On Sigma-Protocols and (packed) Black-Box Secret Sharing Schemes

Claudia Bartoli[1,2] and Ignacio Cascudo[1⋆]

[1]IMDEA Software Institute, Madrid, Spain
{claudia.bartoli,ignacio.cascudo}@imdea.org
[2]Universidad Politécnica de Madrid, Madrid, Spain

Abstract. Σ-protocols are a widely utilized, relatively simple and well understood type of zero-knowledge proofs. However, the well known Schnorr Σ-protocol for proving knowledge of discrete logarithm in a cyclic group of known prime order, and similar protocols working over this type of groups, are hard to generalize to dealing with other groups. In particular with hidden order groups, due to the inability of the knowledge extractor to invert elements modulo the order.

In this paper, we introduce a universal construction of Σ-protocols designed to prove knowledge of preimages of group homomorphisms for any abelian finite group. In order to do this, we first establish a general construction of a Σ-protocol for $\mathfrak{R}$-module homomorphism given only a linear secret sharing scheme over the ring $\mathfrak{R}$, where zero knowledge and special soundness can be related to the privacy and reconstruction properties of the secret sharing scheme. Then, we introduce a new construction of 2-out-of-n packed black-box secret sharing scheme capable of sharing k elements of an arbitrary (abelian, finite) group where each share consists of $k + \log n - 3$ group elements. From these two elements we obtain a generic "batch" Σ-protocol for proving knowledge of k preimages of elements via the same group homomorphism, which communicates $k + \lambda - 3$ elements of the group to achieve $2^{-\lambda}$ knowledge error.

For the case of class groups, we show that our Σ-protocol improves in several aspects on existing proofs for knowledge of discrete logarithm and other related statements that have been used in a number of works.

Finally, we extend our constructions from group homomorphisms to the case of ZK-ready functions, introduced by Cramer and Damgård in Crypto 09, which in particular include the case of proofs of knowledge of plaintext (and randomness) for some linearly homomorphic encryption schemes such as Joye-Libert encryption. However, in the case of Joye-Libert, we show an even better alternative, using Shamir secret sharing over Galois rings, which achieves 2^{-k} knowledge soundness by communicating k ciphertexts to prove k statements.

Keywords: Sigma Protocol, Black-Box Secret Sharing Schemes, Batch Proofs.

1 Introduction

Σ-protocols are one of the most well known and understood types of zero-knowledge proofs. Their simplicity and concrete efficiency makes them widely used in various cryptographic applications and protocols, such as digital signatures, group signatures or anonymous credential systems, as well as secure multiparty computation protocols.

⋆ This work has been partially supported by the grant PIPF-2022/COM-25517, funded by the Madrid Regional Government, and by the projects SecuRing (PID2019-110873RJ-I00/MCIN/AEI/10.13039/501100011033), PRODIGY (TED2021-132464B-I00) funded by MCIN/AEI/10.13039/501100011033/ and the European Union NextGenerationEU/PRTR, and CONFIDENTIAL-6G funded by the European Union (GA 101096435). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

One of the best examples of Σ-protocols is Schnorr's proof of knowledge of discrete logarithm [33]. Given a cyclic group $\mathbb{G} = \langle G \rangle$ of (large) *known* prime order p, and $X \in \mathbb{G}$, the prover claims to a verifier that she knows $w \in \mathbb{Z}_p$ with $X = wG$. To prove it, she samples $r \in \mathbb{Z}_p$ at random and sends $A = rG$, the verifier replies with a uniformly random challenge $c \in \mathbb{Z}_p$, and the prover "opens" the linear combination $z = r + cw$; the verifier accepts if $zG = A + cX$.

This Σ protocol is a zero-knowledge proof of knowledge (ZKPoK) of w with knowledge error $1/p$. However this relies on the ability of inverting the difference $c - c'$ of any two different challenges, which is possible because the group order is known and prime. Namely, from conversations (A, c, z) and (A, c', z') the witness can be extracted as $w = (c - c')^{-1}(z - z')$.

This makes it hard to adapt Schnorr's protocol to other groups and in particular *hidden order groups*, where inverting $c - c'$ modulo the order of the group may not be feasible or even possible. For example, consider the task of proving knowledge of a discrete logarithm in a *class group*, which has an order assumed to be hard to compute. In this scenario existing Σ-protocols either: *i)* resort to using Schnorr's proof with binary challenges (which by itself only yields soundness error $1/2$) repeating it many times to reduce the soundness error, which is somewhat inefficient [9]; *ii)* rely on hardness assumptions which in particular need the basis G to be uniformly random, which makes them harder to use in protocols [10]; or *iii)* are only sound proofs and not proofs of knowledge, in addition to relying on somewhat less studied assumptions [7].

A similar problem arises when proving plaintext knowledge in the Joye-Libert encryption scheme [27], which operates on RSA groups where the plaintext is encoded in a subgroup of $\mathbb{Z}_N^*$ of order 2^l. Mon$\mathbb{Z}_{2^k}$a [14], a protocol for MPC for arithmetic circuits over $\mathbb{Z}_{2^l}$, resorts to extracting only part of the witness, which leads to overheads in the protocol; and a recent threshold ECDSA protocol from [34], facing this same problem, circumvents it constructing a more involved protocol that uses a modified Joye-Libert scheme instead.

The motivation for this work is to generalize Schnorr's protocol, in a manner that it can be extended to a larger family of groups, starting from the observation that Schnorr's protocol can be interpreted in terms of *Shamir secret sharing* in the following sense: with her first message, the prover is committing to randomness r for a degree-1 Shamir sharing of the witness w; the challenge implicitly specifies a share-index; the share z corresponding to that index is sent in the last message by the prover. The secret sharing scheme is a variant of Shamir secret sharing where shares are evaluations of $f(T) = w \cdot T + r \in \mathbb{Z}_p[T]$ at points of $\mathbb{Z}_p$, and the secret is the evaluation at "the point at infinity". The usual security properties of special-soundness and honest-verifier zero-knowledge can be interpreted in terms of 2-reconstruction and 1-privacy of the scheme, respectively.

The observation about this connection between Schnorr's protocol and (1-private, 2-reconstruction)-secret sharing schemes was made in the introduction of [15] (and its journal version [16]). More loosely, it can be related to the MPC-in-the-head paradigm for zero knowledge introduced in [26], in the sense that the prover is implicitly creating views of an MPC protocol (in this case a secret sharing) for virtual parties, committing to them, and then revealing one of them on demand.

More recently, in concurrent work to this one (to appear in Asiacrypt23), [35] developed this connection between secret sharing and Σ- protocols, providing a general construction of Σ-protocols from *verifiable* secret sharing. They subsequently use it to build commit-and-proof arguments for general, non necessarily algebraic, statements using an MPC-in-the-head based approach.

Our work, while starting with the same observation as [35], focuses on improving the *concrete* complexity of Σ-protocols for *algebraic* statements, more concretely proofs of knowledge of preimages of elements via group homomorphisms, later extending this to ZK-ready functions, a notion introduced in [15]. We present a general construction of Σ-protocols for these types of relations from *linear* secret sharing schemes (LSSS), where the properties of special honest-verifier zero knowledge and (ν-)special soundness are based on the linearity, privacy and reconstruction of the linear secret sharing scheme.

In the case of proofs of knowledge of homomorphism-preimage involving groups of known prime order, using degree-1 Shamir secret sharing scheme (1-privacy, 2-reconstruction) leads to Schnorr's protocol in the case of discrete logarithm (and the generalization in [30] for other group homomorphisms); using the Franklin-Yung [23] "packed" (also called "ramp") version of Shamir with 1-privacy, $k+1$-reconstruction and k secrets leads to the batched Schnorr protocol from [24]. These seem to be the best options in this prime-order group scenario.

However, in the case of hidden groups, our construction can be instantiated to yield new results of batched ZKPoKs with improved complexity. By instantiating our construction with a new family of *black-box secret sharing* (BBSS) schemes that we introduce, we obtain Σ-protocols that can be used over any finite abelian group (regardless of its order or structure). In this case, we obtain improved amortized communication complexity with respect to the Σ-protocols in [15,16].

When applied to algebraic statements on class groups (e.g. ZKPoK of discrete logarithm), our protocol improves on previous alternatives [9,10,7] in either amortized complexity, in the first case, or lack of reliance on additional assumptions in the other two (while still matching the amortized complexity of these protocols).

While this black-box secret sharing approach also yields batch ZKPoKs of plaintext knowledge of Joye-Libert ciphertexts, in that case we show a better alternative using Shamir secret sharing over Galois rings. This approach is essentially the one used in [2] to prove knowledge of opening of a vector commitment that was introduced there, but we have not seen it being mentioned explicitly for the case of Joye-Libert and moreover, we provide a more generalized construction that allows more tradeoffs between communication complexity and soundness. We detail these contributions below.

1.1 Contributions

Σ-protocols for knowledge of homomorphism preimage from monotone span programs Given a ring $\mathfrak{R}$, modules $\mathfrak{M}_1$, $\mathfrak{M}_2$ over $\mathfrak{R}$, and a $\mathfrak{R}$-module homomorphism $F : \mathfrak{M}_1 \to \mathfrak{M}_2$, our first goal is to describe a $\Sigma-$protocol for the language $\{(\boldsymbol{w}, \boldsymbol{x}) \in \mathfrak{M}_1^k \times \mathfrak{M}_2^k : F(w_i) = x_i \ \forall i \in [k]\}$ using $\mathfrak{R}$-linear secret sharing schemes, see Theorem 1. This includes the case of group homomorphisms, where $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are two finite abelian groups: we can always see these as modules over $\mathfrak{R} = \mathbb{Z}$, and in cases such as cyclic groups of known order m, they are furthermore modules over $\mathfrak{R} = \mathbb{Z}_m$.

Our construction uses the language of monotone span programs [29] because it allows us to capture simultaneously linear secret sharing schemes over different domains (namely $\mathfrak{M}_1$ and $\mathfrak{M}_2$) as long as they are modules over the same ring. In Section 2 we introduce a definition of k-monotone span programs over a ring $\mathfrak{R}$ for a monotone access structure (Δ, Γ) which yields, for every $\mathfrak{R}$-module $\mathfrak{M}$, $\mathfrak{R}$-LSSS where secrets are vectors of k elements in $\mathfrak{M}$, each share may be one or more elements in $\mathfrak{M}$, and such that there is respectively privacy and reconstruction for (at least) all sets in Δ and Γ.

The structure of the Σ-protocol mimics the idea described above for Schnorr's protocol: namely, the prover chooses randomness in $\mathfrak{M}_1$, sends the images of the randomness via F, receives a challenge specifying one or more share-indices and replies with the corresponding shares of the witness using the randomness committed to in the first message.

In Section 3 we prove that the protocol has zero-knowledge as long as the set $\mathcal{C}$ of possible challenges is contained in Δ. For special-soundness, we introduce the notion of *extraction number* $\nu(\mathcal{C}, \Gamma)$ of the challenge set $\mathcal{C}$ with respect to Γ, which is the minimum ν for which the union of any ν challenges is in Γ. We show that the Σ-protocol has $\nu(\mathcal{C}, \Gamma)$-special soundness, leading to a knowledge error upper bounded by $(\nu - 1)/|\mathcal{C}|$. This bound cannot be improved in general.

New "packed" Black-Box Secret Sharing Schemes with small shares In the case of homomorphisms involving groups of large prime order p (exponential in the security parameter), which are modules over $\mathbb{Z}_p$, using 1-private Shamir's scheme for $k = 1$ and their packed variant for larger k is optimal: in that case,

one has a scheme with an exponential number of shares, so $\mathcal{C}$ is of exponential size and we obtain negligible soundness error already for challenges of size 1.

However, the situation is more interesting in the case of groups of unknown order. In this case, we can instantiate our protocol using the notion of black-box secret sharing [22]. A black-box secret sharing scheme can be applied to *any* abelian finite group $\mathbb{G}$ obliviously to its order and structure: sharing and reconstruction only use black-box access to the group operation, inversion and random sampling of elements. The notion is equivalent to that of a monotone span program over $\mathbb{Z}$. Note that Shamir secret sharing "over the integers" is not a threshold black box secret sharing scheme since it only allows to reconstruct a *multiple* of the secret instead of the secret itself.

It is known [18] that threshold black box secret sharing schemes require shares of average size $\log n$ group elements, where n is the number of parties. A line of work [18,19,20] has led to the construction of general-threshold (t-privacy, $t+1$-reconstruction) black box secret sharing schemes essentially matching the bound. However, in those constructions the secret is just one element of $\mathbb{G}$. Furthermore the aforementioned constructions have one important caveat for its use in our Σ-protocols: in all these constructions the computation *of just one share* has complexity linear in n. This means that, unlike in the prime order groups case, we cannot set n to be exponential in the security parameter.

The situation is better in the specific case of 1-privacy and 2-reconstruction, which we denote $(1, 2, n)$-BBSSS. A family of $(1, 2, n)$-BBSSS with $n = 2^k$, secrets in $\mathbb{G}^k$ and shares in $\mathbb{G}^{2k-1}$ appears implicit in [15], where they use it exactly in the same way as in our Σ-protocol.

In Section 4 we show that this construction can be generalized and improved. First, we show that for general $k, n > 0$ (i.e. not necessarily $n = 2^k$) the construction above can be generalized to obtain $(1, 2, n)$-BBSSS with secrets in $\mathbb{G}^k$ and every share in $\mathbb{G}^{h_*}$ where $h_* = k + \lceil \log n \rceil - 1$. Next we show that this can be improved to $h_* = k + \log n - 2$ if $k = 0 \mod 2$ and $n = 4^m$ for some $m > 0$, and to $h_* = k + \log n - 3$ if $k = 0 \mod 3$ and $n = 8^m$ for some $m > 0$. Moreover, in both [15] and our construction, the shares are computed efficiently using a matrix with entries in $\{-1, 0, 1\}$ making them suitable for our Σ-protocols.

Batched proofs of knowledge for statements on class groups for statements over class groups In Section 5 we apply our Σ-protocol construction from Section 3 together with the black box secret sharing scheme from Section 4 to obtain new protocols for proving batches of statements on class groups. We illustrate these with the examples of ZKPoK of discrete logarithm and ZKPoK of plaintext and randomness corresponding to ciphertexts under the CL-HSM encryption [11], but our results can easily be extended to other types of statements: discrete logarithm equality, correct multiplication of ciphertexts etc.

With this we obtain ZKPoK for those relations that compare positively (in an amortized way) with the ones we know of in the literature. They are more efficient in communication and computation than the proofs using binary challenges from [9], and have similar complexity to the proofs in [10,7] but *do not require additional assumptions* (in particular they can be applied in protocols where the prover might have chosen the basis of the discrete logarithm, unlike [10]), *and they are proofs of knowledge*, as opposed to just sound proofs like [7]. We remark that [7] also contain proofs of plaintext knowledge for a CL-HSM encryption, where the prover shows knowledge of only the plaintext and not the randomness. However, this proof (and also the one in [10]) only works for the version of CL-HSM where the plaintext space $\mathbb{Z}_m$ is such that m is a large prime (or a product of large primes) and would not work directly for more general m, in particular, for the case considered in [13] where $m = 2^u$. Since our PoK work regardless of the factorization of m, as far as we know our proofs are the most efficient proofs of plaintext knowledge (again in an amortized sense) for the more general version of CL-HSM.

Extension for ZK-ready functions and batched PoKs for Joye-Libert We can also extend our construction to deal with proving knowledge of preimage of group elements via *ZK-ready functions*, see Theorem 8. These capture "almost-group homomorphisms" that arise in particular as encryption functions

of schemes such as Paillier [32] or Joye-Libert[27]. Considered as maps that take as argument a plaintext message in an additive group U and a random element in a multiplicative group S, and output a ciphertext in a third multiplicative group X, these encryption functions are homomorphic only "up to a correction factor in the S-argument", meaning that $f(u,s) \cdot f(u',s') = f(u+u', s \cdot s' \cdot \delta(u,u'))$ for some δ depending only on u and u'. We extend our Σ-protocols to deal with this case in a similar way as is done in [15]. In the third message the prover needs to adjust the "S-part of the share" to account for the above phenomenon, since she is opening a linear combination of shares. The one technical difference that we find with the homomorphism case is that because of this correction factor, proving zero-knowledge seems to require additional properties of "share uniformity" and "randomness-uniqueness", see Theorem 8, but these are satisfied by all our BBSSS.

However, in the case of Joye-Libert encryption scheme we can obtain improved packed ZKPoK by using the fact that $(\mathbb{Z}_{2^l})^d$ is isomorphic to the Galois ring $GR(2^l, d)$. A Shamir secret sharing scheme over that Galois ring can then allocate up to 2^d participants, and using this as a LSSS over $\mathbb{Z}_{2^l}$ we obtain batched Σ-protocols for proving knowledge of plaintext for k ciphertexts with 2^{-k} soundness error, which communicate only k elements in $\mathbb{Z}_{2^l}$ and $\mathbb{Z}_N$. Moreover, tradeoffs are possible if we use packed versions of Shamir instead.

1.2 Related work and open questions

As we mentioned earlier, a concurrent paper [35] observed that Schnorr's protocol, as well as other Σ-protocols such as "batched Schnorr" [24], Okamoto [31] and Guillou-Quisqater [25], can be interpreted in terms of secret sharing as described above. Nevertheless, the direction of their work from there on is then different than ours, as they go to investigate proofs of statements containing a non-algebraic component from verifiable secret sharing, using an MPC-in-the-head-like technique. In their work, the protocol is applied to validate statements containing a non-algebraic component, for which they employ MPC-in-the-head. On the other hand we use this connection between secret sharing and Σ-protocols to improve their concrete complexity for some classes of algebraic statements, as we have discussed. Another difference is that the general construction of their work uses Verifiable Secret Sharing (VSS) and the Σ-protocol check is done through the share correctness verification algorithm. Our Σ-protocols can be seen as a special case: cast in their framework, we implicitly define a VSS using the fact that MSPs induce linear secret sharing schemes in both the domain $\mathfrak{M}_1$, and the range $\mathfrak{M}_2$ of the homomorphism F; the verifier checks the validity of a share from $\mathfrak{M}_1$ by comparing it with the share in $\mathfrak{M}_2$, which can be computed by the verifier himself, since he has received the randomness in the first message.

Another related concurrent work is that from [4], which defines the notion of generalized special-soundness given by an access structure Γ, or more precisely, by the set of qualified sets of the structure. In their work Γ collects the sets of challenges that allow to extract a secret from the corresponding transcripts. In our case, when our challenge sets $\mathcal{C}$ are all singletons, i.e. the challenges specify exactly one share-index, our access structure coincides with their Γ: in their language, our scheme has Γ-out-of-$\mathcal{C}$ special soundness for the family Γ of qualified sets of the secret sharing scheme. In particular, one of the uses of this notion in [4] is to construct efficient knowledge extractors for Σ-protocols with tight knowledge error $(\nu-1)/|\mathcal{C}|$ in cases where this is not implied by usual ν-special soundness because of ν being exponential in the security parameter. In fact, our Theorem 1 requires the extractor to compute efficiently the reconstruction vector corresponding to the union of ν challenge sets (which is not possible if we need to read an exponential number ν of share indices, even if the reconstruction vector has a small number of non-zeros). [4, Theorem 1] removes this restriction and implies our protocol is also a ZKPoK with knowledge error $(\nu-1)/|\mathcal{C}|$ as long as Γ satisfies the assumptions of their theorem.

A different comparison, and the best comparison point for our results, is the work [15] and its journal version [16]. They implicitly provide a $(1,2,n)$-black box secret sharing, and they construct Σ-protocols to simultaneously prove k instances of statements, for $n = 2^k$, at $2k-1$ group elements communication cost. As mentioned before, we improve their construction by first, adding more flexibility to it, where our scheme in fact enables to share k secrets among any number of participants, not necessarily $n = 2^k$; and second, by

improving the amortized communication complexity to $k + \log n - s$ for $k = 0 \mod s$ and $n = 2^{sm}$ when $s = 2$ or 3. This leaves a question open whether one can further improve this to larger values of s.

Another interesting question are whether one can obtain (t, r, n)-black box secret sharing schemes that lead to better parameters in the corresponding secret sharing schemes. Works such as [18,19,20] construct threshold $(t, t+1, n)$-BBSSS with quasioptimal (in n) share size $\log n + c$ group elements for small constants c. However, their constructions only consider secrets which are a single group element and furthermore, constructing a single share requires $O(n)$ computation. Nevertheless, an appropriate packed generalization of these schemes could potentially make them useful for our construction if the setting where $t = \text{poly}(\lambda)$ and $n = \text{poly}(\lambda)$ for security parameter λ, leading to negligible soundness error.

With respect to our applications to batched PoK in hidden order groups, we have already discussed the improvements we obtained in the case of class groups, and more details are given in Section 5.

As noted earlier, our construction of batched proofs of plaintext and randomness knowledge for Joye-Libert ciphertexts follows the steps of the proof of knowledge of opening of certain homomorphic vector commitments over rings in [2]. In that work they use that Σ-protocol as basis for a *compressed* Σ-protocol [3] for proving knowledge of a committed vector satisfying a certain linear constraint. We note that compressed Σ-protocols (and bulletproofs [8], on which this abstraction is based) provide an amortization which is different than ours, where the statements consists on proving knowledge of a vector of the form $\mathbf{w}$ such that $L(\mathbf{w}) = x, Com(\mathbf{w}) = C$ for a linear form L and homomorphic commitment Com, and the proofs become logarithmic in the length of the vector. In contrast we are proving knowledge of $\mathbf{w}$ satisfying $F(w_i) = x_i$ for a group homomorphism (or ZK ready function F). We do not rule out that our results can be combined with the compressed Σ-protocol technique to obtain amortized proofs of knowledge of several openings of commitments constrained to a linear form, as is the case in [2].

The problem of batching proofs of knowledge for Joye-Libert ciphertexts, as mentioned above, has arisen in applications such as multiparty computation [14] and threshold ECDSA [34]. These works have found different ways of circumventing the fact that extraction with a straightforward Schnorr protocol would fail due to the fact that the challenges differences are not invertible. In the former work, the authors resort to a proof of knowledge where only part of the witness can be extracted, which creates overhead in the protocol, as it requires to embed the actual data in a larger ring $\mathbb{Z}_{2^l}$. In the latter, the authors also acknowledge that they need to construct a more involved protocol due to this obstacle. Therefore we expect that our results can lead to improvements in these and other applications.

Lastly, the Black Box Secret Sharing Scheme introduced in this paper is formulated through a matrix with coefficients in $\{-1, 0, 1\}$, as detailed in section 4.3. Hence, it would be worthwhile to explore its potential applications in lattice-based cryptography, where such properties are useful, see e.g. [5].

2 Preliminaries

2.1 General notation

Throughout this work, we denote vectors with bold font (e.g. $\boldsymbol{v}$). A bold font of a function is used to represent the vector resulting from applying the function to each element of another vector, e.g. if $\boldsymbol{v} = (v_1, \ldots, v_n)$, then $\boldsymbol{f}(\boldsymbol{v}) := (f(v_1), \ldots, f(v_n))$. $V^\top$ denotes the transpose of a matrix V. $|\mathcal{C}|$ denotes the cardinality of a set $\mathcal{C}$ and $\log n$ denotes the logarithm in base 2 of n. For $m \leq n$, $[m, n]$ represents the set of integers $\{m, \ldots, n\}$ and for $n \geq 1$ we denote by $[n]$ the set $[1, n] = \{1, \ldots, n\}$. For any $T \subset [n]$ and matrix $M \in \mathfrak{R}^{h \times e}$, for a ring $\mathfrak{R}$ and $h \geq n$, M_T denotes the submatrix of M obtained from the $i-$rows, $i \in T$ and h_T is the number of rows of M_T.

2.2 $\Sigma-$protocols

Let $\mathcal{W}$, $\mathcal{X}$ be two finite sets, and let $R \subseteq \mathcal{W} \times \mathcal{X}$ be a relation. A zero-knowledge proof of knowledge is a protocol between a prover P and a verifier V, both of whom have a common input x and where P wants

to convince V she knows a witness $w \in \mathcal{W}$ for x with respect to R, i.e. $(w, x) \in R$, without revealing any additional information about w. Σ-protocols are zero-knowledge proofs that follow the template in Figure 1, where $\mathcal{C}$ is a finite set, called challenge set.

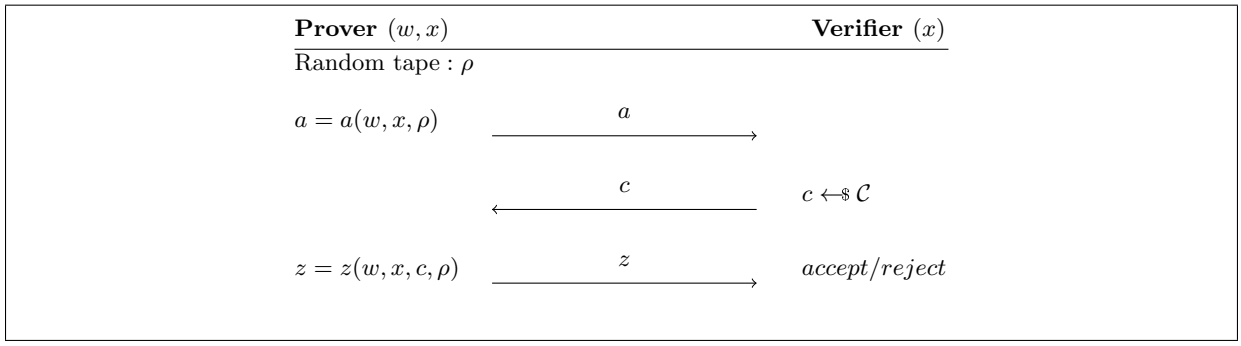

Fig. 1. Σ-protocol template

Definition 1. ***Σ-protocol****: A Σ-protocol for relation R is one that follows the template in Figure 1 and satisfies the properties below:*

- ***Completeness****: if P and V follow the protocol, then V always accepts;*
- ***κ-Special Soundness****: There exists an p.p.t extractor that, given x and κ accepting conversations (a, c_1, z_1),..., (a, c_κ, z_κ), where $c_i \neq c_j$ for all $i, j \in [\kappa]$, efficiently computes w with $(w, x) \in R$;*
- ***Honest-verifier zero-knowledge(HVZK)****: There exists a p.p.t simulator $\mathcal{S}$, which on input $x \in \mathcal{X}$ and $c \in \mathcal{C}$, outputs a "conversation" (a, c, z) with the same distribution as a real conversation between the honest P and V on input x and where V chooses c as a challenge.*

Most definitions in the literature require $\kappa = 2$. However, it is worth to admit larger κ, since in [3] the following result is provided.

Proposition 1 ([3]). *A Σ-protocol with κ-special soundness is a HVZK proof of knowledge with knowledge error at most $(\kappa - 1)/|\mathcal{C}|$.*

2.3 Secret sharing

Definition 2 (Access structure). *Let $n \geq 1$ be an integer, $2^{[n]}$ be the family of all subsets of $[n]$ and let $\Delta, \Gamma \subset 2^{[n]}$. The pair (Δ, Γ) is a* access structure (for $[n]$) *if $\emptyset \in \Delta$, $[n] \in \Gamma$, and $\Delta \cap \Gamma = \emptyset$.*

In addition, (Δ, Γ) is monotone *if the following holds: (i) if $T_1 \in \Delta$ and $T_2 \subset T_1$ then $T_2 \in \Delta$; (ii) if $S_1 \in \Gamma$ and $S_1 \subset S_2$ then $S_2 \in \Gamma$.*

For $0 \leq \mathfrak{t} < \mathfrak{r} \leq n$, the threshold *access structure $(\Delta, \Gamma)_{\mathfrak{t},\mathfrak{r},n}$ is defined by $\Delta = \{T \subseteq [n] : |T| \leq \mathfrak{t}\}$, and $\Gamma = \{S \subseteq [n] : |S| \geq \mathfrak{r}\}$.*

All access structures considered in this work will be monotone.

Definition 3 (Secret sharing). *For the purpose of this paper, a* secret-sharing scheme*(SSS) for a monotone access structure (Δ, Γ) consists of a space of secrets $\mathcal{S}_0$, a space of randomness $\mathcal{R}$, spaces of shares $\mathcal{S}_1, \cdots, \mathcal{S}_n$ and a map $\mathsf{Sh} : \mathcal{S}_0 \times \mathcal{R} \longrightarrow \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, such that, if Sh_A denotes the projection of Sh to $\times_{i \in A} \mathcal{S}_i$:*

– *Every set $S \in \Gamma$ is a* reconstructing set*: for any $a_S \in \times_{i\in S}\mathcal{S}_i$, there exists* at most *one $s \in \mathcal{S}_0$ with $\mathsf{Sh}_S(s,r) = a_S$ for some (possibly non-unique) $r \in \mathcal{R}$;*

– *Every set $T \in \Delta$ is a* privacy set*: for any $a_T \in \times_{i\in T}\mathcal{S}_i$, and any s, s' in $\mathcal{S}_0$ we have $|\{r \in \mathcal{R} : \mathsf{Sh}_T(s,r) = a_T\}| = |\{r' \in \mathcal{R} : \mathsf{Sh}_T(s',r') = a_T\}|$. In other words, conditioned to the shares for T being a_T, every element in $\mathcal{S}_0$ has the same probability of being the secret.*

Definition 4. *A $(\mathfrak{t},\mathfrak{r},n)-$secret-sharing scheme is an SSS for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$.*

In this work we will be considering secret sharing schemes which are linear over certain ring, as defined next.

Definition 5. *Let $\mathfrak{R}$ be a commutative ring with an identity, and $\mathfrak{M}$ a finite module over $\mathfrak{R}$. A linear secret sharing scheme (LSSS) over $\mathfrak{R}$ is a SSS with $\mathcal{S}_0 = \mathfrak{M}^k$, $\mathcal{R} = \mathfrak{M}^e$, $\mathcal{S}_i = \mathfrak{M}^{h_i}$ for $i \in [n]$ where k, e, h_i are positive integers, and $\mathsf{Sh} : \mathfrak{M}^k \times \mathfrak{M}^e \to \times_{i=1}^n \mathfrak{M}^{h_i}$ is given by a* sharing matrix *$M \in \mathfrak{R}^{h\times(k+e)}$ (where $h = \sum_{i=1}^n h_i$). Namely $(\boldsymbol{\sigma}_1,\ldots,\boldsymbol{\sigma}_n)^\top = \mathsf{Sh}(\boldsymbol{s},\boldsymbol{r}) = M(\boldsymbol{s},\boldsymbol{r})^\top$.*

The same matrix $M \in \mathfrak{R}^{h\times(k+e)}$ can define secret sharing schemes for different instances of $\mathfrak{M}$. Seeing M through the lens of *Monotone Span Programs*, which we detailed next, we can capture properties of reconstruction and privacy that apply to every secret sharing scheme defined by M, regardless of the module $\mathfrak{M}$.

2.4 Monotone Span Programs

Monotone Span Programs (MSP) were introduced in [28] and are closely related to LSSS. We provide a slight generalization that endows MSPs with k linearly independent target vectors (rather than just one as in [28]) and detail its relation to LSSS with secrets of size k.

Definition 6 (k-Monotone Span Program). *Let $k, n \geq 1$ be integers, and let (Δ,Γ) be a monotone access structure for $[n]$. A* k-Monotone Span Program (k-MSP) *$\mathcal{M}$ over a ring $\mathfrak{R}$ computing (Δ,Γ) is a quadruple $(\mathfrak{R}, M, \Psi, k)$ with $M \in \mathfrak{R}^{h\times(k+e)}$, $e \geq 0$, $h \geq n$ and $\Psi : [h] \to [n]$ surjective, that satisfies the two properties (P_1) and (P_2) below.*

– *(P_1) for every $T \in \Delta$, there exist vectors $\boldsymbol{\lambda}_T^{(1)}, ..., \boldsymbol{\lambda}_T^{(k)} \in \mathfrak{R}^{k+e}$ with:*

- *$\boldsymbol{\lambda}_T^{(i)} = (\overbrace{0,\ldots,0}^{i-1},1,\overbrace{0,\ldots,0}^{k-i},*,\ldots,*)$, i.e. the projection of $\boldsymbol{\lambda}^{(i)}$ to the first k components is the i-th unit vector.*
- *$\boldsymbol{\lambda}_T^{(i)} \in Ker\ M_T$, i.e., $M_T \cdot {\boldsymbol{\lambda}^{(i)}}^\top = {\mathbf{0}_{h_T}}^\top$.*

– *(P_2) for any $S \in \Gamma$, for all $i \in [k]$, $\boldsymbol{\mu}^{(i)} := (\overbrace{0,\ldots,0}^{i-1},1,\overbrace{0,\ldots,0}^{k+e-i}) \in \mathrm{Im}(M_S^\top)$.*
I.e., there exist vectors $\boldsymbol{\rho}_S^{(i)} \in \mathfrak{R}^{h_S}$, $i = 1,\ldots,k$ (called "reconstruction vectors") that satisfy the equations $\boldsymbol{\rho}_S^{(i)} \cdot M_S = \boldsymbol{\mu}^{(i)}$.

Notation: In some cases we will define a MSP from a collection $M_i \in \mathfrak{R}^{h_i\times(k+e)}$, $i \in [n]$, hence we write $\mathcal{M} = (\mathfrak{R}, \{M_i\}_{i\in[n]}, k)$ meaning M is defined by stacking the blocks M_i in the rows $\Psi^{-1}(\{i\})$.

Monotone Span Programs and Secret Sharing. The following is a direct generalization of a result in [29], see Section A of the Appendix for a proof.

Proposition 2. *Let $\mathcal{M}$ be a k-MSP over a ring $\mathfrak{R}$. If $\mathcal{M}$ computes (Δ,Γ) then every $T \in \Delta$ is a privacy set, and every $S \in \Gamma$ is a reconstructing set of every LSSS over $\mathfrak{R}$ with sharing matrix M.*

3 Σ-protocols from Secret Sharing Schemes

In this section we present a general construction of $\Sigma-$protocols from $\mathfrak{R}$-linear secret sharing schemes (k-MSPs over $\mathfrak{R}$). The Σ-protocols apply to relations of the form $R = \{(\boldsymbol{w}, \boldsymbol{x}) \in \mathfrak{M}_1^k \times \mathfrak{M}_2^k : F(w_i) = x_i \; \forall \, i \in [k]\}$, where $\mathfrak{M}_1, \mathfrak{M}_2$ are modules over $\mathfrak{R}$, and $F : \mathfrak{M}_1 \to \mathfrak{M}_2$ is an (efficiently-computable) $\mathfrak{R}$-linear map. Recalling our notation that $\boldsymbol{F}(\boldsymbol{w}) := (F(w_1), \ldots, F(w_k))$, the relation is abbreviated as $\boldsymbol{F}(\boldsymbol{w}) = \boldsymbol{x}$.

Definition 7. *A* challenge set $\mathcal{C}$ compatible with *the access structure* (Δ, Γ) *is a non-empty subfamily* $\emptyset \neq \mathcal{C} \subseteq \Delta$ *(i.e. a family of* $E \subseteq [n]$ *such that all* $E \in \Delta$*). The* extraction number *of* $\mathcal{C}$ *with respect to* (Δ, Γ) *is*

$$\nu(\mathcal{C}, \Gamma) := min\{\nu > 0 : \bigcup_{i=1}^{\nu} E_i \in \Gamma \; \forall \text{ pairwise distinct } E_1, \ldots, E_\nu \in \mathcal{C}\}.$$

The following theorem showcases our $\Sigma-$protocol construction.

Theorem 1. *Let* $\mathfrak{R}$ *be a ring and* $\mathfrak{M}_1, \mathfrak{M}_2$ *be finite* $\mathfrak{R}-$*modules. We assume we can compute the action of* $\mathfrak{R}$ *on* $\mathfrak{M}_1$, $\mathfrak{M}_2$ *efficiently. Let* $F : \mathfrak{M}_1 \to \mathfrak{M}_2$ *be an* $\mathfrak{R}-$*module homomorphism. Let* $\mathcal{M} = (\mathfrak{R}, M, \Psi, k)$ *be a* $k-$*MSP over* $\mathfrak{R}$ *computing* (Δ, Γ)*, where* $M \in \mathfrak{R}^{h \times (k+e)}$*.*

Let $\mathcal{C}$ *be a compatible challenge set with respect to* (Δ, Γ) *and* $\nu = \nu(\mathcal{C}, \Gamma)$ *its extraction number. Assume it is possible to sample efficiently uniformly from* $\mathcal{C}$ *and* $\mathfrak{M}_1$*, that for every* $E \in \mathcal{C}$*,* M_E *can be computed efficiently, and that for any pairwise distinct* $E_1, \ldots, E_\nu \in \mathcal{C}$ *it is efficient to compute the reconstruction vectors* $\boldsymbol{\rho}_{\mathcal{E}}^{(i)}$*, where* $\mathcal{E} = \bigcup_{i=1}^{\nu} E_i$*, promised by Definition 6.*

Then Figure 2 provides a Σ*-protocol with* ν*-special soundness and perfect honest verifier zero-knowledge for* $R = \{(\boldsymbol{w}; \boldsymbol{x}) \in \mathfrak{M}_1^k \times \mathfrak{M}_2^k : F(w_i) = x_i \; \forall i \in [k]\}$.

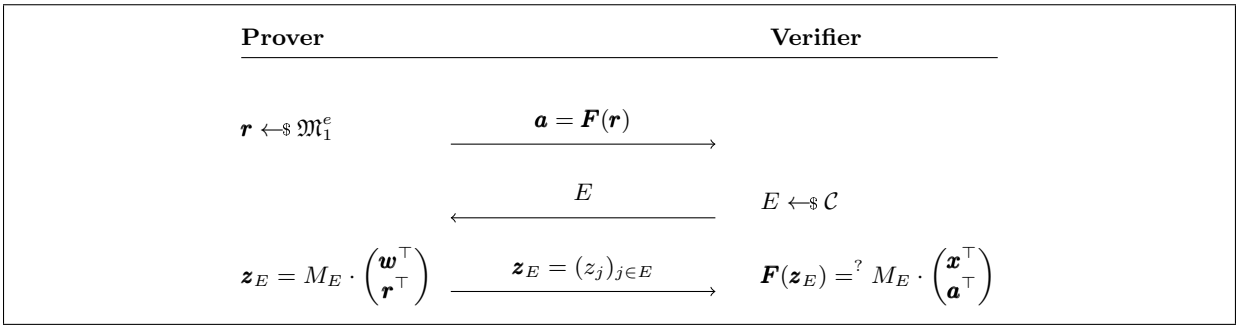

Fig. 2. Σ-protocol from k-MSP

Proof. **Completeness:** Since F is a $\mathfrak{R}$-linear map, it satisfies $F(a \cdot m) = a \cdot F(m)$ and $F(m+n) = F(m) + F(n)$ for $a \in \mathfrak{R}$, $m, n \in \mathfrak{M}_1$, and hence, $\boldsymbol{F}(A \cdot \boldsymbol{m}^\top) = A \cdot \boldsymbol{F}(\boldsymbol{m})^\top$ for any matrix A over $\mathfrak{R}$ and vector $\boldsymbol{m}$ over $\mathfrak{M}_1$ of matching dimensions. Therefore, for the challenge E and $\boldsymbol{z}_E = M_E \cdot \boldsymbol{v}^\top$ we have the equality $\boldsymbol{F}(\boldsymbol{z}_E) = \boldsymbol{F}(M_E \cdot \boldsymbol{v}^\top) = M_E \cdot \boldsymbol{F}(\boldsymbol{v})^\top$ where $\boldsymbol{F}(\boldsymbol{v}) = (\boldsymbol{F}(\boldsymbol{w}), \boldsymbol{F}(\boldsymbol{r})) = (\boldsymbol{x}, \boldsymbol{a})$.

ν-Special Soundness: We want to prove that there is an extractor that, given ν accepted conversations of the form $(\boldsymbol{a}, E_i, \boldsymbol{z}_{E_i})$ for $i \in [\nu]$ with same vector $\boldsymbol{a} \in \mathfrak{M}_2^e$ and different challenges E_i, reconstructs the secrets $w_1, ..., w_k$. Let $\mathcal{E} = \cup_{j=1}^{\nu} E_j$, we define a vector $\boldsymbol{z}_{\mathcal{E}}$ with coordinates indexed by $\mathcal{E}$ as follows: for each $c \in \mathcal{E}$, c is in some E_j. The extractor chooses one such E_j, and defines the c-th coordinate z_c of $\boldsymbol{z}_{\mathcal{E}}$ to be the corresponding coordinate of $\boldsymbol{z}_{E_j}$ (the extractor will work even if different $\boldsymbol{z}_{E_j}$ disagree in a common c-th coordinate). Since the conversations above are accepted, we have $F(z_c) = M_c \cdot (\boldsymbol{x}, \boldsymbol{a})^\top$ for all $c \in \mathcal{E}$ and

therefore $\boldsymbol{F}(\boldsymbol{z}_{\mathcal{E}}) = M_{\mathcal{E}} \cdot (\boldsymbol{x}, \boldsymbol{a})^\top$. By assumption, $\mathcal{E}$ is a reconstruction set for the MSP, consequently, for each $j \in [k]$, there is a reconstruction vector $\boldsymbol{\rho}^j_{\mathcal{E}} \in \mathfrak{R}^{|\mathcal{E}|}$ such that $x_j = \boldsymbol{\rho}^j_{\mathcal{E}} \cdot \boldsymbol{F}(\boldsymbol{z}_{\mathcal{E}})^\top$. Set $w_j = \boldsymbol{\rho}^j_{\mathcal{E}} \cdot \boldsymbol{z}^\top_{\mathcal{E}}$, then $F(w_j) = \boldsymbol{F}(\boldsymbol{\rho}^j_{\mathcal{E}} \cdot \boldsymbol{z}^\top_{\mathcal{E}}) = \boldsymbol{\rho}^j_{\mathcal{E}} \cdot \boldsymbol{F}(\boldsymbol{z}_{\mathcal{E}})^\top = x_j$, for all $j \in [k]$.

Honest-verifier zero-knowledge: We prove the existence of a simulator that, given $\boldsymbol{x}$ in the language and a challenge $E \in \mathcal{C}$, produces conversations whose distribution is the same to that of an honest conversation using a witness $\boldsymbol{w}$ for $\boldsymbol{x}$. Here, intuitively, we will use the fact that in the real protocol the shares $\boldsymbol{z}_E$ do not give any information about $\boldsymbol{w}$ because of the privacy of the secret sharing scheme for the set E, so these can be simulated by a sharing of an arbitrary element, for example $\boldsymbol{0}_k := (0, \ldots, 0) \in \mathfrak{M}_1^k$.

Concretely the simulator:

1. Samples $\widehat{\boldsymbol{r}}$ uniformly at random in $\mathfrak{M}_1^e$ and sets $\widehat{\boldsymbol{z}}_E = M_E(\boldsymbol{0}_k, \widehat{\boldsymbol{r}})^\top$.
2. For $i \in [k]$ let $\boldsymbol{\lambda}^{(i)}$ be an (arbitrary) element from the space

$$\Lambda_{E,i} = \{\boldsymbol{\lambda}^{(i)} \in \mathfrak{R}^{k+e} : (\boldsymbol{\lambda}^{(i)})_i = 1, (\boldsymbol{\lambda}^{(i)})_j = 0 \text{ for } j \in [k], j \neq i,\ {M_E\boldsymbol{\lambda}^{(i)}}^\top = \boldsymbol{0}_E\}$$

   which is non-empty by definition of k-MSP (Definition 6,(P2)). Define $\overline{\boldsymbol{\lambda}}^{(i)}$ to be the projection of $\boldsymbol{\lambda}^{(i)}$ to the last e coordinates.
3. Define $\widehat{\boldsymbol{a}} = \boldsymbol{F}(\widehat{\boldsymbol{r}}) + \overline{\boldsymbol{\lambda}}^{(1)} \cdot x_1 + \cdots + \overline{\boldsymbol{\lambda}}^{(k)} \cdot x_k$ [1]
4. Output $(\widehat{\boldsymbol{a}}, E, \widehat{\boldsymbol{z}}_E)$.

We show that $(\hat{\boldsymbol{a}}, E, \hat{\boldsymbol{z}}_E)$ has the same distribution as a real honest transcript with witness $\boldsymbol{w}$. In fact, $(\hat{\boldsymbol{a}}, E, \hat{\boldsymbol{z}}_E)$ is exactly the real conversation that arises when the prover chooses $\boldsymbol{r} = \widehat{\boldsymbol{r}} + \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} w_i$ as randomness at the beginning of the protocol. Since in the simulation $\widehat{\boldsymbol{r}}$ is chosen uniformly at random, $\boldsymbol{r}$ is also uniformly random.

Indeed, if the prover uses this randomness, the first message of the real conversation is $\boldsymbol{F}(\boldsymbol{r}) = \boldsymbol{F}(\widehat{\boldsymbol{r}} + \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} w_i) = \boldsymbol{F}(\widehat{\boldsymbol{r}}) + \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} x_i = \widehat{\boldsymbol{a}}$, while the third message is $M_E(\boldsymbol{w}, \boldsymbol{r})^\top = M_E(\boldsymbol{w}, \widehat{\boldsymbol{r}} + \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} w_i)^\top = M_E(\boldsymbol{0}, \widehat{\boldsymbol{r}})^\top + M_E(\boldsymbol{w}, \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} w_i)^\top$.

Recall that $\boldsymbol{\lambda}^{(i)} = (0, \ldots, 1, \ldots, 0, \overline{\boldsymbol{\lambda}}^{(i)})$, so $(\boldsymbol{w}, \sum_{i=1}^k \overline{\boldsymbol{\lambda}}^{(i)} w_i) = \sum_{i=1}^k \boldsymbol{\lambda}^{(i)} w_i$. Since ${M_E\boldsymbol{\lambda}^{(i)}}^\top = 0\ \forall i \in [k]$, we conclude $M_E(\boldsymbol{w}, \boldsymbol{r})^\top = M_E(\boldsymbol{0}, \widehat{\boldsymbol{r}})^\top = \widehat{\boldsymbol{z}}_E$. □

Corollary 1. *The protocol in Figure 2 is a Σ-protocol with knowledge error $\leq (\nu - 1)/|\mathcal{C}|$ (from Proposition 1).*

Its average communication complexity is e elements in $\mathfrak{M}_2$ (first message), $\sum_{E_i \in \mathcal{C}} \log |E_i|/|\mathcal{C}|$ bits (second message), $\sum_{E_i \in \mathcal{C}} h_{E_i}/|\mathcal{C}|$ elements in $\mathfrak{M}_1$ (third message).

Example 1. Protocols from Shamir secret sharing and variants As mentioned in the introduction, if $\mathfrak{M}_1 = \mathbb{Z}_p$ and $\mathfrak{M}_2 = \langle g \rangle$ is a group of large prime order p, and $F(w) = g^w$, for the case $k = 1$ (the usual proof of knowledge of DL of one element) we can use $(1, 2, n)$-Shamir's secret sharing scheme with $n \leq p$ and recover Schnorr's protocol. Similarly we can capture known protocols for other languages like discrete log equality. This type of protocols for group homomorphisms of large order have been for example unified under a common framework in [30]. In the case $k > 1$, we obtain the generalization of Schnorr's protocol, we can use a k-MSP for $(\Delta, \Gamma)_{1,k+1,n}$ (again $n \leq p$) where to share $(s_1, \ldots, s_k) \in \mathbb{Z}_p^k$ we sample a random coefficient $r \in \mathbb{Z}_p$ and define the shares as evaluations of $s_1 + s_2 X \cdots + s_k X^{k-1} + r X^k$ in nonzero points of $\mathbb{Z}_p$ (we can include r, the "evaluation at infinity" as an additional share), this is the same construction as in [24]. We obtain a Σ-protocol with $k+1$-special soundness and knowledge soundness k/p. In this case, it does not

[1] Note $\overline{\boldsymbol{\lambda}}^{(i)} \cdot x_i$ denotes the coordinate-wise action of vector $\overline{\boldsymbol{\lambda}}^{(i)} \in \mathfrak{R}^{k+e}$ on the element $x_i \in \mathfrak{M}_2$, that is, if $\overline{\boldsymbol{\lambda}}^{(i)} = (\overline{\lambda}^{(i)}_1, \ldots, \overline{\lambda}^{(i)}_{k+e})$, then $\overline{\lambda}^{(i)} \cdot x_i = (\overline{\lambda}^{(i)}_1 \cdot x_i, \ldots, \overline{\lambda}^{(i)}_{k+e} \cdot x_i)$.

make much sense to use other privacy thresholds, on account of the fact that 1-privacy protocols already lead to negligible soundness if we take p exponential in the security parameter.

Remark 1. – *On exponential number of shares:* Note only M_E, the submatrix of the rows of M corresponding to the challenge E, needs to be computed. Therefore, even if n is exponential in security parameter λ, the protocol can be efficient, as long as M_E can be computed in polynomial time.

– *Threshold access structure:* In Section B.1 of the Appendix we consider the following question: if we fix Δ and Γ to respectively be the sets of at most $\mathfrak{t}$ elements and at least $\mathfrak{r}$ elements, what is the choice of $\mathcal{C}$ that minimizes the knowledge error bound $(\nu-1)/|\mathcal{C}|$? We find out that an optimal choice in that sense is to select $\mathcal{C}$ to be the family of all sets of size exactly $\mathfrak{t}$. However, surprisingly in some cases this is not *the only* choice minimizing this bound and choosing a smaller $\mathcal{C}$ can achieve the same knowledge error, which makes it preferable from the point of view of communication complexity. We prove that optimal choices are characterized by *combinatorial designs.*

– *Optimality of knowledge error bound:* In general the knowledge error bound in Figure 2 is optimal for our protocol: we show in Section B.2 of the Appendix that for the discrete logarithm equality relation a malicious prover breaks soundness with probability $(\nu-1)/|\mathcal{C}|$.

3.1 Non-Interactive $\Sigma-$protocols

The well known Fiat-Shamir heuristic allows to turn the $\Sigma-$protocols from Theorem 1 into a non-interactive argument in the random oracle model. Let $\mathcal{H}:\{0,1\}^* \to \mathcal{C}$ be a random oracle mapping strings into elements of the challenge space. The honest prover constructs $\boldsymbol{a}$ as in the protocol, computes $E=\mathcal{H}(\boldsymbol{x},\boldsymbol{a})$ with the random oracle and $\boldsymbol{z}_E$ as in the protocol and outputs the proof $(\boldsymbol{a},E,\boldsymbol{z}_E)$. The verifier performs the same check as in the protocol, and verifies that $E=\mathcal{H}(\boldsymbol{x},\boldsymbol{a})$, accepting iff both checks pass. This protocol requires the same average communication as in the above corollary, only that now is sent by the prover in one go.

A more interesting question is whether we can perform a commonly used optimization available in Schnorr's protocol and similar ones, consisting on only sending the second and third messages (in our case $(E,\boldsymbol{z}_E)$) in the protocol from Theorem 1. The verifier then reconstructs the first message ($\boldsymbol{a}$) from those messages and the statement $\boldsymbol{x}$ using the verification equation from the Σ-protocol, and finally the verifier checks the random oracle equation $E=\mathcal{H}(\boldsymbol{x},\boldsymbol{a})$.

In our case this works if, given any possible $\boldsymbol{x},E,\boldsymbol{z}_E$, there is a unique $\boldsymbol{a}$ such that $F(\boldsymbol{z}_E)=M_E\cdot(\boldsymbol{x},\boldsymbol{a})^\top$ and this can be efficiently obtained by the verifier. It may be convenient to rephrase the uniqueness condition as follows (which is a consequence of linearity of M): for any $E\in\mathcal{C}$, $\boldsymbol{d}=\boldsymbol{0}$ is the only solution to $\boldsymbol{0}=M_E\cdot(\boldsymbol{0},\boldsymbol{d})^\top$.

In summary, we have:

Prover		**Verifier**
$\boldsymbol{r}\leftarrow\!\!\$\ \mathfrak{M}_1^e, \boldsymbol{a}=\boldsymbol{F}(\boldsymbol{r})$		
$E=\mathcal{H}(\boldsymbol{x},\boldsymbol{a})$		
$\boldsymbol{z}_E=M_E\cdot\begin{pmatrix}\boldsymbol{w}^\top\\ \boldsymbol{r}^\top\end{pmatrix}$	$\xrightarrow{\qquad(E,\boldsymbol{z}_E)\qquad}$	Finds $\boldsymbol{a}$ s.t. $\boldsymbol{F}(\boldsymbol{z}_E)=M_E\cdot\begin{pmatrix}\boldsymbol{x}^\top\\ \boldsymbol{a}^\top\end{pmatrix}$
		$E=^?\mathcal{H}(\boldsymbol{x},\boldsymbol{a})$

Fig. 3. Optimized NI-Σ-protocol from k-MSP

Theorem 2. *In the conditions of Theorem 1, let in addition M satisfy that for any $E\in\mathcal{C}$,*

– $\boldsymbol{d} = \mathbf{0} \in \mathfrak{M}_2^e$ *is the only solution to* $\mathbf{0} = M_E \cdot (\mathbf{0}, \boldsymbol{d})^\top$*;*

– *Given* $\boldsymbol{x} \in \mathfrak{M}_2^k$, $\boldsymbol{y} \in \mathfrak{M}_2^{h_E}$, *computing the unique* $\boldsymbol{a} \in \mathfrak{M}_2^e$ *with* $M_E \cdot (\boldsymbol{x}, \boldsymbol{a})^\top = \boldsymbol{y}$ *is efficient.*

Then assuming the Fiat-Shamir heuristic, the protocol in Figure 3 is a non-interactive zero knowledge proof of knowledge for R, with the same security properties as in Theorem 1, in the random oracle model.

Remark 2. In Section 4 below, we introduce (black box)-secret sharing schemes with 1-privacy (hence $\mathcal{C}$ contains only sets of size 1) where each share is of the form $\boldsymbol{\sigma}_i = N_i \boldsymbol{w} + \boldsymbol{a}$, for a public matrix N_i. In this case, given the index i, the share $\boldsymbol{\sigma}_i$ and the secret $\boldsymbol{w}$, the randomness can be determined uniquely as $\boldsymbol{a} = \sigma_i - N_i \boldsymbol{w}$, and clearly this is an efficient computation as it requires the same operations as constructing a share. Therefore those secret sharing schemes fulfil the additional properties in the theorem above.

4 Packed black-box secret sharing schemes

In this section we present constructions of packed black-box secret sharing schemes with 1-privacy and 2-reconstruction, where our main goal is to optimize the secret-to-share size ratio. We introduce constructions derived from a secret sharing scheme outlined in [15] and make improvements to reduce its share size. The BBSS scheme will be utilized in later applications involving class groups, Section 5. Finally, we will discuss the outcomes when applied on Σ-protocols.

4.1 Background on black-box secret sharing

First introduced by Desmedt and Frankel [22] and further studied in works such as [18,19,20], a black-box secret sharing scheme (BBSSS) is a SSS that can be applied to every finite abelian group, *obliviously to its structure*, i.e., sharing and reconstruction only use black-box access to the group operation and group inverse, as well as random sampling of group elements. As argued in [18], since every abelian group $\mathbb{G}$ is a $\mathbb{Z}$-module, a MSP over $\mathbb{Z}$ computing the access structure (Δ, Γ) yields a black-box secret sharing scheme for that structure, so we can reduce the problem to finding MSPs over $\mathbb{Z}$ for the desired structure.

To the best of our knowledge, previous works have focused on sharing a *single* secret of a group $\mathbb{G}$. [19] provides BBSSS for threshold structures $(\Delta, \Gamma)_{\mathfrak{t},\mathfrak{t}+1,n}$ where $h_i = \lceil \log n \rceil$ for $i \in [n]$, i.e., every share is $\lceil \log n \rceil$ elements of $\mathbb{G}$. This is known to be very close to optimal, as the average share size must be at least $\lfloor \log n \rfloor - 1$ [18]. However, this bound does not rule out that one can share a larger secret "at roughly the same price" (even in the threshold case).

Apart from the fact that these schemes share a single secret, a greater obstacle in using the constructions from the line of work [18,19,20] as a basis for our Σ-protocols is that the computational complexity of *even computing one share* in all those schemes is $\Omega(n)$. That means that Remark 1 does not apply: setting $n = \exp(\lambda)$ would make the computation time be exponential in λ too. Setting $n = \text{poly}(\lambda)$ means that a soundness error of $2^{-\lambda}$ requires to either use a privacy threshold and challenges of size $\Omega(\lambda)$ or (if we want smaller privacy threshold and challenges) to use repetition to amplify soundness, both options incurring still in considerable (although polynomial in λ) communication overhead.

These two issues motivate us to search for (threshold) packed BBSS where the size of the shares does not grow too much in comparison to the secret, and where we can compute each share in time polylog n.

4.2 General framework

For the next constructions, we use the following blueprint: each share will consist of the same number $h_i = h_*$ of group elements, and the corresponding block in the MSP will be of the form $M_i = (N_i | I_{h_* \times h_*}) \in \mathbb{Z}^{h_* \times (k + h_*)}$, where $I_{h_* \times h_*}$ represents the identity matrix of size $h_* \times h_*$. Therefore, the shares of $\boldsymbol{s} \in \mathbb{G}^k$ are $N_i \boldsymbol{s}^\top + \boldsymbol{r}$, $i \in [n]$ (for a uniformly random common $\boldsymbol{r} \in \mathbb{G}^{h_*}$).

First we will set the relation between a family of matrices and the existence of a black box secret sharing scheme in the following theorem.

Theorem 3. *Let $\{N_i\}_{i\in[n]}$ be a collection of matrices with $N_i \in \mathbb{Z}^{h_*\times k}$. Let $M_i := (N_i | I_{h_*\times h_*})$, $i = 1,\ldots,n$, M be the stacking of all these matrices, and let $\mathcal{M} = (\mathbb{Z}, \{M_i\}_{i\in[n]}, k)$.*
If, for all $i \neq j$, $N_i - N_j$ has a left pseudoinverse $R_{ij} \in \mathbb{Z}^{k\times h_*}$ (i.e. $R_{ij}(N_i - N_j) = I_{k\times k}$), then $\mathcal{M}$ is a k-MSP over $\mathbb{Z}$ computing $(\Delta,\Gamma)_{1,2,n}$ and hence a $(1,2,n)$-BBSSS with $\mathcal{S}_0 = \mathbb{G}^k$ and $\mathcal{S}_i = \mathbb{G}^{h_*}$ for $i \in [n]$.

Proof. We argue the properties directly in terms of secret sharing. As mentioned above each share is $\boldsymbol{\sigma}_i = N_i\boldsymbol{s}^\top + \boldsymbol{r}$ where $\boldsymbol{r}$ is uniformly random in $\mathbb{G}^{h_*}$. Hence, each individual share is independent from $\boldsymbol{s}$, and we have 1-privacy. The scheme has 2-reconstruction because for every set $\{i,j\} \subseteq [n]$, there exists R_{ij} such that $R_{ij}(\boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j)^\top = \boldsymbol{s}^\top$. Note here it is crucial R_{ij} has coordinates in $\mathbb{Z}$, so that $R_{ij}(\boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j)^\top$ can be computed with black box access to the group operation and inversion. □

Remark 3. The conditions imply $h_* \geq k$. In the case $h_* = k$, $N_i - N_j$ has an inverse defined over $\mathbb{Z}$ iff $\det(N_i - N_j) = \pm 1$. For the more general case $h_* > k$, if $N_i - N_j$ has a $(k\times k)$-submatrix with determinant ± 1 then it has a left pseudoinverse, but the converse is not necessarily true.

4.3 Constructions of packed $(1,2,n)-$BBSSS

We recall and generalize a black box secret sharing scheme implicit in a Σ-protocol in [15] and its journal version [16]. The construction originally fixed $n = 2^k$ and obtained $h_* = 2k-1$. For our purposes we want more flexibility and show that we can "decouple" both parameters. We present directly our generalization.

We define $\ell = \lceil \log n \rceil$. We identify elements of $[n]$ with pairwise different vectors $i = (i_0,\ldots,i_{\ell-1}) \in \{0,1\}^\ell$. Let the k columns of N_i be shifts of the vector i padded with $k-1$ zeros, as follows:

$$N_i := \begin{pmatrix} i_0 & & \mathbf{0} \\ \vdots & \ddots & \\ i_{\ell-1} & & i_0 \\ & \ddots & \vdots \\ \mathbf{0} & & i_{\ell-1} \end{pmatrix} \in \mathbb{Z}^{(k+\ell-1)\times k}$$

Lemma 1. *For every $i \neq j$, $N_i - N_j$ has an integer left pseudoinverse.*

Proof. Let m be the smallest index in $[0,\ell-1]$ where $i_m \neq j_m$. Then the $k\times k$ submatrix of $N_i - N_j$ containing rows m to $m+k-1$ (where we start indexing rows at 0) is a lower triangular square matrix with its diagonal containing all 1's or -1's, hence with determinant ± 1. □

Combining Lemma 1 with Theorem 3 we have:

Theorem 4. *Let $1 \leq k,n$ be integers. There exists a k-MSP over $\mathbb{Z}$ for $(\Delta,\Gamma)_{1,2,n}$ with $h_i = h_* := k + \lceil \log n \rceil - 1$ for all $i \in [n]$. Consequently there is a packed black-box secret sharing scheme with secrets in $\mathbb{G}^k$, and every share in $\mathbb{G}^{h_*}$. In particular for $n = 2^k$, every share is in $\mathbb{G}^{2k-1}$.*

Next we show that this scheme is *not* always optimal, in terms of share size h_* for given k and n. First, for $k = 2$ and $k = 3$, $n = 2^k$, there are optimal $(1,2,n)$-BBSSS with $h_* = k = \log n$, given by the N_i below. [2]

[2] The matrices have been found by first taking matrices N'_i representing multiplication by different elements of the fields $\mathbb{F}_{2^k}$, which leads to $\det(N'_i - N'_j) = 1 \mod 2$, and then (for $k = 3$) using brute force to fix the cases where $\det(N'_i - N'_j) \neq \pm 1$.

$k = 2$

$$N_1 = \begin{pmatrix} 0\,0 \\ 0\,0 \end{pmatrix},\ N_2 = \begin{pmatrix} 1\,0 \\ 0\,1 \end{pmatrix},\ N_3 = \begin{pmatrix} 0\,1 \\ 1\,1 \end{pmatrix},\ N_4 = \begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix}. \tag{1}$$

$k = 3$

$$N_1 = \begin{pmatrix} 0\,0\,0 \\ 0\,0\,0 \\ 0\,0\,0 \end{pmatrix},\ N_2 = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{pmatrix},\ N_3 = \begin{pmatrix} 0\,1\,0 \\ 0\,0\,1 \\ 1\,1\,0 \end{pmatrix},\ N_4 = \begin{pmatrix} 0\,0\,1 \\ 1\,1\,0 \\ 0\,1\,1 \end{pmatrix}$$

$$N_5 = \begin{pmatrix} 1\,1\,0 \\ 0\,1\,1 \\ 1\,1\,1 \end{pmatrix},\ N_6 = \begin{pmatrix} 0\,1\,1 \\ 1\,1\,1 \\ 1\,0\,1 \end{pmatrix},\ N_7 = \begin{pmatrix} 1\ \ 1\,1 \\ -1\,0\,1 \\ 1\ \ 0\,0 \end{pmatrix},\ N_8 = \begin{pmatrix} 1\ \ 0\ \ 1 \\ -1\ \ 0\ \ 0 \\ 0\ \ -1\,0 \end{pmatrix}. \tag{2}$$

Lemma 2. *The matrices above define, respectively for $k = 2$ and $k = 3$, $(1, 2, n)$-BBSS schemes with $n = 2^k$, secrets in $\mathbb{G}^k$, and each share in $\mathbb{G}^k$.*

The lemma can be verified directly by computing the determinant of every $N_i - N_j$. Next, we use these as a basis for improving on Theorem 4. We need the following generalization of the construction in that theorem, where we replace the entries i_j by square blocks A_j: i.e., given a vector $A = (A_0, \ldots, A_{\ell'-1})$ of ℓ' square matrices where each $A_i \in \mathbb{Z}^{s\times s}$, consider the matrix

$$\tilde{N}_A = \begin{pmatrix} A_0 & & & \mathbf{0} \\ \vdots & A_0 & & \\ A_{\ell'-1} & \vdots & \ddots & \\ & A_{\ell'-1} & & A_0 \\ & & \ddots & \vdots \\ \mathbf{0} & & & A_{\ell'-1} \end{pmatrix} \in \mathbb{Z}^{s\cdot(\ell'+k'-1)\times s\cdot k'}$$

Lemma 3. *Let $\mathcal{N} = \{N_i\}$ be a collection of matrices in $\mathbb{Z}^{s\times s}$ such that for each $i \neq j$, $N_i - N_j$ has an inverse in $\mathbb{Z}^{s\times s}$. Then for every $k' \leq \ell'$ and $A, B \in \mathcal{N}^{\ell'}$ with $A \neq B$, the matrix $\tilde{N}_A - \tilde{N}_B$ has a left pseudoinverse. Therefore, in these conditions there exists a $(1, 2, |\mathcal{N}|^{\ell'})$-BBSSS with secrets in $\mathbb{G}^{k's}$ and each share in $\mathbb{G}^{(\ell'+k'-1)s}$.*

Proof. This is proved similarly to Lemma 1, by considering the first index i in $[0, \ell' - 1]$ where $A_i \neq B_i$ differ, and noticing that the $(sk' \times sk')$ square submatrix of $\tilde{N}_A - \tilde{N}_B$ containing row-blocks i to $i + k' - 1$ is "block-lower triangular" and hence must have determinant $\det(A_i - B_i)^{k'} = (\pm 1)^{k'} = \pm 1$. This proves $\tilde{N}_A - \tilde{N}_B$ has a left pseudoinverse. The last part is a consequence of Theorem 3. □

Lemma 2 give us, for $s = 2$ and 3, families $\mathcal{N}_s$ of matrices in $\mathbb{Z}^{s\times s}$ with $|\mathcal{N}_s| = 2^s$. Plugging each of this constructions into Lemma 3 we have:

Theorem 5. *For $s = 2$ and 3, and any $k', \ell' > 0$, there exists a $(1, 2, 2^{s\ell'})$-BBSSS with secrets in $\mathbb{G}^k$ and each share in $\mathbb{G}^{h_*}$ where $k = s \cdot k'$ and $h_* = s \cdot (\ell' + k' - 1)$. This implies that for any n there exists:*

- *If $k \equiv 0 \mod 2$, a $(1, 2, n) - BBSSS$ with $h_* = 2\lceil \log n/2 \rceil + k - 2$; in particular, if $n = 4^m$ for $m \in \mathbb{N}$, then $h_* = \log n + k - 2$.*
- *If $k \equiv 0 \mod 3$, a $(1, 2, n) - BBSSS$ with $h_* = 3\lceil \log n/3 \rceil + k - 3$; in particular, if $n = 8^m$ for $m \in \mathbb{N}$, then $h_* = \log n + k - 3$.*

Remark 4. Note that the secret sharing schemes from Theorem 5 are indeed of the form of Theorem 3, i.e. the i-th share is computed from the secret $\boldsymbol{s}$ as $\tilde{N}_i \boldsymbol{s}^\top + \boldsymbol{r}$ for some matrix $\tilde{N}_i \in \mathbb{Z}_{h_* \times k}$ and some randomness $\boldsymbol{r}$ common to all shares. Moreover, all entries in $\tilde{N}_i$ are from $\{-1, 0, 1\}$, since they are constructed with the matrices in equations 4.3 as blocks. Each row of every $\tilde{N}_i$ has at most $\log n$ entries which are non-zero. These facts will be important in the application to Σ-protocols for class groups in Section 5.

4.4 Implications for Σ-protocols

Let $\mathbb{G}_1$ be any abelian group. Applying Theorem 1, a packed $(1, 2, n)$-BBSSS with secret in $\mathbb{G}_1^k$ and each share in $\mathbb{G}_1^{h_*}$ induces a $\Sigma-$protocol for *any* relation of the form $R = \{(w_i, x_i) | F(w_i) = x_i, i \in [k]\}$, where $F : \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ is a group homomorphism. The resulting Σ-protocol has $2-$special soundness, a knowledge error of $\frac{1}{n}$, and communicates h_* elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, and $\log n$ bits to communicate the challenge. In particular, if we consider the packed $(1, 2, n)-$black box secret sharing scheme in Theorem 5, for $n = 2^\lambda$ and $k \equiv 0 \bmod 3$ we get the following $\Sigma-$protocol:

Theorem 6. *Let $\mathfrak{R}$ be a ring, $\mathfrak{M}_1$,$\mathfrak{M}_2$ be $\mathfrak{R}-$modules and let $F : \mathfrak{M}_1 \longrightarrow \mathfrak{M}_2$ be a $\mathfrak{R}-$module homomorphism. Then there exists a $\Sigma-$protocol for the relation $R = \{(\boldsymbol{w}, \boldsymbol{x}) \in \mathfrak{M}_1^k \times \mathfrak{M}_2^k : F(w_i) = x_i, \forall i \in [k]\}$, with $2-$special soundness, error soundness $2^{-\lambda}$ and communication complexity $k + \lambda - 3$ elements of $\mathfrak{M}_1$ and $\mathfrak{M}_2$ (and λ bits for the challenge).*

The proof is almost equivalent to the one provided in Theorem 1. Note that in case $k \neq 0 \bmod 3$ we can include additional "superfluous" secrets so that we get an appropriate k. In any case the resulting communication complexity would be no more than $k + \lambda - 1$ elements of $\mathfrak{M}_1$ and $\mathfrak{M}_2$ (and λ bits for the challenge).

5 Proofs of knowledge for statements on class groups

The results presented in Theorem 6 provide assurance that an efficient $\Sigma-$protocol can be constructed to prove knowledge of preimages of group homomorphisms. This capability extends to groups of unknown order because of the usage of a black box secret sharing scheme and in particular class groups, with some slight tweaks due to the fact that the space of witnesses will be an infinite $\mathbb{Z}$-module in this case, as we will see in this section.

Background We consider the framework proposed by Castagnos and Laguillaumie. Given a large integer ℓ, the framework defines a finite commutative group $\hat{G}$ and a cyclic subgroup $G \subset \hat{G}$ both of unknown order. G is in turn a direct product $G \cong F \times G^\ell$, where F is of order ℓ, while the order of G^ℓ is computationally hard to determine. Moreover, F is endowed with an algorithm to compute discrete logarithms easily. While in earlier works [11] ℓ was taken to be prime, subsequent works [21,13,6] have considered other cases, such as ℓ being a power of a prime or a product of primes. In particular, ℓ can be of the form 2^u [13] which is useful for secure computation.

There are several variants of the Castagnos-Laguiallumie encryption, but all consist on essentially applying an El-Gamal-like encryption principle where the encryption of $m \in \mathbb{Z}_\ell$ is of the form $(c_1, c_2) = (g^r, \mathsf{pk}^r f^m)$ with f being a generator of F and the public key pk being of the form $\mathsf{pk} = g^{\mathsf{sk}}$ for a secret key sk. In perhaps the most used form of the scheme, CL-HSM [12], g is a generator of G^ℓ. The owner of sk can obtain m since it can first retrieve $f^m = c_2 c_1^{-\mathsf{sk}}$ (as in El Gamal) and then solve the discrete logarithm in F.

Proofs in the CL framework Proofs of knowledge for statements involving discrete logarithms are not too easy to construct in class groups due to the fact that the order of the group is unknown. Let us consider first the case of proving knowledge of $w \in \mathbb{Z}$ such that $g^w = x$, for some $g, x \in G$, which can be used for proving knowledge of secret keys, proving correct decryption or VSS and distributed key generation [7]. To

achieve zero-knowledge the proof is defined with respect to some $[-S, S]$ interval in which an honest witness is supposed to be (typically in protocols it does not help the malicious prover to have a witness in a larger interval instead).

In these conditions, [9] defines a proof of knowledge, similar to Schnorr, where the challenge is binary and achieves soundness error $1/2$, reduced to $1/2^\lambda$ by repetition. In short the proof, parametrized by $A, \lambda \in \mathbb{N}$ is as follows.

Repeat λ times:

- Prover sends $a = g^u$ where $u \leftarrow\!\!\$\ [0, A]$;
- Verifier sends $b \leftarrow\!\!\$\ \{0, 1\}$;
- Prover replies with $z = u + wb$;
- Verifier checks $z \in [-S, S + A]$ and $g^z = a \cdot x^b$.

Zero-knowledge is achieved as long as $\ell S/A$ is negligible (and ℓ is polynomial).

To increase efficiency, [10] avoids the repetition above by replacing $\{0, 1\}$ by a larger interval $[0, C]$ and increasing A. This achieves $1/C$ error-soundness and zero-knowledge as long as CS/A is negligible. *However*, one needs to rely on two hardness assumptions, called low order and strong root assumption, and perhaps more crucially for applications, the latter in addition requires g to be uniformly random, which may prevent its use in protocols where the adversary chooses g (see [7] for such a situation and more information). Finally, based on a different assumption, called rough order assumption, [7] defines a sound argument that allows to prove the existence of x and allow for adversarially chosen g, but this is *not* a proof of knowledge, achieving only standard soundness.

We also consider the scenario in which the witness has some coordinates in $\mathbb{Z}$ and others in $\mathbb{Z}_\ell$, which occurs for example in proving knowledge of plaintext and randomness (r, m) for a given CL-HSM ciphertext $(c_1, c_2) = (g_\ell^r, \mathsf{pk}^r f^m)$. The proof from [9] can still be used for this relation. The proof of [10] can be used if ℓ is a large enough prime (or has large enough prime factors) but cannot be used (at least as a proof of knowledge) e.g. in the case $\ell = 2^u$, because in this case the difference of two challenges may not be invertible as soon as $C \geq 2$. For the case ℓ prime, [7] defines a proof of plaintext knowledge, where the extractor can extract m but not r; however, again this will not work if $\ell = 2^u$.

Proofs of knowledge from our framework For simplicity, we directly use the BBSSS derived from Theorem 5, defined by $N_1, \ldots, N_n \in \mathbb{Z}^{h_* \times k}$ so that each difference $N_i - N_j$ has a left pseudoinverse. Recall that given a **finite** abelian group $\mathbb{G}$, the shares for $\boldsymbol{w} \in \mathbb{G}^k$ are $\boldsymbol{\sigma}_i = N_i \boldsymbol{w} + \boldsymbol{r}$ for uniformly random $\boldsymbol{r} \leftarrow\!\!\$\ G^{h_*}$.

In the case of the **infinite** group $\mathbb{G} = \mathbb{Z}$, we cannot sample $\boldsymbol{r}$ uniformly in $\mathbb{Z}^{h_*}$ any more. Instead we achieve statistical 1-privacy by sampling $\boldsymbol{r}$ uniformly from an interval $[0, A]^{h_*}$, for some $A \in \mathbb{Z}_{>0}$ large enough so that $\boldsymbol{w} \in [-S, S]^k$ is statistically hidden by each individual share $N_i \boldsymbol{w} + \boldsymbol{r}$. Here it is helpful, that the N_i from Theorem 5 have coefficients in $\{-1, 0, 1\}$, as this prevents A from growing too much. On the other hand, 2-reconstruction holds without change.

Proof of discrete logarithm We first consider the relation $R_{DLCG,k} := \{(\boldsymbol{w}, \boldsymbol{x}) \in \mathbb{Z}^k \times G^k \mid g^{w_i} = x_i\ \forall i = 1, \ldots k\}$ where $G = \langle g \rangle$ is the cyclic group of unknown order mentioned above. Let $\psi : \mathbb{Z} \to G$ be given by $\psi(w) = g^w$ and as usual let $\boldsymbol{\psi}(\boldsymbol{w}) := (\psi(w_1), \ldots, \psi(w_k))$. We assume the honest witness $\boldsymbol{w}$ will be in an interval $[-S, S]^k$. Then, consider the protocol in Figure 4 for the relation $R_{DLCG,k}$, parametrized by the integer A. $\boldsymbol{x}^{N_i}$ is defined as the vector in G^{h_*} whose j-th coordinate is $\prod_{l=1}^k x_l^{(N_i)_{j,l}}$ where $(N_i)_{j,l}$ is the entry in row j, column l of N_i, and $*$ represents coordinate-wise product.

Theorem 7. *Assume $N_1, \ldots, N_n \in \mathbb{Z}^{h_* \times k}$ are such that $N_i - N_j$ has a left pseudoinverse for all $i \neq j$. Moreover let $D \in \mathbb{Z}^+$ such that it is an upper bound for the sum of absolute values of the entries of every row in every matrix (namely, $\forall i \in [n], j \in [h_*]$, we have $\sum_{l=1}^k |(N_i)_{j,l}| \leq D$).*

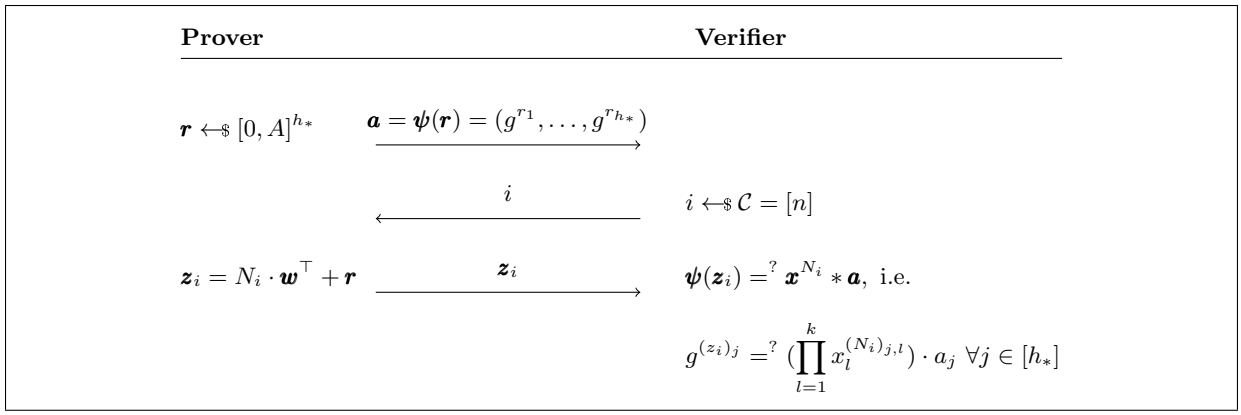

Fig. 4. Σ-protocol over Class Groups

Then the protocol in Figure 4 is a Σ-protocol with 2-special soundness and, as long as $\epsilon = SDh_*/A$ is negligible and h_* is polynomial, it is statistical honest-verifiable zero knowledge. For the specific case of N_i as in Theorem 5, the result above holds with $D = \min\{k, \log n\}$.

Proof. **Completeness:** We have that, for $j \in [h_*]$, $(z_i)_j = (\sum_{l=1}^{k}(N_i)_{j,l} \cdot w_l) + r_j$. Therefore if $x_l = g^{w_l}$ for $l \in [k]$ and $a_j = g^{r_j}$ for $j \in [h_*]$, then clearly

$$g^{(z_i)_j} = (\prod_{l=1}^{k} x_l^{(N_i)_{j,l}}) \cdot a_j \ \forall j \in [h_*]$$

and the protocol accepts.

2-Special soundness: Suppose two conversations $(\boldsymbol{a}, i, \boldsymbol{z}_i)$ and $(\boldsymbol{a}, j, \boldsymbol{z}_j)$ accept where $i \neq j$. This implies $\boldsymbol{\psi}(\boldsymbol{z}_i - \boldsymbol{z}_j) = g^{\boldsymbol{z}_i - \boldsymbol{z}_j} = \boldsymbol{x}^{N_i - N_j}$.

Let R_{ij} be the integer left pseudoinverse of $N_i - N_j$. Then we have that $\boldsymbol{w'} = R_{ij}(\boldsymbol{z}_i - \boldsymbol{z}_j)$ is a witness. Indeed

$$\psi(w'_t) = \psi(\sum_{u=1}^{h_*}(R_{ij})_{t,u}(z_i - z_j)_u) = \sum_{u=1}^{h_*}(R_{ij})_{t,u} \cdot \psi(z_i - z_j)_u = \prod_{u=1}^{h_*}(\boldsymbol{x}^{N_i - N_j})_u^{(R_{ij})_{t,u}} =$$

$$\prod_{u=1}^{h_*}\prod_{l=1}^{k} x_l^{(N_i - N_j)_{u,l}(R_{ij})_{t,u}} = \prod_{l=1}^{k} x_\ell^{\sum_{u=1}^{h_*}(R_{ij})_{t,u}(N_i - N_j)_{u,\ell}} = x_t.$$

Statistical Zero Knowledge: Given $\boldsymbol{x}$ and a challenge $i \in [n]$, the simulator chooses $\boldsymbol{z}_i$ uniformly at random in the set $[-SD, SD + A]^{h_*}$ and then selects the unique $\boldsymbol{a}$ that makes the proof accept, namely $\boldsymbol{a} = \boldsymbol{\psi}(\boldsymbol{z}_i) * \boldsymbol{x}^{-N_i}$.

In the real protocol, each component of $\boldsymbol{z}_i = N_i\boldsymbol{w} + \boldsymbol{r}$ is uniform in some subinterval of $[-SD, SD + A]$ of length A, because $N_i\boldsymbol{w}$ is some fixed vector in $[-SD, SD]^{h_*}$ by assumptions on $\boldsymbol{w}$ and D. As long as $\epsilon = SDh_*/A$ is negligible and h_* is polynomial this is statistically close to the uniform distribution in $[-SD, SD + A]^{h_*}$, with statistical distance given by $1 - (\frac{1}{1+2\epsilon})^{h_*}$. □

Remark 5. The proof from [9], described above, can be cast as the instance of that in Figure 4, with $k = 1$, $n = 2^\ell$ and with N_i being the matrices of dimensions $\ell \times 1$ given by vectors in $\{0, 1\}^\ell$. Note in that case $h_* = \log n = \ell$, $D = 1$.

Corollary 2. *Let $k = 0 \mod 3$. When using the matrices N_i given by the BBSSS from Theorem 5 the protocol in Figure 4 is a Σ-Protocol for $R_{DLCG,k}$ with the following properties where $D = \min\{k, \log n\}$*

- *Knowledge error at most $1/n$;*
- *Statistical honest verifier zero-knowledge with assuming the witness is in $[-S, S]^k$ and $A > 2^\kappa(k + \log n - 3)D \cdot S$ for statistical security parameter κ;*
- *Communication complexity: $k + \log n - 3$ elements of G and $k + \log n - 3$ integers in $[-SD, SD + A]$, as well as $\log n$ bits for the challenge.*

Its Fiat-Shamir non-interactive version therefore communicates $k + \log n - 3$ integers in $[-SD, SD + A]$ (and $\log n$ bits).

Proofs of plaintext and randomness knowledge (and other 'hybrid' statements) The same template applies to situations where part of the witness will be in $\mathbb{Z}$ (because we do not know the order of the cyclic subgroup it acts on) and part in $\mathbb{Z}_q$. The most clear example is that of proofs of knowledge of plaintext, which we will use to illustrate this case. The encryption function of CL-HSM is $\psi : \mathbb{Z}_\ell \times \mathbb{Z} \to G^\ell \times G$ given by $\psi(m, r) = (g_\ell^r, \mathsf{pk}^r \cdot f^m)$ for generators g_ℓ, f respectively of G^ℓ and F, and public key pk in G^ℓ. Let then

$$R_{CL,k} := \{(\boldsymbol{m}, \boldsymbol{r}); (\boldsymbol{c}, \boldsymbol{d}) \in (\mathbb{Z} \times \mathbb{Z}_\ell)^k \times (G^\ell \times G)^k \mid \psi(m_i, r_i) = (c_i, d_i)\ \forall i = 1, \dots k\}.$$

The Σ-protocol for $R_{CL,k}$ is then very similar to that in Figure 4, with the difference that now the space of witnesses is $\mathbb{Z}_\ell \times \mathbb{Z}$, rather than $\mathbb{Z}$. But this is easy to deal with, because our BBSSS can of course be applied to the former space. We present the proof in Figure 5.

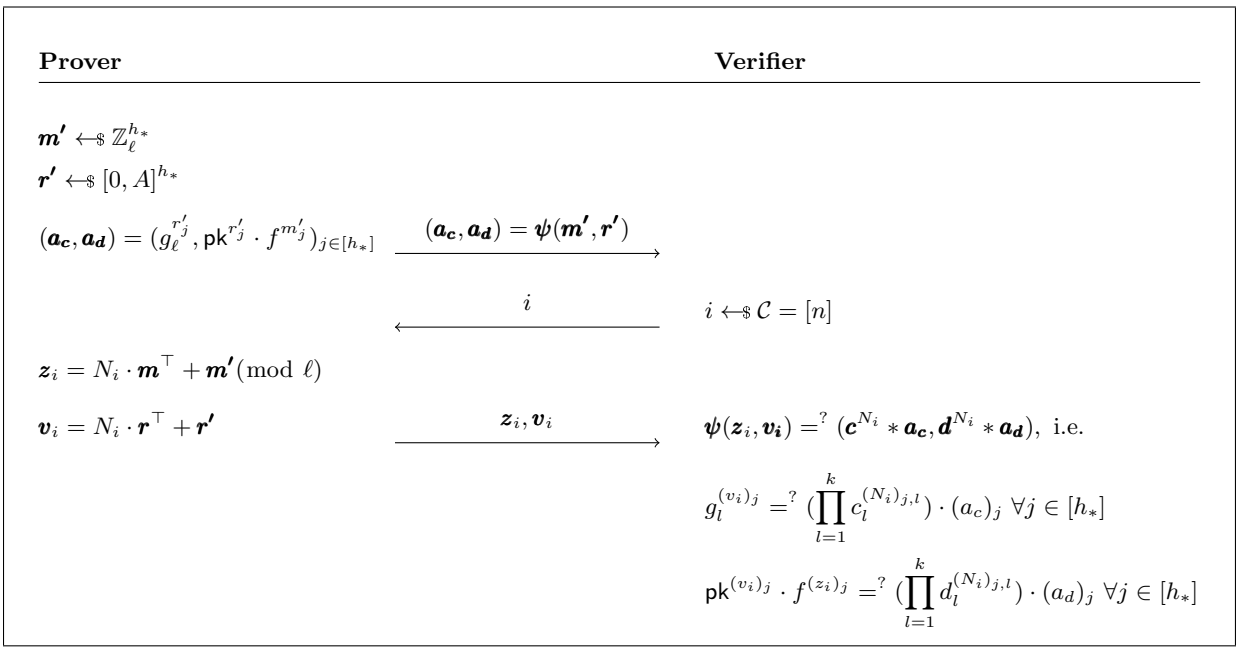

Fig. 5. Σ-protocol over Class Groups for R_{CL}

The analysis of the security properties of this protocol works exactly as in Theorem 7. For the communication complexity, we observe that, in addition to the h_* integers in $[-DS, A + DS]$, the prover sends h_* elements of $\mathbb{Z}_\ell$ but the size of these are independent of D. We also emphasize that this proof has witness extraction regardless of whether ℓ is prime or not.

Remark 6 (Comparisons). We compare the communication cost of the non-interactive version of our proof for the relation $R_{DLCG,k}$ with the protocols we would get by adapting [9,10,7] to this case. For this comparison, we take a statistical security parameter λ, and we aim at having soundness error $2^{-\lambda}$. Moreover, we select $A > 2^{\lambda} h_* S$ in our case, and similarly $A > 2^{\lambda} \ell S$ in [9] and $A > 2^{\lambda} CS$ in [10,7]. We ignore the cost of the challenge, since it is λ bits in all cases, and much smaller than the rest of the proof.

Proof	Communication (bits)	Knowledge	Assumptions
[9]	$\lambda k(\log S + \lambda + \log \lambda)$	Yes	None
[10]	$k(\log S + 2\lambda)$	Yes	Low order, Strong Root, uniform random h
[7]	$k(\log S + 2\lambda)$	No	Rough Order
Figure 4	$(k+\lambda-3)(\log S + \lambda + \log(k+\lambda) + \log \min(\lambda, k))$	Yes	None

For k larger enough than λ (but still much smaller than 2^{λ}) the protocol from Figure 4 saves a multiplicative factor around λ with respect to the proof in [9]. On the other hand, as the dominant factor is $k \log S$ the complexity is quite comparable to the protocols in [10]. However, we emphasize that the advantage we get with our proofs is that we do not require any hardness assumption for the proof, that the basis g does not need to be uniformly random and can be a value that was chosen by the adversary in a protocol and that ours is a proof of knowledge (as opposed to [7,10]).

The case of R_{CL} (Figure 5) is similar, but now in our case the communication includes in addition h_* elements of $\mathbb{Z}_\ell$, while the other protocols we compare with would add k elements instead. We remark again that in that case, for ℓ a large prime, the proof in [10] has an extractor for the plaintext. However, this is not the case if ℓ has a divisor smaller than the challenge bound C, e.g. in $\ell = 2^u$.[3] In contrast, in our case we can extract the witness regardless of the modulus.

Regarding computational complexity, the number of operations of our proof is comparable to the one in [10,7]. When counting the left side of the verification we obtain the same relation as with the last message communication shown in the table, since there is one exponentiation per group element sent. The right side of the verification has a computational complexity of $k \cdot \log n$ group operations, equivalent to [10,7]. Indeed the number of group operations equals the maximum number of non-zero entries in the matrix N_i which is $7/9 \cdot k \cdot \log n$. The exponentiations required in [10,7] lead to $k \cdot \log n$ operations when counting an exponentiation (with exponent in $[0, A]$) as $\log A$ operations.

Remark 7. While we have stated the results above for proofs of discrete logarithm knowledge, they can be easily extended to other statements on the class group, e.g. discrete logarithm equality or more generally linear relations as defined in [7] (namely statements of the form $\bigwedge_{i=1}^{n} Y_i = X_{i,1}^{w_1} \cdot \cdots \cdot X_{i,s}^{w_s}$).

6 Extension to ZK-ready functions

So far we have considered $\Sigma-$protocols for relations of the form $(w, x = F(w))$ where F is a module homomorphism, e.g. a group homomorphism. We now extend our result to the case of ZK-ready functions, defined in [16]. These are maps $f : U \times S \longrightarrow X$ where $(U, +)$, $(S, \cdot)$, $(X, \cdot)$ are groups and are homomorphic "up to a correction factor in their second argument"; namely, we have $f(u, s) \cdot f(u', s') = f(u + u', s \cdot s' \cdot \delta(u, u'))$ for some function δ. This notion is relevant because it e.g. captures encryption functions from several cryptosystems with homomorphic properties (Joye-Libert, Paillier), where U and S are the plaintext and randomness spaces.

[3] We do point out that it may be worthwhile to investigate whether the techniques from Section 6.1 using Galois rings can be used to construct proofs of plaintext knowledge in that case.

Definition 8. *[16] Let $(U,+),(S,\cdot),(X,\cdot)$ be abelian groups, $\mathfrak{R}$ a commutative ring with 1 and $f : U \times S \longrightarrow X$ a function. The function f is said to be ZK-ready with respect to $\mathfrak{R}$ if:*

- *There exist $g : U \longrightarrow X$ and a group homomorphism $h : S \longrightarrow X$ such that $g(0) = 1$, $f(u,s) = g(u)h(s)$ and $(\pi \circ g)(u+u') = (\pi \circ g)(u) \cdot (\pi \circ g)(u')$ for all $u, u' \in U$, $s \in S$, where $\pi : X \longrightarrow X/Im(h)$ is the canonical projection.*
- *Every $a \in \mathfrak{R}$ acts as an endomorphism of X, i.e. $(xy)^a = x^a y^a$, $x^0 = 1, x^1 = x$ for all $x, y \in X$.*
- *U and $Im(\pi)$ are $\mathfrak{R}-$modules, $Im(\pi \circ g)$ is a submodule, $\pi \circ g$ is a $\mathfrak{R}-$module homomorphism and $\pi(x^a) = \pi(x)^a$ for all $a \in \mathfrak{R}$, $x \in X$.*

The next lemma from [16] ensures that this functions satisfy the almost homomorphic property we referred to before.

Lemma 4 ([16]). *Let f be ZK-ready with respect to $\mathfrak{R}$. Then there exist $\delta : U \times U \longrightarrow S$ and $\gamma : \mathfrak{R} \times U \times S \longrightarrow S$ such that for all $a \in \mathfrak{R}$, $u, u' \in U$ and $s, s' \in S$, $f(u,s)f(u',s') = f(u+u', ss'\delta(u,u'))$ and $f(u,s)^a = f(au, \gamma(a,u,s))$.*

From now on we will consider only the case of functions ZK-ready with respect to $\mathbb{Z}$ or $\mathbb{Z}_\ell$. In this scenario, the function $\gamma(a,u,s)$ the second equation can be reduced to a simpler form using the lemma below.

Lemma 5. *If $\mathfrak{R} = \mathbb{Z}$ or $\mathfrak{R} = \mathbb{Z}_\ell$ and f is ZK-ready with respect to $\mathfrak{R}$, then there exists an efficiently computable $\xi : \mathfrak{R} \times U \to S$ such that for $a \in \mathfrak{R}, u \in U, s \in S$, $f(u,s)^a = f(au, s^a \cdot \xi(a,u))$ (where if $\mathfrak{R} = \mathbb{Z}_\ell$, s^a is computed by embedding a in $[0, \ell-1] \subseteq \mathbb{Z}$).*

Proof. For $\mathfrak{R} = \mathbb{Z}$, the proof follows the same as for Theorem 1 in [16], where $\xi(a,u) = \prod_{i=2}^{|a-1|} \delta(i \cdot u, u)^{sign(a)}$. The same holds when $\mathfrak{R} = \mathbb{Z}_\ell$, but then we interpret a as an integer in $[0, \ell-1]$. □

When a matrix $M \in \mathfrak{R}^{n \times m}$ acts over a vector $\boldsymbol{z} = (z_1, \ldots, z_m)$ in either S^m or X^m, we assume it acts as a matrix of integers, even when $\mathfrak{R} = \mathbb{Z}_\ell$ (we just embed $\mathbb{Z}_\ell$ in $[0, \ell-1] \subseteq \mathbb{Z}$ as above). Since the notation of these groups is multiplicative, we will write $\boldsymbol{z}^M := (\prod_{i=1}^m z_i^{M_{1i}}, \ldots, \prod_{i=1}^m z_i^{M_{ni}})^\top \in S^n$ or (X^n), with M_{ij} being the (i,j)-th entry of M). If $N \in \mathfrak{R}^{p \times n}$ is another matrix, we have that $(\boldsymbol{z}^M)^N = \boldsymbol{z}^{N \cdot M}$. The next lemma is a generalization of Lemma 5.

Lemma 6. *If $\mathfrak{R} = \mathbb{Z}$ or $\mathfrak{R} = \mathbb{Z}_\ell$, and f is ZK-ready with respect to $\mathfrak{R}$, then given $m, n > 0$, there exists efficiently computable $\Xi : \mathfrak{R}^{n \times m} \times U^m \to S^n$ such that for all $M \in \mathfrak{R}^{n \times m}, \boldsymbol{u} \in U^m, \boldsymbol{s} \in S^m$, $\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{s})^M = \boldsymbol{f}(M \cdot \boldsymbol{u}, \boldsymbol{s}^M * \Xi(M, \boldsymbol{u}))$ where $*$ denotes coordinate-wise product in S^n.*

We extend our protocol of Theorem 1 so that it can be used to prove knowledge of preimages of elements via a ZK-ready function. However, in this case we need additional assumptions. In the case $\mathfrak{R} = \mathbb{Z}$, we need that the induced SSS on U and S by the MSP M over $\mathbb{Z}$ satisfy "share-uniformity" for the shares corresponding to E, for every challenge set E. Moreover we need "randomness-uniqueness" on the one induced on X, meaning that for a given secret x and given shares for set E, there is at most one choice of randomness that gives those shares for that secret. When $\mathfrak{R} = \mathbb{Z}_\ell$ we can no longer assume M generates a linear secret sharing on S and X, since only U is guaranteed to be a $\mathfrak{R}$-module. However we still state the assumptions above in algebraic terms, and we will see these can be achieved for some constructions of MSPs. We also need an additional assumption that given $y \in X$, it is easy to find $t \in S$ with $f(0,t) = y^\ell$. This is the case when f is the encryption function for Joye-Libert ($\ell = 2^l$).

Theorem 8. *Let $\mathfrak{R}$ be $\mathbb{Z}$ or $\mathbb{Z}_\ell$, $(U,+),(S,\cdot),(X,\cdot)$ finite abelian groups. Let $f : U \times S \to X$ be ZK-ready with respect to $\mathfrak{R}$. Let $\mathcal{M} = (\mathfrak{R}, M, \Psi, k)$ be a $k-$MSP over $\mathfrak{R}$ computing (Δ, Γ), where $M \in \mathfrak{R}^{h \times (k+e)}$. Let*

$\mathcal{C}$ be a compatible challenge set with respect to (Δ, Γ) and $\nu = \nu(\mathcal{C}, \Gamma)$ its extraction number. Assume it is possible to sample efficiently uniform from $\mathcal{C}$, U and S, that for every $E \in \mathcal{C}$, M_E can be computed efficiently, and that for any pairwise distinct $E_1, \ldots, E_\nu \in \mathcal{C}$ it is efficient to compute the reconstruction vectors $\boldsymbol{\rho}_{\mathcal{E}}^{(i)}$, where $\mathcal{E} = \bigcup_{i=1}^{\nu} E_i$, promised by Definition 6. Assume in addition:

- *For any $E \in \mathcal{C}$ the distributions of $M_E(\mathbf{0}, \boldsymbol{u'})^\top$ and $(\mathbf{1}, \boldsymbol{s'})^{M_E}$ when $\boldsymbol{u'} \leftarrow\!\$\ U^e$, $\boldsymbol{s'} \leftarrow\!\$\ S^e$, are uniformly random in U^{h_E}, S^{h_E} respectively.*
- *For any $E \in \mathcal{C}$, any $\boldsymbol{y} \in X^{h_E}$, and any $\boldsymbol{x} \in X^k$ there is at most one $\boldsymbol{a} \in X^e$ such that $(\boldsymbol{x}, \boldsymbol{a})^{M_E} = \boldsymbol{y}$.*
- *For the case $\mathfrak{R} = \mathbb{Z}_\ell$ we also assume that f is such that given $y \in X$ it is easy to find $t \in S$ with $f(0, t) = y^\ell$.*

Let Ξ as in Lemma 6. Then Figure 6 provides a Σ-protocol with ν-special soundness for the relation $R = \{(\boldsymbol{u}, \boldsymbol{s}; \boldsymbol{x}) \in (U^k \times S^k) \times X^k : f(u_i, s_i) = x_i \ \forall i \in [k]\}$.

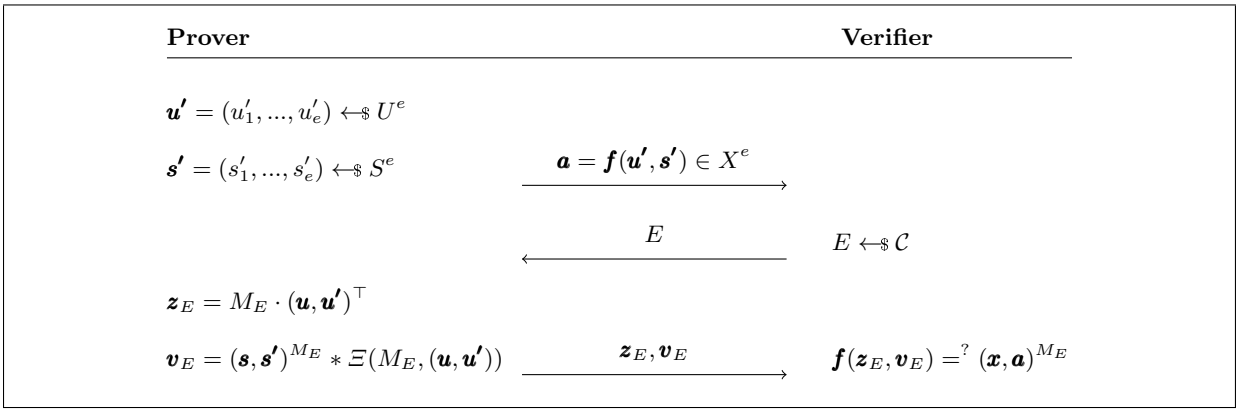

Fig. 6. Σ-Protocol for preimages of ZK-ready functions

Proof. **Correctness**: The correctness of the protocol follows from Lemma 6, since $\boldsymbol{f}(\boldsymbol{z}_E, \boldsymbol{v}_E) = \boldsymbol{f}((\boldsymbol{u}, \boldsymbol{u'}), (\boldsymbol{s}, \boldsymbol{s'}))^{M_E} = (\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{s}), \boldsymbol{f}(\boldsymbol{u'}, \boldsymbol{s'}))^{M_E} = (\boldsymbol{x}, \boldsymbol{a})^{M_E}$.

$\boldsymbol{\nu}$−Special soundness: Given ν accepting conversations $(\boldsymbol{a}, E_1, \boldsymbol{z}_{E_1}, \boldsymbol{v}_{E_1}), \ldots, (\boldsymbol{a}, E_\nu, \boldsymbol{z}_{E_\nu}, \boldsymbol{v}_{E_\nu})$ the extractor proceeds as follows. Let $\mathcal{E} = \bigcup_{i=1}^{\nu} E_j$. Since $\mathcal{E}$ is a reconstructing set for the MSP, there exist reconstruction vectors $\boldsymbol{\rho}_{\mathcal{E}}^j \in \mathfrak{R}^{h_E}$ for $j \in [k]$, such that $\boldsymbol{\rho}_{\mathcal{E}}^j \cdot M_{\mathcal{E}} = (0, \ldots, 0, 1, 0, \ldots, 0)$ over $\mathfrak{R}$, where the 1 is in the j-th position. Let $\boldsymbol{\rho}_{\mathcal{E}}$ be the matrix with rows $\boldsymbol{\rho}_{\mathcal{E}}^j$. Then $\boldsymbol{\rho}_{\mathcal{E}} \cdot M_{\mathcal{E}} = (I_{k\times k} | 0_{k\times e})$.

Now, consider first the case $\mathfrak{R} = \mathbb{Z}$. Then the above matrix equation holds over the integers and consequently $((\boldsymbol{x}, \boldsymbol{a})^{M_\mathcal{E}})^{\boldsymbol{\rho}_\mathcal{E}} = (\boldsymbol{x}, \boldsymbol{a})^{\boldsymbol{\rho}_\mathcal{E} \cdot M_\mathcal{E}} = \boldsymbol{x}$.

Let $\boldsymbol{z}_\mathcal{E}$ and $\boldsymbol{v}_\mathcal{E}$ be vectors whose coordinates are indexed by the elements in $\mathcal{E}$ and are defined as follows: for every $c \in \mathcal{E}$, c belongs to some E_i. If c belongs to more than one E_i, choose one of them. Then the extractor takes the c-th coordinates of $\boldsymbol{z}_\mathcal{E}$ and $\boldsymbol{s}_\mathcal{E}$ to be the corresponding $\mathcal{E}$-th coordinates of $\boldsymbol{z}_{E_i}$ and $\boldsymbol{s}_{E_i}$ respectively. The extractor outputs $\boldsymbol{u} = \boldsymbol{\rho}_\mathcal{E} \cdot \boldsymbol{z}_\mathcal{E}^\top$ and $\boldsymbol{s} = \boldsymbol{v}^{\boldsymbol{\rho}_\mathcal{E}} * \Xi(\boldsymbol{\rho}_\mathcal{E}, \boldsymbol{v})$, which satisfy

$$\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{s}) = \boldsymbol{f}(\boldsymbol{\rho}_\mathcal{E} \cdot \boldsymbol{z}_\mathcal{E}^\top, \boldsymbol{v}^{\boldsymbol{\rho}_\mathcal{E}} * \Xi(\boldsymbol{\rho}_\mathcal{E}, \boldsymbol{v})) = \boldsymbol{f}(\boldsymbol{z}_\mathcal{E}, \boldsymbol{v}_\mathcal{E})^{\boldsymbol{\rho}_\mathcal{E}} = ((\boldsymbol{x}, \boldsymbol{a})^{M_\mathcal{E}})^{\boldsymbol{\rho}_\mathcal{E}} = \boldsymbol{x}.$$

If $\mathfrak{R} = \mathbb{Z}_\ell$ we need to modify the proof as follows. Note that now it does not hold necessarily that $((\boldsymbol{x}, \boldsymbol{a})^{M_\mathcal{E}})^{\boldsymbol{\rho}_\mathcal{E}} = \boldsymbol{x}$ because $\boldsymbol{\rho}_\mathcal{E} \cdot M_\mathcal{E} = (I_{k\times k} | 0_{k\times e})$ holds over $\mathbb{Z}_\ell$ and not over the integers, and X is not necessarily a $\mathbb{Z}_\ell$-module. Over the integers we have in fact $\boldsymbol{\rho}_\mathcal{E} \cdot M_\mathcal{E} = (I_{k\times k} | 0_{k\times e}) + \ell \cdot L$ for some matrix

$L \in \mathbb{Z}^{h_\mathcal{E} \times (k+e)}$. Redoing the operations before we now get $((\boldsymbol{x},\boldsymbol{a})^{M_\mathcal{E}})^{\boldsymbol{\rho}_\mathcal{E}} = \boldsymbol{x} * \boldsymbol{y}^\ell$ for some vector $\boldsymbol{y}$ that can be efficiently computed from $\boldsymbol{\rho}^j_\mathcal{E}$, $M_\mathcal{E}$, $\boldsymbol{x}$ and $\boldsymbol{a}$. Then if we set $\boldsymbol{u}, \boldsymbol{s}$ as before, we have $\boldsymbol{f}(\boldsymbol{u},\boldsymbol{s}) = \boldsymbol{x} * \boldsymbol{y}^\ell$. On the other hand by assumption, we can efficiently construct a vector $\boldsymbol{t}$ with $f(\boldsymbol{0},\boldsymbol{t}) = \boldsymbol{y}^{-\ell}$ (by applying the assumption to each y_j^{-1} where y_j are the coordinates of $\boldsymbol{y}$). Now we have $f(\boldsymbol{u}, \boldsymbol{s} * \boldsymbol{t} * \delta(\boldsymbol{u}, \boldsymbol{0})) = \boldsymbol{x}$ where $\boldsymbol{\delta}$ is the δ of Lemma 4 applied coordinatewise.

Honest-verifier zero-knowledge: As in the proof of Theorem 1, the simulator constructs the simulated third message $(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ as if it was running the real protocol with $\widehat{\boldsymbol{u}} = \boldsymbol{0}$ and, in this case (since S is written multiplicatively), $\widehat{\boldsymbol{s}} = \boldsymbol{1}$. Then the simulator creates $\widehat{\boldsymbol{a}}$ such that the verification for $\boldsymbol{x}$ passes, for which it uses the fact that E is a privacy set for the MSP.

Concretely, consider first the case $\mathfrak{R} = \mathbb{Z}$. Given $\boldsymbol{x} \in X^k$, and $E \in \mathcal{C}$ the simulator:

1. Samples $\widehat{\boldsymbol{u}}' \in U^e$, $\widehat{\boldsymbol{s}}' \in S^e$ uniformly at random, creates $\widehat{\boldsymbol{z}}_E = M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top$ and $\widehat{\boldsymbol{v}}_E = (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}'))$.
2. For $i \in [k]$ let $\boldsymbol{\lambda}^{(i)} \in \mathfrak{R}^{k+e}$ with $(\boldsymbol{\lambda}^{(i)})_i = 1$, $(\boldsymbol{\lambda}^{(i)})_j = 0$ for $j \in [k] \setminus \{i\}$, and $M_E {\boldsymbol{\lambda}^{(i)}}^\top = \boldsymbol{0}_{\widehat{E}}$ which exists by definition of k-MSP (Definition 6,(P2)). Let $\bar{\boldsymbol{\lambda}}^{(i)} = \boldsymbol{\lambda}^{(i)}_{[k+1,k+e]}$ be the vector of the last e coordinates of $\boldsymbol{\lambda}^{(i)}$.
3. Define $\widehat{\boldsymbol{a}}$ as follows: $\widehat{\boldsymbol{a}} = f(\widehat{\boldsymbol{u}}', \widehat{\boldsymbol{s}}') * \prod_{i=1}^k x_i^{\bar{\boldsymbol{\lambda}}^{(i)}}$ where $x_i^{\bar{\boldsymbol{\lambda}}^{(i)}}$ is the vector in X^e obtained by having each coordinate of $\bar{\boldsymbol{\lambda}}^{(i)}$ act on x_i and the products are componentwise.
4. Output $(\widehat{\boldsymbol{a}}, E, (\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E))$.

First note that $(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ is uniformly distributed in $(U \times S)^{h_E}$ by assumption (in the case of $\widehat{\boldsymbol{v}}_E$, the assumption says $(\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E}$ is uniform if $\widehat{\boldsymbol{s}}'$ uniformly random, but note that $\Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}'))$ is independent from $\widehat{\boldsymbol{s}}'$ so the product $\widehat{\boldsymbol{v}}_E$ of both vectors must also be uniform).

Now we show $(\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E} = \boldsymbol{f}(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ as in the real protocol.

Note that $(\boldsymbol{x}, \widehat{\boldsymbol{a}}) = (\boldsymbol{x}, \boldsymbol{f}(\widehat{\boldsymbol{u}}', \widehat{\boldsymbol{s}}') * \prod_{i=1}^k x_i^{\bar{\boldsymbol{\lambda}}^{(i)}}) = (\boldsymbol{x}, \prod_{i=1}^k x_i^{\bar{\boldsymbol{\lambda}}^{(i)}}) * (\boldsymbol{1}, \boldsymbol{f}(\widehat{\boldsymbol{u}}', \widehat{\boldsymbol{s}}'))$.

Now recalling that $\boldsymbol{\lambda}^{(i)} = (\boldsymbol{0}_{i-1}, 1, \boldsymbol{0}_{k-i}, \bar{\boldsymbol{\lambda}}^{(i)})$, we can see that $(\boldsymbol{x}, \prod_{i=1}^k x_i^{\bar{\boldsymbol{\lambda}}^{(i)}}) = \prod_{i=1}^k x_i^{\boldsymbol{\lambda}^{(i)}}$. But since $M_E \boldsymbol{\lambda}^{(i)} = \boldsymbol{0}$ over $\mathbb{Z}$, we have $(\prod_{i=1}^k x_i^{\boldsymbol{\lambda}^{(i)}})^{M_E} = \boldsymbol{1}$ and $(\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E} = (\boldsymbol{1}, \boldsymbol{f}(\widehat{\boldsymbol{u}}', \widehat{\boldsymbol{s}}'))^{M_E}$.

The ZK-readyness of f guarantees that $1 = f(0,1)$ so actually $\boldsymbol{1} = \boldsymbol{f}(\boldsymbol{0}, \boldsymbol{1})$. This means $(\boldsymbol{1}, f(\widehat{\boldsymbol{u}}', \widehat{\boldsymbol{s}}')) = f((\boldsymbol{0}, \widehat{\boldsymbol{u}}'), (\boldsymbol{1}, \widehat{\boldsymbol{s}}'))$ and therefore

$$(\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E} = \boldsymbol{f}((\boldsymbol{0}, \widehat{\boldsymbol{u}}'), (\boldsymbol{1}, \widehat{\boldsymbol{s}}'))^{M_E} = f(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top, (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}')))$$

which is exactly $\boldsymbol{f}(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$.

So we have $(\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E} = \boldsymbol{f}(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ in the simulation and $(\boldsymbol{x}, \boldsymbol{a})^{M_E} = \boldsymbol{f}(\boldsymbol{z}_E, \boldsymbol{v}_E)$ in the real protocol, for the same $\boldsymbol{x}$.

Since $(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ and $(\boldsymbol{z}_E, \boldsymbol{v}_E)$ are equally distributed and, by assumption, respectively $\widehat{\boldsymbol{a}}$ and $\boldsymbol{a}$ are unique satisfying the equations, it then must be the case that $(\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E)$ and $(\boldsymbol{a}, \boldsymbol{z}_E, \boldsymbol{v}_E)$ are equally distributed.

In the case where $\mathfrak{R} = \mathbb{Z}_\ell$, the proof above does not work directly because $M_E \boldsymbol{\lambda}^{(i)} = \boldsymbol{0}$ over $\mathbb{Z}_\ell$ and not over $\mathbb{Z}$. So similarly to the case of soundness, we have $(\prod_{i=1}^k x_i^{\boldsymbol{\lambda}^{(i)}})^{M_E} = \boldsymbol{y}^\ell$ for some vector $\boldsymbol{y}$. This means

$$(\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E} = \boldsymbol{f}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top, (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}'))) * \boldsymbol{y}^\ell.$$

So we need to correct the definition of $\widehat{\boldsymbol{v}_E}$. By the assumption we know it is easy to find $\boldsymbol{t} \in S^{h_E}$ with $f(\boldsymbol{0}, \boldsymbol{t}) = \boldsymbol{y}^\ell$. We define $\widehat{\boldsymbol{v}_E} = (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}')) * \boldsymbol{t} * \boldsymbol{\delta}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}'), \boldsymbol{0})$. Now we get

$$\boldsymbol{f}(\widehat{\boldsymbol{z}}_E, \widehat{\boldsymbol{v}}_E) = \boldsymbol{f}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top + \boldsymbol{0}, (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}')) * \boldsymbol{t} * \boldsymbol{\delta}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top, \boldsymbol{0})) =$$

$$\boldsymbol{f}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top, (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}')) * \boldsymbol{f}(\boldsymbol{0}, \boldsymbol{t}) =$$
$$\boldsymbol{f}(M_E \cdot (\boldsymbol{0}, \widehat{\boldsymbol{u}}')^\top, (\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E} * \Xi(M_E, (\boldsymbol{0}, \widehat{\boldsymbol{u}}'))) * \boldsymbol{y}^\ell = (\boldsymbol{x}, \widehat{\boldsymbol{a}})^{M_E}.$$

Finally observe that $\widehat{\boldsymbol{v}}_E$ is still uniformly random, as with respect to the case $\mathfrak{R} = \mathbb{Z}$ we multiply $(\boldsymbol{1}, \widehat{\boldsymbol{s}}')^{M_E}$ by a vector which is independent from $\widehat{\boldsymbol{s}}'$. □

Remark 8 (achieving additional assumptions). When the challenge space consists of sets $E = \{i\}$ of size 1, and the M_i are of the form $M_i = (N_i | I_{e\times e})$ (in this case $h_E = e$), the assumptions of Theorem 8 are achieved. Indeed in that case, $(\boldsymbol{s}, \boldsymbol{s'})^{M_i} = \boldsymbol{s}^{N_i} * \boldsymbol{s'}$, so if $\boldsymbol{s'}$ is uniform $(\boldsymbol{s}, \boldsymbol{s'})^{M_i}$ is uniform too. This holds even if S is not a $\mathfrak{R}$-module, because we are embedding $I_{e\times e} \in \mathbb{Z}_\ell^{e\times e}$ into $I_{e\times e} \in \mathbb{Z}^{e\times e}$. By the same token $(\boldsymbol{x}, \boldsymbol{a})^{M_i} = \boldsymbol{x}_i^N * \boldsymbol{a}$ so there is exactly one $\boldsymbol{a} = (\boldsymbol{x}^{N_i})^{-1} * \boldsymbol{y}$ satisfying $(\boldsymbol{x}, \boldsymbol{a})^{M_i} = \boldsymbol{y}$. In particular, this holds for the $(1, 2, n)-$ BBSS schemes described in Theorem 3 and in particular the ones in Theorem 5.

If in addition to the uniqueness of $\boldsymbol{a}$ we assume that $\boldsymbol{a}$ can be computed efficiently, as for example in the case of the schemes mentioned in Remark 8, we can define the optimized non-interactive version of the proof as in Figure 7. Finally, plugging the BBSS scheme from Theorem 5 into Theorem 8 we get the following general result for ZK-ready functions with respect to $\mathfrak{R} = \mathbb{Z}$.

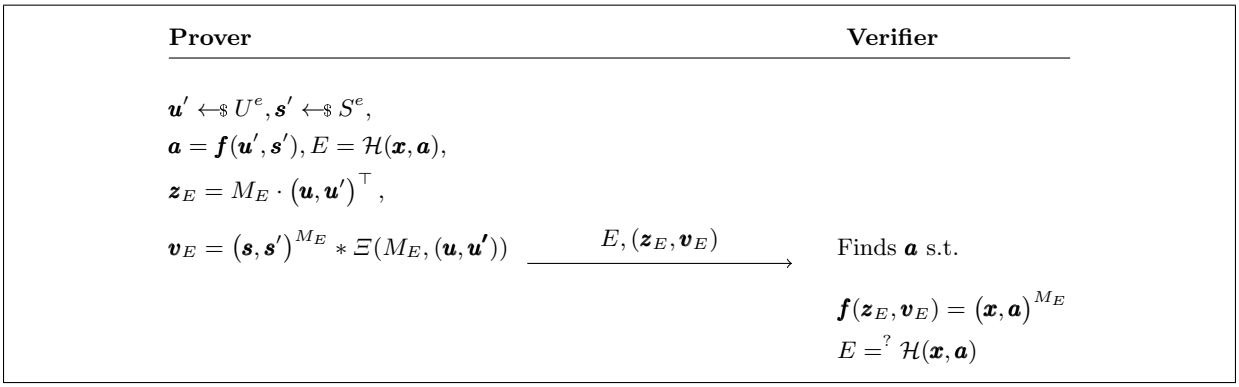

Fig. 7. Optimized NI-Σ-protocol from ZK-ready functions

Corollary 3. *Let $(U, +), (S, \cdot), (X, \cdot)$ finite abelian groups. Let $f : U \times S \to X$ be ZK-ready with respect to $\mathbb{Z}$ and let $\lambda \geq 0$ be a security parameter. Then there exists a $\Sigma-$protocol for the relation $R = \{(\boldsymbol{u}, \boldsymbol{s}; \boldsymbol{x}) \in (U^k \times S^k) \times X^k : f(u_i, s_i) = x_i \ \forall i \in [k]\}$, for $k \geq \lambda$, with $2-$special soundness, knowledge error at most $2^{-\lambda}$ and communication complexity $k + \lambda - 3$ elements of U, S and X (and λ bits for the challenge). Its non-interactive version communicates $k + \lambda - 3$ elements of U, S and λ bits.*

6.1 Improvement for Joye-Libert encryption

We show that Joye-Libert encryption for $\mathbb{Z}_{2^l}$ is a ZK-ready function with respect to both the rings $\mathbb{Z}$ and $\mathbb{Z}_{2^l}$. The former implies that we can apply Corollary 3 to obtain a batch protocol to show knowledge of plaintext and randomness in k Joye-Libert ciphertexts. However, in this case we can do better by considering the encryption function as ZK-ready with respect to $\mathbb{Z}_{2^l}$ and using as secret sharing scheme a version of Shamir over an extension ring, namely a Galois ring. The idea of this construction, at least when we use standard (non-packed) Shamir, is exactly the one in [2] where it was applied to a certain vector commitment scheme construction over $\mathbb{Z}_{2^l}$ instead of Joye-Libert. Moreover, it is a quite direct generalization of Section 4.2 to the setting of fields. However, we have not seen this Σ-protocol applied to Joye-Libert encryption and given its applications, as mentioned in the introduction, we think it is important to point out. In addition, we will also further generalize it by using the packed version of Shamir, so that we can trade communication complexity by soundness.

Joye-Libert encryption. Let (public) $k \in \mathbb{N}$, and two (private) primes with $p \equiv 1 \bmod 2^l$, $q \equiv 3 \mod 4$ and set $N = pq$. Let $g \in \mathbb{Z}_N^*$ of order $\phi(N)/2$. Then let $U = \mathbb{Z}_{2^l}$, $S = \mathbb{Z}_N^*$, $X = \mathbb{Z}_N^*$, $\mathfrak{R} = \mathbb{Z}$; and $f : \mathbb{Z}_{2^l} \times \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^*$ given by $f(u, s) := g^u \cdot s^{2^l}$ where we see u embedded as an integer in $[0, 2^l - 1]$.

f is *not* an homomorphism since the argument u lives in $\mathbb{Z}_{2^l}$ but $g^{2^l} \neq 1$ in $\mathbb{Z}_N^*$, and therefore there will be cases for which the operations over $\mathbb{Z}_{2^l}$ "do not match" the operations over $\mathbb{Z}_N^*$[4].

Lemma 7. *f is ZK-ready with respect to both $\mathfrak{R} = \mathbb{Z}$ and $\mathfrak{R} = \mathbb{Z}_{2^l}$.*

Proof (Sketch). Indeed, let $U = \mathbb{Z}_{2^l}$, $S = X = \mathbb{Z}_N^*$. Consider $g : \mathbb{Z}_{2^l} \longrightarrow \mathbb{Z}_N^*$ such that $g(u) = g^u$ and $h : \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^*$ with $h(s) = s^{2^l}$. Then it is easy to see f satisfies the conditions of Definition 8 for $\mathfrak{R} = \mathbb{Z}$. In the case of $\mathfrak{R} = \mathbb{Z}_{2^l}$, note that x^a for $x \in X$ is computed by embedding a in $[0, 2^l - 1] \subseteq \mathbb{Z}$ and then computing $x^a \in \mathbb{Z}_N^*$. Clearly the action of a defines an endomorphism on $\mathbb{Z}_N^*$. Moreover since $Im\ h = (\mathbb{Z}_N^*)^{2^l}$, $\pi : \mathbb{Z}_N^* \to \mathbb{Z}_N^*/(\mathbb{Z}_N^*)^{2^l}$ where $\pi(x) = x \cdot (\mathbb{Z}_N^*)^{2^l}$. Then $\mathbb{Z}_N^*/(\mathbb{Z}_N^*)^{2^l}$ is a $\mathbb{Z}_{2^l}$-module with the action of $a \in \mathbb{Z}_{2^l}$ on $x \cdot (\mathbb{Z}_N^*)^{2^l}$ being $x^a \cdot (\mathbb{Z}_N^*)^{2^l}$, since $a + b = c \mod 2^l$ implies $x^a \cdot x^b = x^c \cdot y$ with y in $(\mathbb{Z}_N^*)^{2^l}$. From here the rest of the properties can be easily verified.

Next, we compute the function Ξ from Lemma 6. For $u, u' \in \mathbb{Z}_{2^l}$ denote $(u + u')_{\mathbb{Z}}$ the sum in $\mathbb{Z}$ of the representatives of u and u' in $[0, 2^l - 1]$, while $u + u'$ is their sum in $\mathbb{Z}_{2^l}$ (i.e. modulo 2^l). Define the function $\delta : \mathbb{Z}_{2^l} \times \mathbb{Z}_{2^l} \longrightarrow \mathbb{Z}_N^*$ by $\delta(u, u') = g^{q(u,u')}$ where $q(u, u') = \lfloor \frac{(u+u')_{\mathbb{Z}}}{2^l} \rfloor$. Then we get $f(u, s)f(u', s') = f(u + u', ss'\delta(u, u'))$.[5] For $M \in \mathbb{Z}^{n \times m}$, we get $f(\boldsymbol{u}, \boldsymbol{s})^M = f(M \cdot \boldsymbol{u}, \boldsymbol{s}^M * \Xi(M, \boldsymbol{u}))$, where

$$\Xi(M, \boldsymbol{u}) = \begin{pmatrix} \prod_{i=1}^m \xi(M_{1i}, u_i) \\ \vdots \\ \prod_{i=1}^m \xi(M_{ni}, u_i) \end{pmatrix} = \begin{pmatrix} g^{\lfloor \frac{(M_1 \boldsymbol{u})_{\mathbb{Z}}}{2^l} \rfloor} \\ \vdots \\ g^{\lfloor \frac{(M_n \boldsymbol{u})_{\mathbb{Z}}}{2^l} \rfloor} \end{pmatrix}.$$

Note that here for $i = 1, \ldots n$, M_i denote the rows of the matrix M and $(M_i \boldsymbol{u})_{\mathbb{Z}}$ represent the multiplication of the row M_i and the vector $\boldsymbol{u}$ as elements with coordinates in $\mathbb{Z}$ (again embedding the coordinates of $\boldsymbol{u}$ in $[0, 2^l - 1]$). On the other hand $M \cdot \boldsymbol{u}$ denotes a matrix-vector product on $\mathbb{Z}_{2^l}$, where this is computed using of actions of elements of $\mathbb{Z}$ (the matrix entries) on elements on $\mathbb{Z}_{2^l}$ (the coordinates of $\boldsymbol{u}$) and sums in $\mathbb{Z}_{2^l}$.

Now we define our secret sharing scheme for secrets in $(\mathbb{Z}_{2^l})^k$. In order to do this recall the definition of Galois ring $GR(2^l, d)$:

Definition 9. *A Galois ring is a ring of the form $Z_{p^l}[Y]/(F(Y))$ where $F(Y)$ is a polynomial in $Z_{p^l}[Y]$ such that its reduction modulo p is irreducible in $Z_p[Y]$. Any two Galois rings $Z_{p^l}[Y]/(F(Y))$, $Z_{p^l}[Y]/(F'(Y))$ where F, F' are of the same degree d are isomorphic, and hence we denote by $GR(p^l, d)$ any of them.*

Lemma 8 ([1]). *Given a Galois ring $GR(p^l, d)$ the subset $S = \{a_0 + a_1 Y + \cdots + a_{d-1} Y^{d-1} : a_i \in [0, p-1]\}$ is an exceptional set, meaning that for any x, x' in S, $x - x'$ is invertible, and it has p^d elements.*

Lemma 9. *Let S be an exceptional set of $\mathfrak{R}' = GR(p^l, d)$, and $S' = \{\alpha_1, \ldots, \alpha_n\} \subseteq S$ be an exceptional subset of size $n \leq |S| = p^d$. Let $0 < t < n$.*

We define the Shamir secret sharing scheme with space of secrets $\mathcal{S}_0 = (\mathfrak{R}')^{k'}$, randomness space $\mathcal{R} = (\mathfrak{R}')^t$ and spaces of shares $\mathcal{S}_i = \mathfrak{R}'$ for all $i \in [n]$ given by $\mathsf{Sh}(\boldsymbol{s}, \boldsymbol{r}) = (m(\alpha_1), \ldots, m(\alpha_n))$ where $m(X) = r_1 + \cdots + r_t X^{t-1} + s_1 X^t + \cdots + s_{k'} X^{t+k'-1} \in \mathfrak{R}'[X]$. Then this secret sharing scheme has t-privacy, $(t + k')$-reconstruction, and it is linear over $\mathfrak{R}'$.

[4] e.g. $(2^l - 1) + 1 = 0$ in $\mathbb{Z}_{2^l}$, but $f(2^l - 1, 1) \cdot f(1, 1) = g^{2^l} \neq f(0, 1)$.

[5] In the example of the previous footnote if $u = 2^l - 1$, $u' = 1$, we would have $(u + u')_{\mathbb{Z}} = 2^l$, so $q(u, u') = 1$ and $\delta(u, u') = g$. Indeed $f(2^l - 1, 1) \cdot f(1, 1) = g^{2^l} = f(0, g) = f(0, 1 \cdot g)$.

Proof. **Reconstruction:** Let $\mathfrak{R}'[X]_{\leq t+k'-1}$ be the set of polynomials in $\mathfrak{R}'[X]$ with degree at most $t+k'-1$. By [17, Theorem 11.124], if S is an exceptional set (called "admissible" in [17]) of $\mathfrak{R}'$ then for every subset $\{\beta_1,\ldots,\beta_{t+k'}\}$ of S of size $t+k'$, the evaluation map $\mathfrak{R}'[X]_{\leq t+k'-1} \to (\mathfrak{R}')^{t+k'}$ given by $m(X) \mapsto (m(\beta_1),\ldots,m(\beta_{t+k'}))$ is an isomorphism of $\mathfrak{R}'$-modules.

This already implies $t+k'$-reconstruction of the secret sharing scheme, since any $t+k'$ evaluations of $m(X)$ on points of S determine uniquely $m(X)$ and hence $s_1,\ldots,s_k$.

Privacy: As for t-privacy, again by [17, Theorem 11.124] given a subset $\{\beta_1,\ldots,\beta_t\}$ of S of size t, the evaluation map $\mathfrak{R}'[X]_{\leq t-1} \to (\mathfrak{R}')^t$ given by $r(X) \mapsto (r(\beta_1),\ldots,r(\beta_t))$ is an isomorphism.

Therefore, given t shares $a_1,\ldots,a_t$ where $a_i \in \mathfrak{R}'$ and any element $(s_1,\ldots,s_{k'})$ in the space of secrets $(\mathfrak{R}')^{k'}$, define $b_i = a_i - (s_1\beta_i^t + \cdots + s_{k'}\beta_i^{t+k'-1})$. Then there is a unique polynomial $r(X) = r_1 + \cdots + r_tX^{t-1}$ of degree at most $t-1$ such that $r(\beta_i) = b_i$ for $i \in [t]$. Consequently there is exactly one polynomial of the form $m(X) = r_1 + \cdots + r_tX^{t-1} + s_1X^t + \cdots + s_{k'}X^{t+k'-1}$ with $m(\beta_i) = a_i$ for i in $[t]$. Since this is for any fixed set of indices $\{\beta_1,\ldots,\beta_t\}$, any fixed set of shares $a_1,\ldots,a_t$, and any possible secret, there is t-privacy. □

Now we want to recast the SSS in Lemma 9 as a SSS with $\mathcal{S}_0 = (\mathbb{Z}_{p^l}^d)^{k'}, \mathcal{S}_i = \mathbb{Z}_{p^l}^d$, $\mathcal{R} = (\mathbb{Z}_{p^l}^d)^t$. Let $\phi : \mathbb{Z}_{p^l}^d \to \mathfrak{R}' = GR(2^l,d)$, be a module isomorphism, where $\phi(a_1,\ldots,a_d) = a_1 + a_2Y + \cdots + a_dY^{d-1}$. Then we define $\mathsf{Sh}(\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{k'},\boldsymbol{r}_1,\ldots,\boldsymbol{r}_t) = (\phi^{-1}(m(\alpha_1)),\ldots,\phi^{-1}(m(\alpha_n)))$ where $m(X) = \phi(\boldsymbol{r}_1) + \cdots + \phi(\boldsymbol{r}_t)X^{t-1} + \phi(\boldsymbol{s}_1)X^t + \cdots + \phi(\boldsymbol{s}_{k'})X^{t+k'-1}$.
The scheme is clearly linear over $\mathbb{Z}_{p^l}$ because ϕ is a isomorphism of modules. Therefore, it defines a dk'-MSP $(\mathbb{Z}_{p^l}, \{M_i\}_{i\in[n]}, dk')$ where each $M_i \in \mathbb{Z}_{p^l}^{d\times(d+dt)}$.

Note that for $t=1$, calling $\boldsymbol{r} = \boldsymbol{r}_1$ each share is of the form $N_i \cdot (\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{k'})^\top + \boldsymbol{r}$ for some matrix N_i, so $M_i = (N_i|I_{d\times d})$. Note that, when $t=1$, the share for the i-th participant, seen as an element in $GR(2^l,d)$ is given as $m(\alpha_i) = \phi(\boldsymbol{r}) + \phi(\boldsymbol{s}_1)\alpha_i + \cdots + \phi(\boldsymbol{s}_{k'})\alpha_i^{k'}$, which can be seen as the product

$$(\alpha_i, \alpha_i^2, \ldots, \alpha_i^{k'}, 1) \cdot \begin{pmatrix} \phi(\boldsymbol{s}_1) \\ \vdots \\ \phi(\boldsymbol{s}_{k'}) \\ \phi(\boldsymbol{r}) \end{pmatrix}.$$

Now, when considering the secret sharing scheme over $\mathbb{Z}_{p^\ell}^d$, the shares of each participant i are computed by a matrix $M_i \in \mathbb{Z}_{p^l}^{d\times(d+dt)}$. To obtain such M_i, one replaces in the vector $(\alpha_i, \alpha_i^2, \ldots, \alpha_i^{k'}, 1)$ each α_i^j by the $d\times d$ matrix representing multiplication by α_i^j over $\mathbb{Z}_{2^l}^d$, and replaces 1 by $I_{d\times d}$. The former is the matrix that represents the $\mathbb{Z}_{p^l}$-linear map

$$\begin{array}{ccc} \mathbb{Z}_{p^l}^d & \longrightarrow & \mathbb{Z}_{p^l}^d \\ \boldsymbol{x} & \mapsto & \phi^{-1}(\phi(\boldsymbol{x}) \cdot \alpha_i^j) \end{array}.$$

Therefore, the matrix M_i is indeed of the form $N_i|I_{d\times d}$ and the i-th share is given by $\phi^{-1}(m(\alpha_i)) = M_i(\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{k'},\boldsymbol{r})^\top$.

By Remark 8, the scheme in this case satisfies the conditions of Theorem 8. So we can obtain the following result by setting $p=2$, $t=1$, $k=dk'$, $n=2^d$ above:

Corollary 4. *Let $k>0$. For every $d>0$ with $d \mid k$, there exists a Σ-protocol which is a Zero Knowledge Proof of Knowledge for the relation*

$$R_{JL,k} = \{(\boldsymbol{u},\boldsymbol{s};\boldsymbol{x}) \in \mathbb{Z}_{2^l}^k \times (\mathbb{Z}_N^*)^k \times (\mathbb{Z}_N^*)^k : x_i = g^{u_i}s_i^{2^l}\, \forall i \in [k]\}$$

with $(\frac{k}{d}+1)$-special soundness, knowledge error $\frac{k}{d2^d}$, and whose non-interactive version has size d elements of both $\mathbb{Z}_{2^l}$ and $\mathbb{Z}_N^*$, and d bits.

In particular when $k = d$, this proof has error soundness $1/2^k$ and communicates k elements of both $\mathbb{Z}_{2^l}$ and $\mathbb{Z}_N^*$, and k bits.

References

1. M. Abspoel, R. Cramer, I. Damgård, D. Escudero, and C. Yuan. Efficient information-theoretic secure multiparty computation over $\mathbb{Z}/p^k\mathbb{Z}$ via galois rings. In D. Hofheinz and A. Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 471–501. Springer, Heidelberg, Dec. 2019.
2. T. Attema, I. Cascudo, R. Cramer, I. Damgård, and D. Escudero. Vector commitments over rings and compressed Σ-protocols. In E. Kiltz and V. Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 173–202. Springer, Heidelberg, Nov. 2022.
3. T. Attema and R. Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In D. Micciancio and T. Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 513–543. Springer, Heidelberg, Aug. 2020.
4. T. Attema, S. Fehr, and N. Resch. Generalized special-sound interactive proofs and their knowledge soundness. Cryptology ePrint Archive, Paper 2023/818, 2023. `https://eprint.iacr.org/2023/818`. *To appear in TCC 23.*
5. M. Ball, A. Çakan, and T. Malkin. Linear threshold secret-sharing with binary reconstruction. In S. Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography, ITC 2021, July 23-26, 2021, Virtual Conference*, volume 199 of *LIPIcs*, pages 12:1–12:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
6. C. Bouvier, G. Castagnos, L. Imbert, and F. Laguillaumie. I want to ride my BICYCL : BICYCL implements cryptography in class groups. *J. Cryptol.*, 36(3):17, 2023.
7. L. Braun, I. Damgård, and C. Orlandi. Secure multiparty computation from threshold encryption based on class groups. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 613–645, Cham, 2023. Springer Nature Switzerland.
8. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.
9. G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Two-party ECDSA from hash proof systems and efficient instantiations. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 191–221. Springer, Heidelberg, Aug. 2019.
10. G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker. Bandwidth-efficient threshold EC-DSA. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 266–296. Springer, Heidelberg, May 2020.
11. G. Castagnos and F. Laguillaumie. Linearly homomorphic encryption from ddh. In *Topics in Cryptology–CT-RSA 2015*, pages 440–457. Springer, 2015.
12. G. Castagnos, F. Laguillaumie, and I. Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In T. Peyrin and S. Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 733–764. Springer, Heidelberg, Dec. 2018.
13. G. Castagnos, F. Laguillaumie, and I. Tucker. Threshold linearly homomorphic encryption on $\mathbf{Z}/2^k\mathbf{Z}$. In S. Agrawal and D. Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 99–129. Springer, Heidelberg, Dec. 2022.
14. D. Catalano, M. Di Raimondo, D. Fiore, and I. Giacomelli. Mon$\mathbb{Z}_{2^k}$a: Fast maliciously secure two party computation on $\mathbb{Z}_{2^k}$. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 357–386. Springer, Heidelberg, May 2020.
15. R. Cramer and I. Damgård. On the amortized complexity of zero-knowledge protocols. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 177–191. Springer, Heidelberg, Aug. 2009.
16. R. Cramer, I. Damgård, and M. Keller. On the amortized complexity of zero-knowledge protocols. *Journal of Cryptology*, 27(2):284–316, Apr. 2014.
17. R. Cramer, I. Damgård, and J. B. Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
18. R. Cramer and S. Fehr. Optimal black-box secret sharing over arbitrary Abelian groups. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 272–287. Springer, Heidelberg, Aug. 2002.
19. R. Cramer, S. Fehr, and M. Stam. Black-box secret sharing from primitive sets in algebraic number fields. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 344–360. Springer, Heidelberg, Aug. 2005.
20. R. Cramer and C. Xing. Blackbox secret sharing revisited: A coding-theoretic approach with application to expansionless near-threshold schemes. In A. Canteaut and Y. Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 499–528. Springer, Heidelberg, May 2020.

21. P. Das, M. J. Jacobson Jr., and R. Scheidler. Improved efficiency of a linearly homomorphic cryptosystem. In C. Carlet, S. Guilley, A. Nitaj, and E. M. Souidi, editors, *Codes, Cryptology and Information Security - Third International Conference, C2SI 2019, Rabat, Morocco, April 22-24, 2019, Proceedings - In Honor of Said El Hajji*, volume 11445 of *Lecture Notes in Computer Science*, pages 349–368. Springer, 2019.
22. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, Heidelberg, Aug. 1990.
23. M. K. Franklin and M. Yung. Communication complexity of secure computation (extended abstract). In *24th ACM STOC*, pages 699–710. ACM Press, May 1992.
24. R. Gennaro, D. Leigh, R. Sundaram, and W. S. Yerazunis. Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 276–292. Springer, Heidelberg, Dec. 2004.
25. L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 216–231, New York, NY, 1990. Springer New York.
26. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.
27. M. Joye and B. Libert. Efficient cryptosystems from 2^k-th power residue symbols. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 76–92. Springer, Heidelberg, May 2013.
28. M. Karchmer and A. Wigderson. Characterizing non-deterministic circuit size. In *25th ACM STOC*, pages 532–540. ACM Press, May 1993.
29. M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, pages 102–111. IEEE Computer Society, 1993.
30. U. M. Maurer. Unifying zero-knowledge proofs of knowledge. In B. Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.
31. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 308–318, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
32. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.
33. C.-P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, Aug. 1990.
34. H. Xue, M. H. Au, M. Liu, K. Y. Chan, H. Cui, X. Xie, T. H. Yuen, and C. Zhang. Efficient multiplicative-to-additive function from joye-libert cryptosystem and its application to threshold ecdsa. Cryptology ePrint Archive, Paper 2023/1312, 2023. `https://eprint.iacr.org/2023/1312`. *To appear in ACM CCS 23.*
35. M. Zhang, Y. Chen, C. Yao, and Z. Wang. Sigma protocols from verifiable secret sharing and their applications. In J. Guo and R. Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 208–242, Singapore, 2023. Springer Nature Singapore.

Appendix

A Proof of Proposition 2

Privacy: If $T \in \Delta$, we show that $\boldsymbol{\sigma}_T \in \mathrm{Im}\ M_T$ gives no information about the secret by showing that for any two secrets $\boldsymbol{s} = (s_1, \ldots, s_k)$ and $\boldsymbol{s^*} = (s_1^*, \ldots, s_k^*)$ in $\mathfrak{M}^k$ the sets $\mathcal{S}(\boldsymbol{s}, \boldsymbol{\sigma}_T) = \{\boldsymbol{r} \in \mathfrak{M}^e : M_T(\boldsymbol{s}, \boldsymbol{r}) = \boldsymbol{\sigma}_T\}$ and $\mathcal{S}(\boldsymbol{s^*}, \boldsymbol{\sigma}_T) = \{\boldsymbol{r^*} \in \mathfrak{M}^e : M_T(\boldsymbol{s^*}, \boldsymbol{r^*}) = \boldsymbol{\sigma}_T\}$ have the same number of elements.

We construct a bijection between the two sets using the vectors $\boldsymbol{\lambda}_T^{(1)}, \ldots, \boldsymbol{\lambda}_T^{(k)} \in \mathfrak{R}^{k+e}$ promised by property (P1). Concretely define $\boldsymbol{w} = \boldsymbol{\lambda}_T^{(1)}(s_1^* - s_1) + \cdots + \boldsymbol{\lambda}_T^{(k)}(s_k^* - s_k) \in \mathfrak{M}^{k+e}$ (here note each $\lambda_i \in \mathfrak{R}^{k+e}$, and each $(s_i^* - s_i) \in \mathfrak{M}$, so $\boldsymbol{\lambda}_T^{(i)}(s_i^* - s_i)$ is a vector in $\mathfrak{M}^{k+e}$ obtained by letting each coordinate of $\boldsymbol{\lambda}_T^{(i)}$ act on $(s_i^* - s_i)$). Note that since the projection of each $\boldsymbol{\lambda}_T^{(i)}$ to its first k coordinates is the i-th unit vector, $\boldsymbol{w}$ is of the form $(\boldsymbol{s^*} - \boldsymbol{s}, \boldsymbol{a})$ for some $\boldsymbol{a} \in \mathfrak{M}^e$.

The bijection is given by $\boldsymbol{r} \in \mathcal{S}(\boldsymbol{s}, \boldsymbol{\sigma}_T) \mapsto \boldsymbol{r}^* = \boldsymbol{r} + \boldsymbol{a}$. Indeed note that then $(\boldsymbol{s}^*, \boldsymbol{r}^*) = (\boldsymbol{s}, \boldsymbol{r}) + \boldsymbol{w}$ and hence $M_T(\boldsymbol{s^*}, \boldsymbol{r^*})^\top = M_T(\boldsymbol{s}, \boldsymbol{r})^\top + \sum_{i=1}^n M_T(\boldsymbol{\lambda}_T^{(i)})^\top (s_i^* - s_i) = M_T(\boldsymbol{s}, \boldsymbol{r})^\top = \boldsymbol{\sigma}_T$, where we have used $M_T(\boldsymbol{\lambda}_T^{(i)})^\top = \boldsymbol{0}$ (second item of P2). Therefore, indeed $\boldsymbol{r}^* \in S(\boldsymbol{s}^*, \boldsymbol{\sigma}_T)$. Moreover, the map $\boldsymbol{r} \mapsto \boldsymbol{r^*}$ is clearly injective and its inverse (given by $\boldsymbol{r^*} \mapsto \boldsymbol{r} = \boldsymbol{r^*} - \boldsymbol{a}$) takes $\mathcal{S}(\boldsymbol{s^*}, \boldsymbol{\sigma}_T)$ into $\mathcal{S}(\boldsymbol{s}, \boldsymbol{\sigma}_T)$ by symmetry, so the map is indeed a bijection.

Reconstruction: Let $S \in \Gamma$. Note that the total vector of shares received by set S is $\boldsymbol{\sigma}_S^\top = M_S \cdot \boldsymbol{v}^\top$ for some $\boldsymbol{v} = (\boldsymbol{s}, \boldsymbol{r}) \in \mathfrak{M}^{k+e}$. For $i = 1, \ldots, k$ the i-th coordinate of the secret s_i can be recovered from $M_S \cdot \boldsymbol{v}^\top$, as follows. Since $\boldsymbol{\mu}^{(i)} \in \mathrm{Im}(M_S^\top)$ by property (P2) of the definition of k-MSP, there exists a $\boldsymbol{\rho}_S^{(i)} \in \mathfrak{R}^{h_S}$ such that $M_S^\top (\boldsymbol{\rho}_S^{(i)})^\top = (\boldsymbol{\mu}^{(i)})^\top$. Transposing this expression gives $\boldsymbol{\rho}_S^{(i)} \cdot M_S = \boldsymbol{\mu}^{(i)}$. Multiplying on the right both sides by $\boldsymbol{v}^\top$ (i.e. making both sides act on $\boldsymbol{v}^\top \in \mathfrak{M}^e$), we get $\boldsymbol{\rho}_S^{(i)} \cdot \boldsymbol{\sigma}_S^\top = \boldsymbol{\rho}_S^{(i)} \cdot (M_S \cdot \boldsymbol{v}^\top) = \boldsymbol{\mu}^{(i)} \cdot \boldsymbol{v}^\top = s_i$. Hence the i-th coordinate of the secret is determined uniquely by the share vector $\boldsymbol{\sigma}_S$ and $\boldsymbol{\rho}_S^{(i)}$ (which only depends on the MSP and S).

B Details about Σ-protocols from threshold LSSS

B.1 Threshold access structures

In this Section we study what are the optimal challenge sets for *threshold* structures $(\Delta, \Gamma)_{\mathfrak{t},\mathfrak{r},n}$ in our protocols from Theorem 1 and later in Theorem 8. For $\mathfrak{t} > 1$, the extraction number $\nu(\mathcal{C}, \Gamma)$ may depend on the choice of $\mathcal{C}$.[6] In the following theorem we establish that the optimal knowledge error guaranteed by Corollary 1 is obtained by taking $\mathcal{C}$ as the family of *all* subsets of size $\mathfrak{t}$. However, in some cases the same knowledge error can be *also* attained by *strictly smaller* families of challenges, which consequently yield Σ-protocols with the same soundness and smaller communication than using all possible sets of size $\mathfrak{t}$ as challenges. The optimal challenge sets can be characterized in terms of combinatorial designs.

Theorem 9. *For any challenge set $\mathcal{C}$ compatible with $(\Delta, \Gamma)_{\mathfrak{t},\mathfrak{r},n}$, the extraction number $\nu := \nu(\mathcal{C}, \Gamma)$ satisfies $(\nu - 1)/|\mathcal{C}| \leq \binom{\mathfrak{r}-1}{\mathfrak{t}} / \binom{n}{\mathfrak{t}}$. Equality is achieved iff 1) $\mathcal{C}$ only contains challenges of size exactly $\mathfrak{t}$ and 2) every set $A \subseteq [n]$ of size $\mathfrak{r} - 1$ contains exactly the same number of sets from $\mathcal{C}$ (which in that case is necessarily $\nu - 1$). In particular, equality is achieved for $\mathcal{C} = \{E \subseteq [n] : |E| = \mathfrak{t}\}$.*

Proof. Let $\mathcal{A}_{\mathfrak{r}-1}$ the set of all $A \subseteq [n]$ of size exactly $\mathfrak{r} - 1$. For each such A, note there are at most $\nu - 1$ challenge sets $E \in \mathcal{C}$ such that $E \subseteq A$, because if there were ν or more, then their union would be contained

[6] For $\mathfrak{t} = 1$, clearly $\nu(\mathcal{C}, \Gamma) = \mathfrak{r}$ for every compatible $\mathcal{C}$.

in A, so it would have size at most $\mathfrak{r}-1$ and therefore it would not be in Γ contradicting the definition of ν. Since there are $\binom{n}{\mathfrak{r}-1}$ sets A in $\mathcal{A}_{\mathfrak{r}-1}$, we have

$$\sum_{A\in\mathcal{A}_{\mathfrak{r}-1}} |\{E\in\mathcal{C} : E\subseteq A\}| \leq (\nu-1)\binom{n}{\mathfrak{r}-1}$$

We note that equality holds if and only if every A of size $\mathfrak{r}-1$ contains exactly $\nu-1$ sets from $\mathcal{C}$. This is equivalent to condition 2 in the second part of the statement. Indeed, if every set A of size $\mathfrak{r}-1$ contains the same amount of sets from $\mathcal{C}$, this amount must be $\nu-1$: we have already argued that A can contain at most $\nu-1$ such sets, but we also know, by definition of ν, that there are $\nu-1$ sets in $\mathcal{C}$ whose union has size less than $\mathfrak{r}$ and hence has to be contained in some set of size $\mathfrak{r}-1$.

On the other hand, there are exactly $\binom{n-|E|}{\mathfrak{r}-1-|E|}$ sets A in $\mathcal{A}_{\mathfrak{r}-1}$ containing a given $E\in\mathcal{C}$, since this number gives the choices of the remaining $\mathfrak{r}-1-|E|$ elements that we can add to E to get a set of size $\mathfrak{r}-1$. Since $|E|\leq\mathfrak{t}$, we have $\binom{n-|E|}{\mathfrak{r}-1-|E|}\geq\binom{n-\mathfrak{t}}{\mathfrak{r}-1-\mathfrak{t}}$. Therefore

$$\sum_{A\in\mathcal{A}_{\mathfrak{r}-1}} |E\in\mathcal{C} : E\subseteq A| = \sum_{E\in\mathcal{C}}\binom{n-|E|}{\mathfrak{r}-1-|E|}\geq\binom{n-\mathfrak{t}}{\mathfrak{r}-1-\mathfrak{t}}|\mathcal{C}|.$$

Equality holds if and only if every set in $|\mathcal{C}|$ is of size $\mathfrak{t}$ (condition 2 in the second part of the statement)

Putting the two sum bounds together we get

$$\frac{\nu(\mathcal{C},\Gamma)-1}{|\mathcal{C}|}\geq\frac{\binom{n-\mathfrak{t}}{\mathfrak{r}-1-\mathfrak{t}}}{\binom{n}{\mathfrak{r}-1}}=\frac{(\mathfrak{r}-1)\cdot(\mathfrak{r}-2)\cdots(\mathfrak{r}-\mathfrak{t})}{n\cdot(n-1)\ldots(n-\mathfrak{t}+1)}=\frac{\binom{\mathfrak{r}-1}{\mathfrak{t}}}{\binom{n}{\mathfrak{t}}}$$

with equality holding if and only if conditions 1 and 2 in the second part of the statement of the theorem hold.

Finally $\mathcal{C}=\mathcal{A}_{\mathfrak{t}}$, the family of all sets of size $\mathfrak{t}$, clearly satisfies conditions 1 and 2. □

The family of all sets of size $\mathfrak{t}$ is not necessarily the only choice of $\mathcal{C}$ achieving optimal soundness in the theorem above. We define a soundness optimal challenge set as follows.

Definition 10. *A soundness-optimal challenge set $\mathcal{C}$ for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$ is a challenge set compatible with $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$ and such that:*

1. *every $E\in\mathcal{C}$ has size exactly $\mathfrak{t}$;*
2. *every set $A\subseteq[n]$ of size $\mathfrak{r}-1$ contains exactly the same number of sets from $\mathcal{C}$ (which, as we have seen needs to be $\nu-1$).*

Moreover, we say that $\mathcal{C}$ is minimal if it has minimal $\mathcal{C}$ (and equivalently minimal $\nu=2$) among all soundness-optimal challenge set $\mathcal{C}$ for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$.

We will now show that soundness-optimal challenge sets are actually equivalent to combinatorial designs.

Definition 11. *A $u-(n,m,\lambda)$-design consists of a family $\mathcal{B}$ of subsets of $[n]$, called blocks, such that each block contains m elements and each set of u elements from $[n]$ is contained exactly in λ blocks.*

Lemma 10. *Let $\mathcal{B}$ be a $u-(n,m,\lambda)$-design. Consider $\overline{\mathcal{B}}=\{[n]\setminus B : B\in\mathcal{B}\}$. Then:*

- *Every set in $\overline{\mathcal{B}}$ contains $n-m$ elements.*
- *Each set of $n-u$ elements in $[n]$ contains exactly λ blocks.*

Corollary 5. *Let $\mathcal{B}$ be a $(n-\mathfrak{r}+1)-(n,n-\mathfrak{t},\lambda)$-design. Then $\mathcal{C}=\overline{\mathcal{B}}$ is a soundness-optimal challenge set for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$ with extraction number $\nu=\lambda+1$.*

Conversely if $\mathcal{C}$ is a soundness-optimal challenge set with extraction number ν then $\overline{\mathcal{C}}$ is a $(n-\mathfrak{r}+1)-(n,n-\mathfrak{t},\nu-1)$-design.

Proof. Since every set in $\mathcal{C}$ has $\mathfrak{t}$ elements, then $\mathcal{C}$ is compatible with the access structure and satisfies condition 1. By Lemma 10 every set of $\mathfrak{r}-1$ elements in $[n]$ contains exactly $\lambda=\nu-1$ blocks from $\mathcal{C}$. This proves that the union of any $\nu=\lambda+1$ sets in $\mathcal{C}$ has size at least $\mathfrak{r}$, and therefore ν is the extraction number. Moreover, it also establishes condition 2., so the challenge set has optimal knowledge error. The converse is analogous. □

For the case of $\nu=2$ (which corresponds to the classical 2-special-soundness), $\mathcal{B}$ would be a $u-(n,m,1)$ design. This type of designs are called Steiner systems and denoted $S(u,m,n)$ (note the change of order) and have been studied thorougly. For example, it is known that lines in a projective plane over a field $\mathbb{F}_q$ yield a $S(2,q+1,q^2+q+1)$ Steiner system. This in turn is a $2-(q^2+q+1,q+1,1)$-design and gives us soundness-optimal challenge sets for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$ with $\mathfrak{t}=q^2$, $\mathfrak{r}=q^2+q$, $n=q^2+q+1$ and $\nu=2$. Another example is given by the supports of codewords of weight 4 in an extended Hamming code, which yield a $S(3,4,2^s)$ Steiner system, and leads to constructions of soundness-optimal challenge sets for $(\Delta,\Gamma)_{\mathfrak{t},\mathfrak{r},n}$ with $n=2^s$ for any $s\geq 3$ and $\mathfrak{t}=n-4$, $\mathfrak{r}=n-2$, and again with $\nu=2$.

B.2 Optimality of knowledge soundness bound

We show that we cannot in general prove a better bound for knowledge soundness of our general protocol than that of Corollary 1, that is $\frac{\nu-1}{|\mathcal{C}|}$, where ν is the extraction number defined in 7 and $|\mathcal{C}|$ is the size of the challenge space. In particular we present an example where an adversarial prover convinces the verifier of a wrong statement with probability exactly $(\nu(\Delta,\Gamma)-1)/|\mathcal{C}|$ regardless of the access structure, MSP and challenge space used (as long as we are using a MSP over the field $\mathbb{Z}_p$ where the witness live and that Γ is a "maximal" reconstructing family for that MSP, see below).

The language we use for this counter-example is discrete logarithm equality. Consider a cyclic group $\mathbb{G}$ of order a prime p written additively, let $G,H\in\mathbb{G}$ and the following injective group homomorphism

$$\begin{array}{rccc} F:\mathbb{Z}_p & \longrightarrow & \mathbb{G}^2 \\ w & \mapsto & (wG,wH) \end{array}$$

We will also use the notations $F_1(w):=wG$ and $F_2(w):=wH$, so $(F(w)=F_1(w),F_2(w))$. We define the relation

$$R:=\{(w,(X_1,X_2))\in\mathbb{Z}_p\times\mathbb{G}^2 : F(w)=(X_1,X_2)\}.$$

Let (Δ,Γ) be an access structure and let $\mathcal{M}$ be any 1−MSP over $\mathbb{Z}_p$ computing (Δ,Γ). We will assume that Γ is maximal under this condition, i.e. $\mathcal{M}$ is not an 1−MSP over $\mathbb{Z}_p$ for any (Δ,Γ') where $\Gamma\subsetneq\Gamma'$. Let $\mathcal{C}$ be a challenge space compatible with (Δ,Γ) and with extraction number ν. In that case, we have a Σ-protocol for knowledge of preimage of F defined from $\mathcal{M}$ and $\mathcal{C}$ as in Theorem 1, see Figure 8.

We remark that in the final check of the protocol the MSP (more precsiely the submatrix M_E) is acting on the group $\mathbb{G}^2$. This action is defined as the coordinate-wise application of the action on $\mathbb{G}$, so $M_E\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{A}\end{pmatrix}:=M_E\begin{pmatrix}X_1 & X_2\\ A_1 & A_2\end{pmatrix}$, i.e. this can be seen as two checks $\boldsymbol{F_i}(\boldsymbol{z}_E)=M_E\cdot\begin{pmatrix}X_i\\ A_i\end{pmatrix}$ for $i=1,2$, that must pass in order for the verifier to accept.

By definition of ν, there are $\nu-1$ challenge sets $E_1,\ldots,E_{\nu-1}\in\mathcal{C}$ such that their union U is not in Γ. Because U is not in Γ and the maximality condition assumed above, U is not a reconstructing set for the

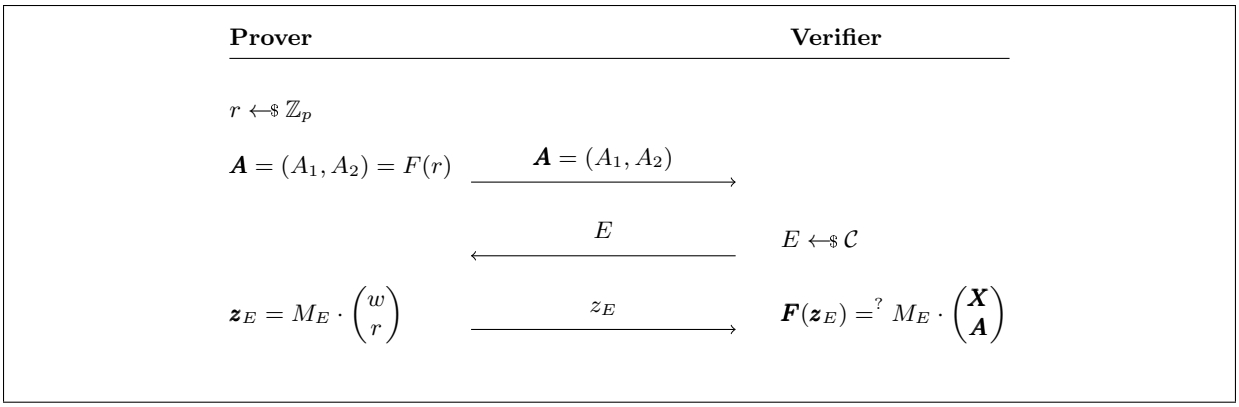

Fig. 8. Σ-protocol for DLE from $\mathcal{M}$ and $\mathcal{C}$

LSSS over $\mathbb{Z}_p$ [7]. Therefore, there must exist two secrets $w \neq w'$ in $\mathbb{Z}_p$ and a vector of shares z_U such that z_U is compatible with both w and w', i.e. there are r and r' such that $z_U = M_U(w, r)^\top = M_U(w', r')^\top$. Given that, an adversarial prover can convince the verifier, with probability exactly $(\nu - 1)/|\mathcal{C}|$, of the fact that $X_1, X_2' = (wG, w'H)$ is in the language, despite it is not. Indeed, she

- Computes and sends $\boldsymbol{A} = (F_1(r), F_2(r'))$ as its first message;
- Receives a challenge E and checks whether $E \in \{E_1, \ldots, E_{\nu-1}\}$;
- In that case, it replies with $\boldsymbol{z}_E = M_E(w, r)^\top$ which also equals $M_E(w', r')^\top$ because $E \subseteq U$.

If $E \in \{E_1, \ldots, E_{\nu-1}\}$, which happens with probability $(\nu - 1)/|\mathcal{C}|$, the strategy above always passes the checks because $\boldsymbol{F_1}(\boldsymbol{z}_E) = \boldsymbol{F_1}(M_E(w, r)^\top) = M_E\,(X_1, A_1)$ and $\boldsymbol{F_2}(\boldsymbol{z}_E) = \boldsymbol{F_2}(M_E(w', r')^\top) = M_E\,(X_2, A_2)$.

[7] Here we are using implicitly that this is a LSSS for $\mathbb{Z}_p$, the field of definition of the MSP, so if a set is not reconstructing with respect to the MSP, it is also not reconstructing with respect to the LSSS over $\mathbb{Z}_p$ induced by it.