \\ \gamma_1, \ldots, \gamma_n \in \mathbb{Z}_q\big) : |S| = k, P_i = g^{x_i} h^{\gamma_i} \text{ for all } i \in S, C(x_1, \ldots, x_n) = 0 \big\}. \end{aligned} \tag{11}$$

Note that in this relation the prover is only committed to k-out-of-n scalars x_i, i.e., it can choose $n - k$ scalars freely.

To handle this extension of the partial knowledge scenario we deploy the circuit ZK techniques from [AC20b]. For these techniques to be applicable all we need to show is that we can open homomorphisms and linear forms on the same Pedersen vector commitment. In [AC20b] it is namely shown how circuit ZK protocols, for arbitrary arithmetic circuits, are derived from the functionality of opening linear forms on Pedersen vector commitments.

However, for any homomorphism $f : \mathbb{Z}_q^n \to \mathbb{G}_T$ and any linear form $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$ it is easily seen that the following map is again a homomorphism

$$(f, L) : \mathbb{Z}_q^n \to \mathbb{G}_T \times \mathbb{Z}_q \quad \mathbf{x} \mapsto (f(\mathbf{x}), L(\mathbf{x})).$$

So the functionality of Protocol Π_c, opening homomorphism, trivially extends to the functionality of opening homomorphism *and* linear forms on the same vector commitment.

Applying this approach directly results in a protocol for relation $R_{\text{PartialCirc}}$ where the communication costs, from prover to verifier, are roughly $6 \log_2(n)$ elements. These communication costs can be reduced to roughly $4 \log_2(n)$ elements by incorporating the linear form evaluation into the Pedersen vector commitment as in [AC20b].

Remark 3. Various other (natural) circuit ZK scenarios exist. For example, when the circuit $C : \mathbb{Z}_q^k \to \mathbb{Z}_q$ only takes the scalars x_i for $i \in S$ as input. Many of these scenarios are easily dealt with by plug and play (modular design) with the techniques from [AC20b].

5.3 General Access Structures

Thus far, we have restricted ourselves to provers that claim to know the solutions of some (secret) subset S, of cardinality at least k, of n (public) DL problems $P_i = g^{x_i}$, i.e., the secret subset S is an element of a *threshold* access structure

$$\Gamma_{k,n} = \{A \subset \{1, \ldots, n\} : |A| \geq k\} \subset 2^{\{1,\ldots,n\}}.$$

Here, we describe how the protocols from Section 4 can easily be generalized to arbitrary monotone access structures $\Gamma \subset 2^{\{1,\ldots,n\}}$, i.e., to provers that claim to know the solutions of some subset of $S \in \Gamma$ of n DL problems. Recall that Γ is called a monotone access structure if for all $A \in \Gamma$ and for all $B \supset A$ it holds that $B \in \Gamma$. The proofs of partial knowledge of [CDS94] already considered arbitrary access structures and we adapt their techniques by combining it with our compression framework.

Our proofs of k-out-of-n partial knowledge implicitly deploy a linear secret sharing scheme (LSSS) for access structure $\Gamma^*_{k,n} = \Gamma_{n-k,n}$. Here, Γ^* denotes the *dual* of access structure, generally given by

$$\Gamma^* = \{A \subset \{1, \ldots, n\} : A^c \notin \Gamma\}.$$

More concretely the protocols of Section 4 use Shamir's secret sharing scheme and the polynomial $p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j$ defines a secret sharing of the field element 1.

To construct a proof of partial knowledge for monotone access structure Γ we simply replace $p(i)$ by the i-th share (which may consist of several field elements, depending on the expansion factor) of a linear secret sharing of 1, with the randomness chosen so that the "right" shares (i.e, those corresponding to the x_i's that the prover does not know) vanish.

Note that an honest prover knows $(x_i)_{i \in S}$ for some $S \in \Gamma$. Hence, $S^c \notin \Gamma^*$ and for this reason the appropriate secret sharing of 1 exists, showing completeness of the generalized proof of partial knowledge.

Special soundness follows from the following observation. Let $A \subset \{1, \ldots, n\}$ be the subset for which all the corresponding shares vanish. Then, by linearity of the secret sharing scheme and since the secret sharing reconstructs to 1, it follows that $A \notin \Gamma^*$. Hence, $A^c \in \Gamma$ and special soundness follows as before.

The communication complexity of the resulting protocol depends logarithmically on the size of the LSSS for Γ^*, which is given by the monotone-span-program complexity of Γ^* [SJM91] and which coincides with the monotone-span-program complexity of Γ [Gál95].

6 Acknowledgements

Thomas Attema has been supported by EU H2020 project No 780701 (PROMETHEUS) and by the Vraaggestuurd Programma Veilige Maatschappij, supervised by the Innovation Team of the Dutch Ministry of Justice and Security. Ronald Cramer has been supported by ERC ADG project No 74079 (ALGSTRONGCRYPTO) and by the NWO Gravitation Programme (QSC).

References

AC20a. Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. *IACR Cryptol. ePrint Arch.*, 2020:152, 2020.

AC20b. Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In *CRYPTO*, Lecture Notes in Computer Science. Springer, 2020.

BAZB20. Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. In *Financial Cryptography*, Lecture Notes in Computer Science. Springer, 2020.

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society, 2018.

BCC+15. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *ESORICS (1)*, volume 9326 of *Lecture Notes in Computer Science*, pages 243–265. Springer, 2015.

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016.

BdM93. Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285. Springer, 1993.

BG13. Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 646–663. Springer, 2013.

Cam97. Jan Camenisch. Efficient and generalized group signatures. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 465–479. Springer, 1997.

CD98. Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441. Springer, 1998.

CDN15. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

CDS94. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

Cra96. Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, 1996.

CvH91. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

Dia20. Benjamin E. Diamond. "Many-out-of-Many" proofs with applications to anonymous Zether. *IACR Cryptol. ePrint Arch.*, 2020:293, 2020.

ESS+19. Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 67–88. Springer, 2019.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

Gál95. Anna Gál. *Combinatorial methods in Boolean function complexity*. PhD thesis, University of Chicago, 1995.

GK15. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280. Springer, 2015.

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC*, pages 291–304. ACM, 1985.

HBHW20. Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. *Zcash Protocol Specication - Version 2020.1.7*, 2020.

Jiv19. Aram Jivanyan. Lelantus: Towards confidentiality and anonymity of blockchain transactions from standard assumptions. *IACR Cryptol. ePrint Arch.*, 2019:373, 2019.

JM20. Aram Jivanyan and Tigran Mamikonyan. Hierarchical one-out-of-many proofs with applications to blockchain privacy and ring signatures. *IACR Cryptol. ePrint Arch.*, 2020:430, 2020.

Lin03. Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *J. Cryptology*, 16(3):143–184, 2003.

Mer80. Ralph C. Merkle. Protocols for public key cryptosystems. In *IEEE Symposium on Security and Privacy*, pages 122–134. IEEE Computer Society, 1980.

MGGR13. Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*, pages 397–411. IEEE Computer Society, 2013.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

Sha79. Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

SJM91. Gustavus J. Simmons, Wen-Ai Jackson, and Keith M. Martin. The geometry of shared secret schemes. *Bulletin of the Institute of Combinatorics and its Applications*, 1:71–88, 1991.

STY00. Tomas Sander, Amnon Ta-Shma, and Moti Yung. Blind, auditable membership proofs. In *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 53–71. Springer, 2000.