Traceable Secret Sharing Schemes for General Access Structures

Oriol Farràs and Miquel Guiot

Universitat Rovira i Virgili, Tarragona, Spain
oriol.farras@urv.cat, miquel.guiot@urv.cat

Abstract. Traceable secret sharing complements traditional schemes by enabling the identification of parties who sell their shares. In the model introduced by Boneh, Partap, and Rotem [CRYPTO'24], a group of corrupted parties generates a reconstruction box R that, given enough valid shares as input, reconstructs the secret. The goal is to trace R back to at least one of the corrupted parties using only black-box access to it. While their work provides efficient constructions for threshold access structures, it does not apply to the general case. In this work, we extend their framework to general access structures and present an information-theoretic traceable scheme supporting them.
In the course of our construction, we also contribute to the study of anonymous secret sharing, a notion recently introduced by Bishop et al. [CRYPTO'25], which strengthens classical secret sharing by requiring that shares do not reveal the identities of the parties holding them. We further advance this area by proposing new and stronger definitions, and presenting an anonymous scheme for general access structures that satisfies them.

Keywords: Secret sharing scheme · anonymity · traceability

This work is the full version of the paper presented at EUROCRYPT 2026, [FG26].

Table of Contents

1 Introduction **3**
- 1.1 Our Results . . . 4
- 1.2 Our Techniques . . . 6
- 1.3 Related Work . . . 10
- 1.4 Open Questions . . . 12
- 1.5 Organization . . . 13

2 Preliminaries **13**
- 2.1 Notation . . . 13
- 2.2 Secret Sharing Schemes . . . 13

3 Anonymous Secret Sharing **15**
- 3.1 Definition . . . 15
- 3.2 Anonymous Schemes for Specific Access Structures . . . 18
- 3.3 Anonymous Scheme for General Access Structures . . . 20
- 3.4 Hiding Anonymous Secret Sharing . . . 25

4 Traceable Secret Sharing for General Access Structures **29**
- 4.1 The Tracing Model . . . 29
- 4.2 Projection Traceable Secret Sharing . . . 30
- 4.3 Traceable Secret Sharing . . . 34

5 A Projection Traceable Scheme for General Access Structures **38**
- 5.1 The Need for Anonymity . . . 39
- 5.2 The Scheme . . . 39
- 5.3 Proof of Theorem 5.1 . . . 45
- 5.4 Further Optimizations . . . 48

6 A Traceable Scheme for General Access Structures **50**
- 6.1 The Need for Hiding the Access Structure . . . 50
- 6.2 The Scheme . . . 53
- 6.3 Proof of Theorem 6.5 . . . 56
- 6.4 Further Optimizations . . . 62

1 Introduction

A secret sharing scheme is a cryptographic primitive in which a dealer shares a secret among a set of parties, ensuring that only certain subsets of them, called authorized, can recover the secret. The collection of these authorized subsets is known as the access structure of the scheme. The primary measure of efficiency for such schemes is the size of the shares.

Secret sharing schemes were introduced independently by Shamir [Sha79] and Blakley [Bla79] in 1979. They presented schemes for threshold access structures, where any subset of parties can recover the secret as long as their size exceeds a given threshold. Since then, secret sharing schemes have been used as a building block in many cryptographic protocols, including secure multiparty computation and threshold cryptography [Bei25]. Although threshold schemes are suitable for many of these applications, some scenarios call for more general access structures, such as in weighted cryptography or attribute-based encryption.

In this context, secret sharing schemes for general access structures, that is, for any monotone collection of subsets of parties, were first presented by Ito, Saito and Nishizeki [ISN87]. Following that, several works presented more efficient schemes for these access structures [BL88,LV18,ABF+19,ABNP20,AN21], but the best known information-theoretic construction still has shares of exponential size.

While these schemes focus on ensuring that unauthorized subsets cannot recover the secret, they do not address other dishonest behaviors. In particular, most constructions do not prevent parties from copying or selling their shares. This lack of accountability can be problematic in certain scenarios, as it gives parties an opportunity to leak or sell their shares without facing consequences.

Recently, to address this issue, Goyal, Song, and Srinivasan [GSS21], and Boneh, Partap, and Rotem [BPR24] introduced the notion of traceable secret sharing schemes. In the model presented in the latter work [BPR24], the adversary uses the shares of an unauthorized subset of parties to construct a reconstruction box, which can then reconstruct the secret once it receives enough valid shares. It is assumed that this reconstruction box is eventually obtained by the authorities, who can interact with it in a black-box manner in order to trace it back to its original constructors. Within this framework, a traceable secret sharing scheme must satisfy two key properties: *traceability*, the ability to identify at least one party whose share was used in building the reconstruction box; and *non-imputability*, which ensures that no honest party is falsely blamed. In addition, the scheme must guarantee that the secret remains hidden throughout the tracing procedure.

Although the aforementioned tracing proposals are efficient in terms of share size, they are limited to threshold access structures. This work builds on the framework of [BPR24], with the aim to provide a traceable secret sharing scheme for general access structures.

1.1 Our Results

In this work, we present a traceable secret sharing scheme for general access structures. To the best of our knowledge, this is the first construction that brings traceability into the general access structure setting, representing an initial step towards efficient traceable secret sharing beyond threshold access structures. As part of our contribution, we extend and generalize the definitions of traceable secret sharing introduced by Boneh, Partap, and Rotem to handle arbitrary access structures.

In the course of our construction, we also contribute to the emerging line of research on *anonymous secret sharing* [PCO20,EBG+24,BGI+25], which strengthens standard secret sharing by requiring that shares do not reveal the identity of the parties they belong to. We enrich this framework by providing further definitions and constructing a new anonymous scheme for general access structures. This notion plays a central role in our traceable scheme and is of independent interest. Our main result is stated below.

Theorem 1.1 (Informal). *For any access structure over n parties, there exists a traceable secret sharing scheme with perfect privacy and statistical correctness, traceability, and non-imputability. Moreover, the tracing procedure only requires $O(\mathrm{poly}(n))$ queries to the reconstruction box.*

As in the work of Boneh, Partap, and Rotem [BPR24], the guarantees of statistical correctness and traceability rely on the fact that the shares and the secret come from a domain of super-polynomial size, which ensures that both tasks succeed with overwhelming probability. Similarly, our statistical guarantee for non-imputability also stems from this domain size. However, in contrast to their approach, we do not rely on one-way functions, as we deliberately design our construction to be fully information-theoretic. In terms of efficiency, although the share size is exponential (as inherent in all known information-theoretic constructions for general access structures), the number of queries required by our tracing algorithm is polynomial.

To construct such a scheme, a necessary first step is to expand and generalize the definitions of traceable secret sharing to the setting of general access structures. This generalization is technically involved, as it must account for the more complex configuration of authorized and unauthorized subsets. More in detail, the threshold case is simpler because all parties are treated uniformly, and authorized subsets are determined solely by their size. Therefore, the reconstruction box only needs to handle queries containing a fixed number of shares, which corresponds to the number of additional parties required to meet the threshold when combined with the corrupted subset. In contrast, for general access structures, authorized subsets may have an intricate description and do not depend only on their size. As a result, the definition of reconstruction box must be generalized to handle input queries consisting of any unauthorized subset that, when combined with the corrupted parties, forms an authorized set.

Additionally, we also extend the notion of reconstruction boxes to handle a more challenging and realistic setting in which they may leak only partial

information about the secret, rather than the full value. In particular, we focus on the minimal case where the reconstruction box outputs just a single bit, representing the smallest unit of useful information that can still be exploited by the tracer. These aspects are discussed in more detail in Section 4.

Furthermore, we show that we can get efficient traceable schemes with polynomial share size for the family of *low slices*. These are access structures over n parties with a threshold $t \ll n$, where the authorized subsets include all subsets of size greater than t, along with an arbitrary collection of subsets of size exactly t. The statement is given below.

Corollary 1.2 (Informal). *For any t-slice access structure over n parties with $t = O(1)$, there exists a traceable secret sharing scheme with perfect privacy and statistical reconstruction, traceability and non-imputability. Moreover, its share size is $\tilde{O}(\text{poly}(n))$ and the tracing procedure only requires $\tilde{O}(\text{poly}(n))$ queries to the reconstruction box.*

As pointed out in [BPR24], to trace efficiently, i.e., with a polynomial number of queries, shares cannot be linked to the parties to which they belong. Although this requirement may seem counterintuitive, given that the goal of traceability is to identify corrupted parties based on a reconstruction box that operates on their shares, it is in fact essential (see Section 5.1).

To formalize this requirement, we build upon the notion of anonymous secret sharing schemes [PCO20,EBG+24,BGI+25], where the identity of any unauthorized subset of parties remains hidden from their shares. We extend the definitions introduced by Bishop et al. [BGI+25] by additionally requiring that the reconstruction process itself does not leak any information about the identities of the participating parties, and we provide a scheme satisfying this strengthened notion of anonymity. Furthermore, we show that slight modifications to classical ideal schemes for threshold access structures also satisfy our anonymity requirements. These results are formalized in the following statements.

Theorem 1.3 (Informal). *For any access structure over n parties, there exists an anonymous secret sharing scheme with perfect privacy and statistical correctness. In the special case of threshold access structures, perfect correctness can also be achieved.*

Finally, we take a step further and define the notion of *hiding anonymous secret sharing*, which strengthens anonymity by adding two requirements: the reconstruction algorithm must operate independently of the access structure, and the shares must not leak any information about it. We construct such a scheme for general access structures by building on top of our anonymous scheme, and it serves as a key component in the design of our traceable secret sharing scheme.

Theorem 1.4 (Informal). *For any access structure over n parties, there exists a hiding anonymous secret sharing scheme with perfect privacy and statistical correctness.*

Indeed, we show that hiding the access structure is a necessary step if one wants to trace back all corrupted parties, rather than identifying only one of them.

Theorem 1.5 (Informal). *There does not exist any traceable secret sharing scheme for general access structures that traces back all the corrupted parties and keeps the access structure public.*

1.2 Our Techniques

Our construction of a traceable secret sharing scheme for general access structures proceeds in three main steps. First, we build an anonymous scheme by extending the classical DNF construction. Next, we enhance this scheme to satisfy the stronger notion of hiding anonymity, ensuring that neither the shares nor the reconstruction algorithm reveal any information about the access structure. Finally, we build on this hiding anonymous scheme to design our traceable construction. This approach is motivated by an observation from [BPR24], who noted that for certain threshold access structures, efficient tracing (i.e. a polynomial number of queries) requires anonymity.

Anonymous Schemes for General Access Structures. To construct an anonymous secret sharing scheme, where neither the shares nor the reconstruction procedure reveal the identity of the parties, we adopt a combinatorial approach. Specifically, we eliminate any reliance on the configuration of the access structure, as this could potentially leak information about which parties hold which shares. Instead, we additively share the secret independently for each authorized subset.

However, this strategy results in each party receiving one share for every authorized subset it belongs to, and thus the size of a party's share leaks the number of such subsets, breaking anonymity. To address this, we assign dummy random shares to each party for every authorized subset it does not belong to, thereby equalizing the share size of each party.

To enable reconstruction, each authorized subset of parties must identify the portion of their shares corresponding to their subset. However, they cannot do so directly because their shares reveal no information about their role in the access structure. To overcome this challenge, we employ an indicator technique: in addition to the actual secret, we also share a fixed indicator value (namely, 0) using the same anonymous scheme. This allows the parties to test different combinations of their shares and identify the correct subset (i.e., the one that reconstructs the indicator). Moreover, if the domain of shares is sufficiently large, the probability that an incorrect reconstruction involving dummy shares yields the indicator is negligible.

When a subset of parties want to reconstruct the secret, we need to ensure that the process does not reveal their identities. For that, each party only provides the portion of its shares that is labeled by the size of the subset. This

approach aligns with a common feature in many secret sharing schemes, where only a specific part of each share is needed during reconstruction. We formalize this idea through a deterministic algorithm called Projection, which models a local computation that each party performs on its share based solely on the size of the reconstructing subset. This step is crucial for preserving anonymity, as revealing the entire share during reconstruction could inadvertently expose party identities. For example, if the reconstructing subset is not minimal, exposing full shares might reveal which of its proper subsets are minimal, thereby leaking structural information of the parties.

Finally, we note that our anonymous scheme differs from the one proposed by Bishop et al. [BGI+25], and we discuss these differences in detail in Section 3.1.

Example 1.6. We illustrate these ideas with a toy access structure Γ over three parties p_1, p_2, p_3 given by

$$\Gamma = \{\{p_1, p_2\}, \{p_2, p_3\}, \{p_1, p_2, p_3\}\}.$$

Let $\mathbb{Z}_m$ be the ring of integers modulo m for a sufficiently large m, and let $s \in \mathbb{Z}_m$ be the secret. First, we show how to share the secret s among Γ:

1. For each $A \in \Gamma$, share s independently among A with the additive scheme[1], i.e.

$$\mathsf{sh}_1 = \{r^s_{1,1}, r^s_{3,1}\},\ \mathsf{sh}_2 = \{s - r^s_{1,1}, r^s_{2,2}, r^s_{3,2}\},\ \mathsf{sh}_3 = \{s - r^s_{2,2}, s - r^s_{3,1} - r^s_{3,2}\}.$$

2. For each $A \in \Gamma$, send dummy random values to the parties not in A[2], i.e.

$$\mathsf{sh}_1 = \{r^s_{1,1}, r^s_{2,1}, r^s_{3,1}\}, \qquad \mathsf{sh}_2 = \{s - r^s_{1,1}, r^s_{2,2}, r^s_{3,2}\},$$
$$\mathsf{sh}_3 = \{r^s_{3,1}, s - r^s_{2,2}, s - r^s_{3,1} - r^s_{3,2}\},$$

   where the blue color indicates that these shares are not the result of any sharing process, but are instead sampled uniformly at random from $\mathbb{Z}_m$.

3. Repeat the steps (1) and (2) with the indicator 0 and group the parts of the shares in pairs of the form (indicator, secret), i.e.

$$\mathsf{sh}_1 = \{(r^0_{1,1}, r^s_{1,1}), (r^0_{2,1}, r^s_{2,1}), (r^0_{3,1}, r^s_{3,1})\},$$
$$\mathsf{sh}_2 = \{(-r^0_{1,1}, s - r^s_{1,1}), (r^0_{2,2}, r^s_{2,2}), (r^0_{3,2} r^s_{3,2})\},$$
$$\mathsf{sh}_3 = \{(r^0_{3,1}, r^s_{3,1}), (-r^0_{2,2}, s - r^s_{2,2}), (-r^0_{3,1} - r^0_{3,2}, s - r^s_{3,1} - r^s_{3,2})\}.$$

Next, we show how the authorized subset $\{p_2, p_3\}$ recovers the secret s from their shares $\mathsf{sh}_2, \mathsf{sh}_3$:

[1] At this point, the size of the share of p_2 is different from the size of the shares of p_1 and p_3, which reveals its identity.

[2] Now, the size of the share of p_2 is equal from the size of the shares of p_1 and p_3, hiding its identity.

1. Since p_2, p_3 is an authorized subset of size two, each party computes its projected share of level two. These projected shares correspond to the portions of their shares associated with all authorized subsets of size two, i.e.

$$\mathsf{psh}_2 = \{(-r^0_{1,1}, s - r^s_{1,1}), (r^0_{2,2}, r^s_{2,2})\},\ \mathsf{psh}_3 = \{(r^0_{3,1}, r^s_{3,1}), (-r^0_{2,2}, s - r^s_{2,2})\}.$$

2. Reconstruct for all the indicator shares to check which indicator corresponds to the value 0, i.e.

$$-r^0_{1,1} + r^0_{3,1} \neq 0, \quad r^0_{2,2} + (-r^0_{2,2}) = 0.$$

3. For the corresponding component, reconstruct the secret using its shares, i.e.

$$r^s_{2,2} + (s - r^s_{2,2}) = s.$$

Hiding Anonymous Schemes for General Access Structures. To enhance our anonymous scheme to satisfy the hiding property, we first observe that its reconstruction algorithm already operates independently of the access structure: authorized subsets can locally test combinations of their shares using the indicator technique, without requiring any knowledge of the access structure itself. Then, what remains is to ensure that the shares do not leak information about the access structure. In our anonymous scheme, the size of any share reflects the number of authorized subsets in the access structure, which leaks information about it.

To prevent this, we increase the shares of each party with a dummy share for each unauthorized subset of the access structure. As a result, every party receives a share of uniform size, determined solely by the total number of parties. This modification suffices to transform our anonymous scheme into a hiding anonymous one.

Traceable Schemes for General Access Structures. The core idea behind our traceable secret sharing scheme is to make the reconstruction box itself reveal the identities of the corrupted parties who built it. To keep the exposition clear, in this discussion we assume that the reconstruction box outputs the full secret rather than a single bit. The technically more challenging and general case, in which the reconstruction box outputs only one bit, is handled and explained in detail in Section 6.

At a high level, omitting some technical subtleties discussed later in Section 4, this is achieved by exploiting two key properties of our hiding anonymous scheme. First, sets of shares generated from the same scheme but corresponding to different secrets are indistinguishable to unauthorized subsets.[3] Second, due to anonymity, these shares cannot be linked to any specific parties.

Using these facts, we proceed as follows. For each unauthorized subset, we select one party and assign it a unique secret identifier. We then additively

[3] This property is referred to as *multi-dealer anonymity* in the work of Bishop et al. [BGI+25], and is discussed in more detail in Section 3.

share this identifier for an authorized subset that contains the unauthorized subset. The parties in the unauthorized subset receive their corresponding additive shares, while all other parties in the access structure receive dummy shares[4]. The remaining real additive shares are kept by the tracer and later used as queries to the reconstruction box.

More specifically, in order to trace the corrupted parties, the tracer iteratively tests all such retained shares by querying the reconstruction box with them and checking whether the output matches the identifier of a party. Since the reconstruction box cannot distinguish these shares from the valid ones associated with the actual secret, it accepts them as legitimate input. In particular, when the query includes the retained shares corresponding to the unauthorized subset that built the reconstruction box, it reconstructs the associated identifier, thereby revealing the identity of one of the corrupted parties. The hiding property of our scheme, which ensures that no information about the access structure is leaked, plays a subtle but essential role in this step: it prevents the reconstruction box from detecting inconsistencies in the reconstruction process, a technical detail further discussed in Section 6.

Example 1.7. We now illustrate these ideas again with the toy access structure Γ over three parties p_1, p_2, p_3 given by

$$\Gamma = \{\{p_1, p_2\}, \{p_2, p_3\}, \{p_1, p_2, p_3\}\}.$$

Let $\mathbb{Z}_m$ be the ring of integers modulo m for a sufficiently large m, let $s \in \mathbb{Z}_m$ be the secret, and assume that we have already shared s among $\{p_1, p_2, p_3\}$ with a hiding anonymous scheme Σ for Γ.

First, we show how we add traceability for the unauthorized subset $\{p_1\}$:

1. Select an identifier $\mathsf{id}_1 \in \mathbb{Z}_m$.
2. Select an authorized subset A of Γ containing $\{p_1\}$, i.e.

$$A = \{p_1, p_2\}.$$

3. Generate the shares of id_1 according to Σ for the authorized subset A, i.e.

$$(\mathsf{sh}_1^{\mathsf{id}_1}, \mathsf{sh}_2^{\mathsf{id}_1}, \mathsf{sh}_3^{\mathsf{id}_1}) \leftarrow \Sigma(\mathsf{id}_1, A),$$

   where the blue color indicates that these shares are not the result of any sharing process, but are instead sampled uniformly at random from $\mathbb{Z}_m$.
4. Generate a dummy random share sh_2^r for the party p_2.[5]
5. Send $\mathsf{sh}_1^{\mathsf{id}_1}$, sh_2^r, and $\mathsf{sh}_3^{\mathsf{id}_1}$ to p_1, p_2, and p_3 respectively.
6. Retain $\mathsf{sh}_2^{\mathsf{id}_1}$ for the tracer.

[4] These dummy shares ensure anonymity by keeping share sizes uniform.

[5] Since $p_3 \notin A = \{p_1, p_2\}$, $\mathsf{sh}_3^{\mathsf{id}_1}$ is already a dummy random share.

The procedure to add traceability for the remaining unauthorized subsets is analogous. Next, we show how to trace back the corrupted subset $\{p_1\}$:

1. Query the reconstruction box R with each of the shares sh given to the tracer during the sharing phase, gathering the corresponding outputs , i.e.

$$\mathsf{output} \leftarrow R(\mathsf{sh}).$$

2. For each query, check if the output of the reconstruction box R is the indicator id of any of the parties. If the check passes, accuse the corresponding party, i.e.

$$\mathsf{output} \stackrel{?}{=} \mathsf{id}.$$

3. Since the corrupted set $\{p_1\}$ constructed the reconstruction box R, with high probability it will only output the indicator id_1 when queried with $\mathsf{sh}_2^{\mathsf{id}_1}$, i.e.

$$\mathsf{id}_1 \leftarrow R(\mathsf{sh}_2^{\mathsf{id}_1}).$$

   Therefore, party p_1 is accused.

1.3 Related Work

In this section, we review prior work on the construction of secret sharing schemes for general access structures, the development of anonymous secret sharing schemes, and the design of traceable secret sharing schemes.

Secret Sharing Schemes for General Access Structures. In 1987, Ito, Saito, and Nishizeki [ISN87] presented the first secret sharing scheme for general access structures, in which the share size is 2^n. A year later, Benaloh and Leichter [BL88] provided a method for constructing a scheme from any monotone Boolean function that reduced the share size to $2^{n-o(n)}$. For nearly three decades, there was little progress in improving the efficiency of secret sharing for general access structures. This changed with the breakthrough work of Liu and Vaikuntanathan [LV18], who constructed schemes with share size $2^{0,994n}$, based on a cryptographic primitive known as conditional disclosure of secrets [LVW17,LVW18]. Their result initiated a new line of research that gradually built more efficient schemes [ABF+19,ABNP20,AN21], culminating in the current best-known construction, which has a share size of $\left(\frac{3}{2}\right)^n = 2^{0.585n}$.

Despite much effort, the known lower bounds for general access structures remain significantly weaker. Csirmaz [Csi96,Csi97] proved the best known bound to date, showing that there exists an access structure that requires a share size of at least $\Omega(\frac{n}{\log n})$ and a total share size of $\Omega(\frac{n^2}{\log n})$. Narrowing the gap between these lower and upper bounds remains a major open problem in the study of secret sharing for general access structures.

Anonymous Secret Sharing Schemes. The notion of anonymous secret sharing was initially explored through the lens of *anonymous reconstruction*, where the goal is to recover the secret without using the identities of the participating parties. Early work by Stinson and Vanstone [SV88] examined this concept in the setting of threshold access structures, under the assumptions of perfect correctness and privacy. Subsequent results by Phillips and Phillips [PP92] investigated the feasibility of anonymous ideal secret sharing schemes, proving that such schemes are possible only for threshold structures with threshold values $t \in \{1, n\}$, or when the access structure is the closure of a complete bipartite graph. Blundo and Stinson [BS97] expanded this study by considering both threshold and certain infinite classes of non-threshold structures. For some of these, they established a lower bound of $\Omega(\log n)$ in the information ratio.

More recently, the notion of anonymous reconstruction has been extended to encompass *share anonymity*, which requires that shares from an unauthorized subset do not reveal any information about the identities of the parties that hold them. Early contributions in this direction include the works of [GMO03,PCO20], with the latter providing the first general construction for arbitrary access structures. This line of work was further developed by Eldridge et al. [EBG+24], who constructed efficient statistical anonymous schemes for threshold structures and demonstrated their relevance in practical scenarios, such as mitigating stalker detection threats in Bluetooth tracking devices (e.g., Apple AirTags). Later, Bishop et al. [BGI+25] formalized the notion of anonymous secret sharing by introducing a hierarchy of anonymity definitions, improving the general construction of [PCO20], and presenting both exponential lower bounds for information-theoretic schemes and efficient constructions in the computational setting. Finally, Con [Con25] showed that Shamir's original scheme could be adapted to achieve perfect anonymity for gap-threshold structures. A detailed comparison between our work and that of Bishop et al. [BGI+25] is provided in Section 3.

Traceable Secret Sharing Schemes. The notion of traceable secret sharing was first explored by Goyal, Song, and Srinivasan [GSS21], who proposed a construction for threshold access structures based on the Goldreich-Levin decoding algorithm [GL89]. Their scheme incurs a quadratic overhead in share size compared to standard threshold schemes. Subsequently, Boneh, Partap, and Rotem [BPR24] revisited this concept and introduced an alternative traceability model, presenting two new constructions for threshold access structures based on the classical schemes of Shamir [Sha79] and Blakley [Bla79]. Notably, these schemes achieve only a constant overhead in share size. For a detailed comparison between the two traceability models, we refer the reader to [BPR24].

Following these foundational contributions, recent works have extended traceable secret sharing to broader settings. Hoffmann [Hof24] proposed a construction for weighted threshold access structures using Mignotte's scheme [Mig83], while Baghery et al. [BEMS25] designed a traceable verifiable secret sharing scheme for threshold access structures. Moreover, some new lines of research have

introduced notions closely related to traceable secret sharing, such as secret sharing with snitching [DFLM24], traceable verifiable random functions [BPR25], and deniable secret sharing [CDK+25].

After making public this work, Goyal, Jain, and Partap [GJP25] proposed an alternative construction of traceable secret sharing schemes for general access structures. Their approach, however, differs substantially from ours. First, they adopt a modified and incomparable setting in which the reconstruction box also includes a label specifying an unauthorized subset I, and correctness is only required when the box is queried with the shares of all parties in I. This modeling choice eliminates the need for anonymous secret sharing. Moreover, although their construction is efficient in terms of share size, it is computational in nature and relies on indistinguishability obfuscation, whereas our scheme is information-theoretic and linear.

1.4 Open Questions

Our results raise several open questions about traceable secret sharing schemes. In the following, we highlight four topics in which future work could lead to significant improvements of state-of-the-art solutions.

Information-theoretic Traceable Secret Sharing. A clear direction for future research is to construct traceable schemes for general access structures using more efficient underlying schemes than the DNF-based one. In particular, it is natural to ask whether any of the recent constructions [LV18,ABF+19,ABNP20,AN21], which achieve significantly smaller share size, can be extended to support traceability. Another related question is to develop efficient traceable secret sharing schemes for useful families of access structures, such as graphs or slices. As in the case of threshold access structures, their structural constraints might allow for more efficient solutions.

Anonymous Secret Sharing. With respect to anonymous schemes, an important direction is to understand how our extended notions of anonymous reconstruction and hiding anonymity relate to the anonymity definitions introduced by Bishop et al. [BGI+25]. These definitions capture different aspects of anonymity, and a formal comparison may clarify their relative strength or applicability. In addition, it would be valuable to explore the relationship between anonymous and traceable schemes in more detail. Specifically, one could investigate whether it is possible to construct traceable schemes that do not require anonymity for some access structures, or conversely, to prove whether anonymity is an essential condition for achieving traceability in any access structure. Understanding this connection could lead to more efficient constructions or reveal fundamental insights into the limitations of these types of schemes.

Lower Bounds. Another important open question is understanding the lower bounds for traceable and anonymous schemes. Unlike common secret sharing

schemes, which are mainly focused on privacy, anonymous and traceable schemes must also guarantee that the identities of the participants remain hidden or that leaked shares can be traced back to their parties. These extra requirements may impose extra constraints on the constructions of such schemes, potentially leading to higher lower bounds on the share size. In this regard, Bishop et al. [BGI+25] provided exponential lower bounds for schemes that satisfy a stronger notion of anonymity, which they call *fully anonymous schemes*. This notion appears to be stronger than the anonymity required for traceability, leaving open the challenge of establishing lower bounds for traceable schemes and for weaker forms of anonymity.

1.5 Organization

In Section 2, we present the necessary preliminaries on secret sharing schemes. In Section 3, we introduce the notions of anonymous and hiding anonymous secret sharing schemes, highlight their similarities and differences with the definitions proposed by Bishop et al. [BGI+25], and construct such schemes for general access structures, proving Theorem 1.3, and Theorem 1.4. In Section 4, we extend the definition of traceable secret sharing introduced by Boneh, Partap, and Rotem [BPR24] to support arbitrary access structures. In Section 5, we present a general construction that satisfies a relaxed notion of traceability. Later, in Section 6, we build upon this construction to obtain a traceable secret sharing scheme for general access structures, proving Theorem 1.1 and Corollary 1.2.

2 Preliminaries

2.1 Notation

We notate $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ for the sets of natural, integer, and real numbers, respectively. For $m \in \mathbb{N}$, we denote the ring of integers modulo m by $\mathbb{Z}_m$, and for any prime power $q \in \mathbb{N}$, we notate $\mathbb{F}_q$ for the finite field of q elements.

For $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ as $[n]$. For a set S, we denote its cardinal by $|S|$, its power set by 2^S, and we denote by $s \leftarrow_\$$ the process of sampling a value s from the uniform distribution over S. For a pair of distributions X, Y defined over the same domain Ω, we define the statistical distance between them as $\mathrm{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$.

We denote vectors $\boldsymbol{x}$ using bold symbols, their i-th coordinates as x_i, and their support as $\mathrm{supp}(\boldsymbol{x}) = \{i \in [n] \ : \ x_i \neq 0\}$. The unary vector is denoted by $\mathbf{1}^n \in \{0,1\}^n$, and the zero vector is denoted by $\mathbf{0}^n \in \{0,1\}^n$. For any vectors $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$, we say that $\boldsymbol{x} \leq \boldsymbol{y}$ if and only if $x_i \leq y_i$ for all $i \in [n]$.

We say that a function $\mathrm{negl} : \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial $p(\lambda)$ there exists $\lambda_0 \in \mathbb{N}$ such that $\mathrm{negl}(\lambda) \leq \frac{1}{p(\lambda)}$ for every $\lambda \geq \lambda_0$.

2.2 Secret Sharing Schemes

In the following, we present the basics of secret sharing schemes.

Definition 2.1 (Access Structure). *Let $A = [n]$ be a set of n parties. A collection $\Gamma \subseteq 2^A$ is monotone if $B \in A$ and $B \subseteq C$ imply that $C \in \Gamma$. An n-party access structure is a monotone collection $\Gamma \subseteq 2^A$ of non-empty subsets of A. Sets in Γ are called* authorized*, and sets not in Γ are called* unauthorized*.*

Sometimes, it is convenient to view an access structure as a monotone Boolean function.[6]

Definition 2.2. *A Boolean function $f_\Gamma : \{0,1\}^n \to \{0,1\}$ represents an access structure Γ over n parties if it holds that $f_\Gamma(\boldsymbol{x}) = 1$ if and only if $\mathrm{supp}(\boldsymbol{x}) \in \Gamma$.*

Since every access structure Γ is monotone by definition, f_Γ is also monotone. In this work, we alternate between representing access structures as monotone collections of subsets or as monotone Boolean functions, depending on which representation is more convenient at any given time.

We now define information-theoretic secret sharing schemes.

Definition 2.3 (Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let Γ be an access structure over n parties. A secret sharing scheme for Γ is a pair of a randomized algorithm* Share *and deterministic algorithm* $\mathsf{Reconstruction}$ *such that*

- ***Perfect Correctness.*** *For any secret $s \in \mathcal{S}$ and any authorized set $A \in \Gamma$, it holds that*
$$\mathbf{Pr}[s = \mathsf{Rec}(\mathsf{Share}(s)_A, A)] = 1,$$
where $\mathsf{Share}(s)_A$ denotes the restriction of the output of $\mathsf{Share}(s)$ to the parties in A.
- ***Perfect Privacy.*** *For any secrets $s, s' \in \mathcal{S}$, any unauthorized set $B \notin \Gamma$, and any possible set of shares $\{\mathsf{sh}_i\}_{i \in B}$, it holds that*
$$\mathbf{Pr}[\{\mathsf{sh}_i\}_{i \in B} = \mathsf{Share}(s)_B] = \mathbf{Pr}[\{\mathsf{sh}_i\}_{i \in B} = \mathsf{Share}(s')_B].$$

In the above definition, we imposed perfect correctness and privacy: reconstruction must succeed with probability one for every authorized subset, and the distributions of shares held by any unauthorized subset must be identical for any pair of secrets. However, these requirements can be relaxed. Specifically, given a security parameter $\lambda \in \mathbb{N}$, we may require reconstruction to succeed with high probability, and we may allow the distributions corresponding to different secrets on unauthorized subsets to have small statistical distance with respect to λ. These relaxed notions give rise to the definition of statistical secret sharing schemes.

Definition 2.4 (Statistical Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|S| \geq 2$, let $\lambda \in \mathbb{N}$ be the security parameter, and let Γ be an access structure over n parties. A statistical secret sharing scheme for Γ is a pair of a randomized algorithm* Share *and deterministic algorithm* $\mathsf{Reconstruction}$ *such that*

[6] A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is monotone if for every $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$ such that $\boldsymbol{x} \leq \boldsymbol{y}$, it holds that $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$.

- ***Statistical Correctness.*** *For any secret $s \in \mathcal{S}$ and any authorized set $A \in \Gamma$, there exists a negligible function* negl *such that*

$$\mathbf{Pr}[s = \mathsf{Rec}(\mathsf{Share}(\mathbf{1}^\lambda, s)_A, A)] \geq 1 - \mathrm{negl}(\lambda),$$

  where $\mathsf{Share}(\mathbf{1}^\lambda, s)_A$ *denotes the restriction of the output of* $\mathsf{Share}(\mathbf{1}^\lambda, s)$ *to the parties in A.*
- ***Statistical Privacy.*** *For any secrets $s, s' \in \mathcal{S}$, and any unauthorized set $B \notin \Gamma$, there exists a negligible function* negl *such that*

$$\mathrm{SD}(\mathsf{Share}(\mathbf{1}^\lambda, s)_B, \mathsf{Share}(\mathbf{1}^\lambda, s')_B) \leq \mathrm{negl}(\lambda).$$

In practice, it is common to combine both definitions, leading to schemes that provide perfect correctness with statistical privacy, or statistical correctness with perfect privacy. This hybrid approach is the one we will adopt throughout this work.

3 Anonymous Secret Sharing

In this section, we study anonymous secret sharing schemes. We begin by introducing new definitions that extend the existing framework and compare them with the notions proposed by Bishop et al. [BGI+25]. Next, we formally prove that simple and efficient anonymous schemes can be constructed for specific families of access structures. Later, we present a general construction that realizes any access structure under our anonymity definitions, and we compare it with the schemes of [PCO20,BGI+25]. Finally, we introduce the notion of *hiding anonymous schemes* and construct such a scheme for general access structures.

3.1 Definition

Anonymous secret sharing schemes, as defined by Bishop et al. [BGI+25], extend classical secret sharing by adding two properties beyond correctness and privacy. The first, *share anonymity*, requires that the shares of any unauthorized subset do not reveal any information about the identities of the parties holding them. The second, *anonymous reconstruction*, ensures that the reconstruction procedure can be performed without knowing the identities of the parties involved.

To formalize share anonymity, Bishop et al. [BGI+25] introduce a hierarchy of three definitions: *single-dealer anonymity*, *multi-dealer anonymity*, and *uniform anonymity*. The weakest, single-dealer anonymity, requires the shares of any unauthorized subset to be indistinguishable from those of any other unauthorized subset of the same size. Multi-dealer anonymity strengthens this by requiring indistinguishability between shares derived from the same secret and shares derived from two independently chosen secrets. Finally, uniform anonymity is the strongest notion, requiring the shares of any unauthorized subset to be indistinguishable from uniformly random values. These definitions are related,

with uniform anonymity implying multi-dealer anonymity, which in turn implies single-dealer anonymity. In their work [BGI+25], schemes satisfying both uniform anonymity and anonymous reconstruction are called *fully anonymous schemes*. We refer the reader to [BGI+25] for the precise definitions and a detailed comparison.

Our definition of anonymous secret sharing extends this framework by strengthening the notion of anonymous reconstruction. Specifically, we require not only that the reconstruction algorithm does not rely on the identities of the parties, but also that the inputs provided to it do not leak any information about those identities. That is, the inputs given by any two authorized subsets of the same size must be indistinguishable. We refer to this stronger requirement as *projected anonymous reconstruction*.

To support this, we introduce an additional algorithm, $\mathsf{Projection}$, which models a local transformation applied by each party to their share. This deterministic algorithm takes as input a share and an integer indicating the size of the subset attempting reconstruction, and returns the portion of the share relevant for authorized subsets of that size. This approach reflects a natural principle: in many secret sharing schemes, only a portion of each share is needed for reconstruction.

Intuitively, since the goal is to prevent the reconstruction process from leaking any information about the parties, it must avoid relying on the internal structure of the access structure. Therefore, the only information that parties are allowed to use is the size of the subset they belong to, as this is inherently revealed during reconstruction. The $\mathsf{Projection}$ algorithm captures this constraint, ensuring that each party only exposes the minimal part of their share needed for reconstructing the secret.

In this context, our anonymous secret sharing schemes must satisfy the share anonymity condition (in the sense of [BGI+25]) and projected anonymous reconstruction. We now formalize the definition.

Definition 3.1 (Anonymous Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let Γ be an access structure over n parties. An anonymous secret sharing scheme for Γ it is a triple of the following algorithms:*

- $\mathsf{Share}(s) \mapsto \{\mathsf{sh}_i\}_{i\in[n]}$ *is the randomized share algorithm. It takes as input a secret $s \in \mathcal{S}$. It outputs a set of shares $\{\mathsf{sh}_i\}_{i\in[n]}$.*
- $\mathsf{Proj}(\mathsf{sh}_i, k) \mapsto \mathsf{psh}_{i,k}$ *is the deterministic projection algorithm. It takes as input a share sh_i and a size $k \in [n]$. It outputs the projected share $\mathsf{psh}_{i,k}$ of level k.*
- $\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of projected shares of level k for an authorized subset A and the level k. It outputs the secret s.*

Apart from the perfect correctness and privacy requirements, these algorithms must satisfy the following requirements:

- ***Perfect projected correctness.*** *For any secret $s \in \mathcal{S}$ and any authorized set A with $|A| = k$, it holds that*

$$\mathbf{P}[s = \mathsf{Rec}(\{\mathsf{Proj}(\mathsf{Share}(s)_i, k)\}_{i\in A}, k)] = 1.$$

– ***Perfect share anonymity.*** *For any secret $s \in \mathcal{S}$, any unauthorized sets A, B with $|A| = |B| = k$, and any possible set of k shares $\{\mathsf{sh}_i\}_{i\in[k]}$, it holds that*

$$\mathbf{P}[\mathsf{Share}(s)_A = \{\mathsf{sh}_i\}_{i\in[k]}] = \mathbf{P}[\mathsf{Share}(s)_B = \{\mathsf{sh}_i\}_{i\in[k]}].$$

– ***Perfect projected anonymous reconstruction.*** *For any secret $s \in \mathcal{S}$, any authorized sets A, B with $|A| = |B| = k$, and any possible set of k projected shares $\{\mathsf{psh}_{i,k}\}_{i\in[k]}$ of level k, it holds that*

$$\begin{aligned}\mathbf{P}[\{\mathsf{Proj}(\mathsf{Share}(s)_i, k)\}_{i\in A} =& \{\mathsf{psh}_{i,k}\}_{i\in[k]}] \\ &= \mathbf{P}[\{\mathsf{Proj}(\mathsf{Share}(s)_i, k)\}_{i\in B} = \{\mathsf{psh}_{i,k}\}_{i\in[k]}].\end{aligned}$$

Remark 3.2. In the above definition, we adopt the weakest notion of share anonymity, namely *single-dealer anonymity*. This choice is motivated by the fact that, for constructing traceable secret sharing schemes, this weaker notion is sufficient.

Remark 3.3. As in the standard definition of secret sharing, the requirements of anonymous secret sharing can also be relaxed. In particular, instead of demanding perfect guarantees, one can define *statistical share anonymity* and *statistical projected anonymous reconstruction*. These relaxed variants lead to the definition of statistical anonymous secret sharing schemes.

Anonymous secret sharing schemes allow parties to reconstruct the secret without knowing, either before or after the reconstruction process, the identities of the other participating parties. As a consequence, a subset of parties may not know whether it is authorized or not. This can be problematic in certain applications, where it is important to prevent an unauthorized set from reconstructing a convincing but incorrect value. In such settings, parties may require a way to determine whether a given subset is actually authorized to reconstruct the secret. This requirement is captured by the following additional property, which we call *perfect status verification*.

Definition 3.4 (Perfect Status Verification).
There exists a deterministic algorithm $\mathsf{Verify}(\{\mathsf{sh}_i\}_{i\in A}) \mapsto \{0,1\}$ such that for any $s \in \mathcal{S}$, $B \in \Gamma$ and $C \notin \Gamma$ it holds that

$$\mathbf{Pr}[\mathsf{Verify}(\mathsf{Share}(s)_B) = 1] = \mathbf{Pr}[\mathsf{Verify}(\mathsf{Share}(s)_C) = 0] = 1.$$

Remark 3.5. We choose to present the notion of perfect status verification separately from the definition of anonymous secret sharing schemes, as we view it as an additional property that may be desirable in certain applications, rather than a core requirement of anonymous secret sharing itself. Moreover, we note that all anonymous schemes presented in this work trivially satisfy the notion of status verification in its statistical form.

3.2 Anonymous Schemes for Specific Access Structures

Next, we present simple constructions of anonymous secret sharing schemes for the n-out-of-n and threshold access structures. These constructions are obtained by making slight modifications to classical schemes, adapting them to satisfy the anonymity requirements introduced earlier.

Anonymous Additive Secret Sharing Schemes. We begin by observing that the standard additive secret sharing scheme for the n-out-of-n access structure already satisfies the requirements of an anonymous scheme. This is because the shares of any unauthorized subset are uniformly random and independent of the secret, making them indistinguishable from truly random values, ensuring share anonymity. Additionally, since there is only a single authorized subset, the projected anonymous reconstruction requirement is trivially satisfied by any scheme realizing this access structure. In this case, the projection algorithm is simply the identity function, as the entire share is always used during reconstruction, without the need for any further computation. The formal statement follows.

$\mathsf{Share}(s)$:

1. Set sh_i to be a uniformly random bit for $i \in [n]$, i.e.
$$\mathsf{sh}_1, ..., \mathsf{sh}_{n-1} \leftarrow_{\$} \{0,1\}^{n-1}.$$
2. Let sh_n be the subtraction of the random bits to the secret, i.e.
$$\mathsf{sh}_n \leftarrow s - \sum_{i=1}^{n-1} \mathsf{sh}_i.$$
3. Output $\{\mathsf{sh}_i\}_{i\in[n]}$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$:

1. Set $\mathsf{psh}_{i,k} \leftarrow \mathsf{sh}_i$, i.e. the projection algorithm is the identity function.
2. Output $\mathsf{psh}_{i,k}$.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k)$:

1. Set $s' \leftarrow \sum_{i\in[n]} \mathsf{psh}_{i,k}$.
2. Output s'.

Fig. 1: Anonymous additive secret sharing scheme.

Theorem 3.6. *Let Γ be the n-out-of-n threshold access structure over n parties. For any $s \in \{0,1\}$ the protocol of Fig. 1 is an anonymous secret sharing scheme realizing Γ with a share size of one bit.*

Proof. Since Projection is the identity function, the privacy and correctness properties follow directly from the uniform random selection of the first $n-1$ bits. For share anonymity, it suffices to observe that the shares of any unauthorized subset of size k are indistinguishable from k uniformly random bits by construction. Moreover, projected anonymous reconstruction is trivially satisfied, as there is only one authorized subset. Finally, it is clear that the share size is simply one bit. □

Remark 3.7. The above scheme satisfies the strongest notion of share anonymity, *uniform anonymity*, as shares of any unauthorized subset are indistinguishable from uniform values.

Remark 3.8. The above scheme can be extended to handle secrets and shares from any finite cyclic group, i.e. taking values from $\mathbb{Z}_m$ for any $m \in \mathbb{N}$.

Anonymous Threshold Secret Sharing Schemes. We now consider Shamir's scheme for threshold access structures. The main obstacle to satisfying our anonymity definition is that the evaluation points reveal the identity of the party holding each share. However, for reconstruction, only the set of evaluation points matters, not which party holds which one. Therefore, to anonymize the scheme we apply a uniformly random permutation to the evaluation points and assign to each party the share corresponding to their image under this permutation. As in the previous case, the projection algorithm is simply the identity function, since the entire share is used directly in any reconstruction process. This preserves correctness and privacy while breaking the link between parties and their shares. This is formalized in the following protocol.

Theorem 3.9 (Theorem 1.3 restated). *Let Γ be a threshold access structure over n parties. For any $s \in \mathbb{F}_q$ with $q \geq n$ the protocol of Fig. 2 is an anonymous secret sharing scheme realizing Γ with share size $O(\log q)$.*

Proof. Since Projection is the identity function, the privacy and correctness requirements hold because of the ones from the underlying Shamir scheme. Moreover, the requirements for share anonymity and projected anonymous reconstruction are satisfied due to the uniform random selection of the permutation σ and the interpolation theorem. Finally, since the secret and the shares are pairs of elements in $\mathbb{F}_q$, the share size is $2\log q = O(\log q)$. □

Remark 3.10. In this case, the scheme does not satisfy either of the stronger share anonymity notions, as each party receives a distinct evaluation point. Achieving them would require sampling evaluation points uniformly at random from a field of super-polynomial size, leading to statistical correctness. This approach is the one followed in the schemes of [BPR24,EBG+24].

Notation:

1. Let P_n denote the set of permutations of n elements.

$\mathsf{Share}(s)$:

1. Select a permutation $\sigma \in P_n$ uniformly at random i.e.

$$\sigma \leftarrow_{\$} P_n.$$

2. Secret share s using Shamir secret sharing scheme with threshold t, i.e.

$$\mathsf{sh}_1, ..., \mathsf{sh}_n \leftarrow \mathsf{Shamir}(s, n, t).$$

3. Set $j \leftarrow \sigma(i)$ and $\mathsf{sh}_i \leftarrow (j, p(j))$ for $i \in [n]$, where $p(x) \in \mathbb{F}_q[x]$ is the uniformly random polynomial constructed in Shamir scheme.
4. Output $\{\mathsf{sh}_i\}_{i\in[n]}$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$:

1. Set $\mathsf{psh}_{i,k} \leftarrow \mathsf{sh}_i$, i.e. the projection algorithm is the identity function.
2. Output $\mathsf{psh}_{i,k}$.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k)$:

1. Parse $\mathsf{psh}_{i,k}$ as (α_i, β_i) for $i \in A$.
2. Set $s' \leftarrow \sum_{i\in A} \left(\prod_{j\in A\setminus\{i\}} \frac{\alpha_j}{\alpha_j - \alpha_i}\right) \beta i$.
3. Output s'.

Fig. 2: Anonymous threshold secret sharing scheme.

3.3 Anonymous Scheme for General Access Structures

We now describe an anonymous secret sharing scheme for general access structures, based on the classical DNF construction. Our goal is to ensure that neither the shares nor the reconstruction procedure reveals the identities of the participating parties. To this end, we adopt a combinatorial approach that eliminates any reliance on the internal structure of the access structure, as such dependencies could leak information about which parties hold which shares.

As in the DNF construction, the secret is shared independently for each authorized subset. To do so, we employ the DNF additive scheme, which is also anonymous, as a building block. However, a direct application of this idea results in parties receiving shares only for the subsets they belong to. Since the number of such subsets varies across parties, this causes their share size to differ, which can leak information about their position in the access structure. To prevent this, we ensure that each party receives one share for every authorized subset (filling in with dummy random shares when necessary), which leads to a uniform share size across all the parties.

To further prevent parties from inferring the identity of other parties during reconstruction, we carefully define the Projection algorithm. In our construction, it simply selects the parts of the shares corresponding to authorized subsets of the same size as the subset currently attempting to reconstruct the secret. This restriction is essential to preserve anonymity: if a party were to reveal its full share, it could leak information about him. For example, if the reconstructing subset is not minimal, exposing the full shares could reveal which proper subset of it is minimal, thereby leaking information of the parties.

For the reconstruction process, we would like it to be as in the DNF scheme. However, the parties cannot directly identify which portion of their shares corresponds to their authorized subset, since the secret is unknown and the inclusion of dummy shares obscures the access structure. To address this, we incorporate an indicator technique. Alongside the actual secret, the dealer also distributes a known public indicator (specifically, the value 0) using the same sharing process. Parties can then test subsets of their projected indicator shares to identify the one that reconstructs the indicator 0, revealing the correct portion of their projected shares from which they must recover the secret. If the underlying field is large enough, the probability that an incorrect combination of projected indicator shares reconstructs the indicator is negligible, ensuring statistical correctness.

Finally, another potential source of leakage arises from the ordering of the authorized subsets during the sharing process. If the dealer selects subsets of a given size in a fixed order, parties might infer their position in the access structure by identifying the index of the share component they use to recover the secret. To prevent this, the dealer samples a fresh random permutation over the authorized subsets of each given size and distributes the shares according to this randomized order. This guarantees that the ordering of share components does not reveal any structural information.

The complete description of this anonymous scheme for general access structures is provided in Fig. 3.

Theorem 3.11 (Theorem 1.3 restated). *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(n)$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m$, the protocol of Fig. 3 is an anonymous secret sharing scheme with statistical reconstruction realizing Γ with share size $O(2^n \log m)$.*

Proof. Let $t = |\Gamma|$ be the total number of authorized subsets in the access structure, and let $t_k = |\Gamma_k|$ denote the number of authorized subsets of size k.

First, note that all shares are either generated using the anonymous additive scheme of Fig. 1 or sampled uniformly at random from $\mathbb{Z}_m$. Therefore, the privacy of the overall scheme directly follows from the privacy guarantees of the underlying additive scheme.

For share anonymity, it suffices to notice that the shares of any unauthorized subset of size k are indistinguishable from a set of $2tk$ uniformly random values from $\mathbb{Z}_m$.

The projected anonymous reconstruction property is also satisfied. Specifically, for any authorized subset of size k, the level-k projected shares consist

Notation:

1. Let Γ_k be the family of authorized subsets in Γ of size k.
2. Let P_k denote the set of permutations of $|\Gamma_k|$ elements.

$\mathsf{Share}(s)$:

1. For $k \in [n]$
   (a) Select a permutation $\sigma_k \in P_k$ uniformly at random, i.e.
   $$\sigma_k \leftarrow_{\$} P_k.$$
   (b) For $A \in \Gamma_k$
      i. Set $B \leftarrow \sigma_k(A)$.
      ii. Share $0, s \in \mathbb{Z}_m$ with the DNF additive scheme Σ among the parties in A, i.e.
      $$\{(\mathsf{sh}^0_{i,k,B}, \mathsf{sh}^s_{i,k,B})\}_{i\in A} \leftarrow (\Sigma(0, A), \Sigma(s, A)).$$
      iii. Set $\mathsf{sh}_{i,k,B} \leftarrow (\mathsf{sh}^0_{i,k,B}, \mathsf{sh}^s_{i,k,B})$ for $i \in A$.
      iv. Send two random elements $r_0, r_s \in \mathbb{Z}_m$ to each of the parties not in A, i.e.
      $$\mathsf{sh}_{i,k,B} \leftarrow_{\$} (\mathbb{Z}_m, \mathbb{Z}_m) \text{ for } i \in [n] \setminus A.$$
   (c) Set $\mathsf{sh}_{i,k} \leftarrow \{\mathsf{sh}_{i,k,B}\}_{B\in\mathcal{J}}$ for $i \in [n]$, where $\mathcal{J} = \{\sigma_k(A) : A \in \Gamma_k\}$.
2. Set $\mathsf{sh}_i \leftarrow \{\mathsf{sh}_{i,k}\}_{k\in[n]}$ for $i \in [n]$.
3. Output $\{\mathsf{sh}_i\}_{i\in[n]}$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$:

1. Parse sh_i as $\{\mathsf{sh}_{i,k}\}_{k\in[n]}$.
2. Set $\mathsf{psh}_{i,k} \leftarrow \mathsf{sh}_{i,k}$, i.e. the projection algorithm is the projection function.
3. Output $\mathsf{psh}_{i,k}$.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k)$:

1. Parse $\mathsf{psh}_{i,k}$ as $\{\mathsf{sh}_{i,k,B}\}_{B\in\Gamma_k}$ for $i \in A$.
2. Set $s' \leftarrow_{\$} \mathbb{Z}_m$.
3. For $B \in \Gamma_k$
   (a) Parse $\mathsf{sh}_{i,k,B}$ as $(\mathsf{sh}^0_i, \mathsf{sh}^s_i)$ for $i \in A$.
   (b) Set $u \leftarrow \sum_{i\in A} \mathsf{sh}^0_i$.
   (c) If $u = 0$, set $s' \leftarrow \sum_{i\in A} \mathsf{sh}^s_i$ and **break**.
4. Output s'.

Fig. 3: Anonymous secret sharing scheme for any access structure Γ.

of $2(t_k - 1)k$ uniformly random values from $\mathbb{Z}_m$, k shares from the anonymous

additive scheme that reconstruct to the indicator value 0, and k shares from the additive scheme that reconstruct to the actual secret.

We now analyze statistical correctness. By construction, for any authorized subset A of size k, we have:

$$0 = \sum_{i \in A} \mathsf{psh}^0_{i,k,B} \quad \text{and} \quad s = \sum_{i \in A} \mathsf{psh}^s_{i,k,B},$$

where $B = \sigma_k(A)$. Therefore, the reconstruction algorithm fails only if there exists another subset $C \in \Gamma_k$, with $C \neq B$, such that $\sum_{i \in A} \mathsf{psh}^0_{i,k,C} = 0$, mistakenly identifying C as the correct authorized subset. However, for each such incorrect subset, the shares are distributed as uniform random values from $\mathbb{Z}_m$, and since m is super-polynomial in the number of such subsets and in the security parameter λ, the probability of this occurring is negligible in λ.

Finally, regarding share size, observe that there are at most 2^n authorized subsets, and each party receives two shares of $\log m$ bits for each of these subsets. Therefore, the total share size per party is $O(2^n \log m)$. □

Remark 3.12. As in the additive case, the proof of Theorem 3.11 shows that the general anonymous scheme satisfies the strongest notion of share anonymity, *uniform anonymity*, since the shares of any unauthorized subset are indistinguishable from uniformly random values. Moreover, since Bishop et al. [BGI+25] proved an exponential lower bound on the share size for information-theoretic fully anonymous schemes, this construction is nearly optimal with respect to share size.

Remark 3.13. The general anonymous scheme can be optimized by combining it with the anonymous threshold scheme of Fig. 2. Specifically, if Γ is an access structure in which all unauthorized subsets have size at most $t-1$, then we can reduce the share size by applying the general anonymous scheme only to the authorized subsets of size at most $t-1$. For all authorized subsets of size at least t, it suffices to share the secret to all the parties using the anonymous threshold scheme of Fig. 2 with threshold t.

Remark 3.13 leads to efficient anonymous schemes for a specific family of access structures: low and high slices. That is, access structures over n parties given by t-slices and $(n-t)$-slices in which $t \ll n$.

Corollary 3.14. *Let $n \in \mathbb{N}$, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(n)$, let $q = \lambda^{\omega(1)}$ be a prime power, let $t \in [n]$ with $t = O(1)$, and let Γ be a t-slice (resp. an $(n-t)$-slice) access structure over n parties. For any $s \in \mathbb{F}_q$ there exists an anonymous secret sharing scheme with statistical reconstruction realizing Γ with share size $O(n^t \log q)$.*

Proof. By construction, an access structure over n parties given by a t-slice (resp. a $(n-t)$-slice) with $t = O(1)$ only has $O(n^t)$ authorized subsets with size strictly less than t (resp. $O(n^t)$ authorized subsets). Thus, it suffices to apply the scheme of Remark 3.13 (resp. of Theorem 3.11). □

Comparison with the General Scheme of [PCO20,BGI$^+$25]. First, we outline the ideas behind the anonymous secret sharing scheme presented in the works of [PCO20,BGI$^+$25]. Their scheme is a fully anonymous construction, satisfying uniform anonymity and anonymous reconstruction as defined in [BGI$^+$25], with statistical correctness and perfect privacy. The main idea is to share the secret independently with each minimal authorized subset using a vector-based approach. Specifically, for each such subset, the parties receive linearly dependent vectors over a finite field, and the secret bit is embedded in the linear dependency among these vectors. Parties not included in the minimal authorized subset instead receive linearly independent vectors, which do not contain any information about the secret. For a detailed description of this construction, we defer to their works [PCO20,BGI$^+$25]

Before comparing both constructions, it is important to highlight a structural difference: their scheme satisfies anonymous reconstruction, while our scheme satisfies the stronger notion of projected anonymous reconstruction. For this reason, in our scheme we share the secret not only with minimal authorized subsets, but also with every other authorized subset. In contrast, if we were to relax this requirement and adopt the standard notion of anonymous reconstruction from [BGI$^+$25], then sharing only with minimal authorized subsets would suffice, as we would not need to worry about non-minimal subsets deducing which of their subsets are minimal during reconstruction.

Conversely, their vector-based scheme could be extended to satisfy our stronger anonymity notion by also sharing the secret with every authorized subset, not just the minimal ones. Thus, the overall structure and intuition behind both constructions are similar: in both, the secret is shared independently across authorized subsets. The key difference lies in the techniques used: while their scheme relies on linear algebra and geometric properties of vector spaces, our approach is combinatorial and builds on additive secret sharing.

A notable advantage of their scheme is that it does not require an explicit indicator mechanism to identify which parts of the shares correspond to valid reconstructions. The linear dependence or independence of the vectors directly signals to the parties whether they form an authorized subset. In our construction, since we share group elements instead of vectors, there is no such relation and, as a result, we need to include a second sharing of a public indicator value to help parties identify the correct portion of their shares.

In terms of share size, both constructions are similar, as each scheme requires sharing the secret with a number of subsets proportional to the number of minimal authorized subsets and require super-polynomial large fields. However, our scheme offers a couple of practical advantages. First, it naturally supports secrets of arbitrary size, whereas the scheme of [PCO20,BGI$^+$25] is built for one-bit secrets and requires further adaptations to handle larger secrets. Second, the operations performed in our Share and Reconstruction algorithms are computationally simpler, involving only random sampling and addition over a finite cyclic group. These features also make our anonymous scheme particularly well suited

as a building block for our traceable construction for general access structures, which requires repeatedly executing sharing and reconstruction procedures.

3.4 Hiding Anonymous Secret Sharing

In this subsection, we first introduce the notion of hiding anonymous secret sharing schemes, which strengthens the notion of anonymity already presented. Then, we present such a construction for general access structures.

Definition. The definition of hiding anonymous schemes builds upon the notion anonymous schemes. In particular, apart from requiring that shares and reconstruction do not reveal the identities of the participating parties, we now consider a setting where the access structure itself is unknown to the parties. Instead, they are only aware that the access structure belongs to some known family of candidate access structures. In this context, the goal is to ensure that any set of parties learn no additional information about the specific access structure being used beyond what is inherently unavoidable.

More in detail, due to the correctness of scheme, a subset of parties will naturally learn which of their subsets are authorized and which are not, simply by testing whether reconstruction is possible. However, beyond this, a hiding anonymous scheme requires that the parties cannot distinguish which access structure, among all those compatible with their observed authorized and unauthorized subsets, was actually used to generate their shares.

To formalize this, we generalize the standard secret sharing model to consider schemes defined over a family of access structures, rather than a single one. In this framework, the Share algorithm receives as input the specific access structure from the family under which it must share the secret. In contrast, the $\mathsf{Projection}$ and $\mathsf{Reconstruction}$ are independent of the access structure, ensuring that reconstruction works uniformly across the entire family.

Finally, we require that, for every access structure in the family, the restriction of the scheme to that structure, that is, the Share algorithm instantiated with that specific access structure together with the common $\mathsf{Projection}$ and $\mathsf{Reconstruction}$ algorithms, must satisfy all the properties of an anonymous secret sharing scheme. The precise definition follows.

Definition 3.15 (Hiding Anonymous Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let $\mathcal{F}$ be a family of access structures over n parties. A hiding anonymous secret sharing scheme for $\mathcal{F}$ is a triple of the following algorithms:*

- $\mathsf{Share}(s, \Gamma) \mapsto \{\mathsf{sh}_i\}_{i\in[n]}$ *is the randomized share algorithm. It takes as input a secret $s \in \mathcal{S}$ and an access structure $\Gamma \in \mathcal{F}$. It outputs a set of shares $\{\mathsf{sh}_i\}_{i\in[n]}$.*
- $\mathsf{Proj}(\mathsf{sh}_i, k) \mapsto \mathsf{psh}_{i,k}$ *is the deterministic projection algorithm. It takes as input a share sh_i and a size $k \in [n]$. It outputs the projected share $\mathsf{psh}_{i,k}$ of level k.*

– $\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of projected shares of level* k *for an authorized subset* A *of some* $\Gamma \in \mathcal{F}$ *with* $|A| = k$ *and the level* k*. It outputs the secret* s*.*

These algorithms must satisfy the following requirements:

– ***Restricted Anonymous Schemes.*** *For any* $\Gamma \in \mathcal{F}$*, the restriction of the* Share *algorithm to* Γ *together with the* Projection *and* Reconstruction *algorithms form an anonymous secret sharing scheme for* Γ*.*
– ***Perfect Hideout.*** *For any* $s \in \mathcal{S}$*,* $k \in [n]$*, any subset* $A \subseteq [n]$ *such that* $|A| = k$*, any two access structures* $\Gamma, \Gamma' \in \mathcal{F}$ *such that* Γ *and* Γ' *agree in all subsets of* A*, and any possible set of* k *shares* $\{\mathsf{sh}_i\}_{i\in[k]}$*, it holds that*

$$\mathbf{P}[\mathsf{Share}(s, \Gamma)_A = \{\mathsf{sh}_i\}_{i\in[k]}] = \mathbf{P}[\mathsf{Share}(s, \Gamma')_A = \{\mathsf{sh}_i\}_{i\in[k]}].$$

Remark 3.16. As in Remark 3.3, the requirements of hiding anonymous schemes can also be relaxed. In particular, instead of demanding perfect guarantees, one can define *statistical restricted anonymous schemes* and *statistical hideout*. These relaxed variants lead to the definition of statistical hiding anonymous secret sharing schemes.

The Scheme. We now explain the idea behind the construction of hiding anonymous secret sharing schemes for a family of access structures. The main challenge in this setting is to prevent the parties from learning any information about the specific access structure used, beyond what is inherently revealed by the ability (or inability) to reconstruct the secret.

The anonymous scheme from Fig. 3 is already close to satisfying this notion. In particular, its Projection and Reconstruction algorithms are independent of the access structure, and thus already compatible with the hiding requirement. The only obstacle preventing the scheme from being hiding is that the total share size of each party depends on the number of authorized subsets in the selected access structure. Therefore, if the family contains access structures with varying numbers of authorized subsets, the share size itself would leak information about which access structure was used.

To address this issue, we modify the scheme as follows. Instead of generating shares only for the authorized subsets of the selected access structure, we consider the set of all authorized subsets that appear in any access structure in the family. For those subsets that belong to the selected access structure, we apply the same procedure as in Fig. 3, using the anonymous additive scheme to share both the secret and the indicator. For all other subsets, we simply assign dummy shares to the parties. This padding ensures that the share size is uniform across all access structures in the family and thus does not leak any structural information. The rest of the scheme remains unchanged, and the correctness, privacy, and anonymity guarantees follow from the ones in the previous construction.

A full specification of the hiding anonymous scheme for a family of access structures is presented in Fig. 4.

Notation:

1. Let $\Delta = \bigcup_{\Gamma' \in \mathcal{F}} \Gamma'$ be the family of all authorized subsets in $\mathcal{F}$.
2. Let Δ_k be the family of subsets in Δ of size k.
3. Let P_k denote the set of permutations of $|\Delta_k|$ elements.

$\mathsf{Share}(s, \Gamma)$:

1. For $k \in [n]$
   (a) Select a permutation $\sigma_k \in P_k$ uniformly at random, i.e.
   $$\sigma_k \leftarrow_{\$} P_k.$$
   (b) For $A \in \Delta_k$
      i. Set $B \leftarrow \sigma_k(A)$.
      ii. If $A \in \Gamma$
         A. Share $0, s \in \mathbb{Z}_m$ with the DNF additive scheme Σ among the parties in A, i.e.
         $$\{(\mathsf{sh}^0_{i,k,B}, \mathsf{sh}^s_{i,k,B})\}_{i \in A} \leftarrow (\Sigma(0, A), \Sigma(s, A)).$$
         B. Set $\mathsf{sh}_{i,k,B} \leftarrow (\mathsf{sh}^0_{i,k,B}, \mathsf{sh}^s_{i,k,B})$ for $i \in A$.
         C. Send two random elements $r_0, r_s \in \mathbb{Z}_m$ to each of the parties not in A, i.e.
         $$\mathsf{sh}_{i,k,B} \leftarrow_{\$} (\mathbb{Z}_m, \mathbb{Z}_m) \text{ for } i \in [n] \setminus A.$$
      iii. If $A \notin \Gamma$
         A. Send two random elements $r_0, r_s \in \mathbb{Z}_m$ to each of the n parties, i.e.
         $$\mathsf{sh}_{i,k,B} \leftarrow_{\$} (\mathbb{Z}_m, \mathbb{Z}_m) \text{ for } i \in [n].$$
   (c) Set $\mathsf{sh}_{i,k} \leftarrow \{\mathsf{sh}_{i,k,B}\}_{B \in \mathcal{J}}$ for $i \in [n]$, where $\mathcal{J} = \{\sigma_k(A) : A \in \Delta_k\}$.
2. Set $\mathsf{sh}_i = \{\mathsf{sh}^k_i\}_{k \in [n]}$ for $i \in [n]$.
3. Output $\{\mathsf{sh}_i\}_{i \in [n]}$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$:

1. The same as in the general anonymous scheme of Fig. 3.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i \in A}, k)$:

1. Set $t \leftarrow \binom{n}{k}$.
2. Parse $\mathsf{psh}_{i,k}$ as $\{\mathsf{sh}_{i,k,B}\}_{B \in \Delta_k}$ for $i \in A$.
3. Set $s' \leftarrow_{\$} \mathbb{Z}_m$.
4. For $B \in \Delta_k$
   (a) Parse $\mathsf{sh}^k_{i,j}$ as $(\mathsf{sh}^0_i, \mathsf{sh}^s_i)$ for $i \in A$.
   (b) Set $u \leftarrow \sum_{i \in A} \mathsf{sh}^0_i$.
   (c) If $u = 0$, set $s' \leftarrow \sum_{i \in A} \mathsf{sh}^s_i$ and **break**.
5. Output s'.

Fig. 4: Hiding anonymous secret sharing scheme for any access structure Γ of a family of access structures $\mathcal{F}$.

Theorem 3.17 (Theorem 1.4 restated). *Let $\mathcal{F}$ be a family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(n)$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m$, the protocol of Fig. 4 is a hiding anonymous secret sharing scheme for $\mathcal{F}$ with statistical reconstruction and share size $O(2^n \log m)$.*

Proof. First, observe that for any $\Gamma \in \mathcal{F}$, the Share algorithm shares the secret in the same way as the anonymous scheme from Fig. 3, and both the Projection and Reconstruction algorithms are identical to those in that scheme. Therefore, the restricted anonymous scheme condition follows directly from the proof of Theorem 3.11.

We now establish the hiding property. Let $A \subseteq [n]$ be an arbitrary subset of parties, and let $\Delta = \bigcup_{\Gamma \in \mathcal{F}} \Gamma$ be the family of all authorized subsets of all access structures in $\mathcal{F}$. Since all parties receive shares of equal length, and the shares of any unauthorized subset are indistinguishable from uniformly random values (due to the privacy proof of Theorem 3.11), it suffices to restrict our analysis to the case in which $A \in \Delta$.

With this in mind, we analyze the distribution of the portions of the shares held by the parties in $A \in \Delta$ when shared according to two access structures $\Gamma, \Gamma' \in \mathcal{F}$ compatible with A, that is, Γ and Γ' agree in all subsets of A.

For any subset $B \subseteq A$ such that $B \in \Gamma \cap \Gamma'$, the corresponding parts of the shares are generated identically in both schemes, i.e. using the same anonymous additive secret sharing scheme. Hence, their distributions are the same.

For any subset $B \subseteq A$ such that $B \notin \Gamma \cap \Gamma'$, the shares held by the parties in A are indistinguishable from uniformly random values due to the privacy property of the scheme. Thus, their distributions are the same. This establishes the hiding property.

Finally, since there are at most 2^n authorized subsets and each party receives two shares of $\log m$ bits per subset, the total share size is $O(2^n \log m)$. □

Remark 3.18. As in Remark 3.13, the scheme described above can be optimized by combining it with the anonymous threshold scheme of Fig. 2. Specifically, if $\mathcal{F}$ is a family of access structures in which all access structures satisfy that all their unauthorized subsets have size at most $t-1$, then we can reduce the share size by applying the general anonymous scheme only to the authorized subsets of size at most $t-1$. For all authorized subsets of size at least t, it suffices to share the secret to all the parties using the anonymous threshold scheme of Fig. 2 with threshold t.

Remark 3.18 leads to efficient hiding anonymous schemes for a specific family of access structures: low and high slices. That is, access structures over n parties given by t-slices and $(n-t)$-slices in which $t \ll n$.

Corollary 3.19. *Let $n \in \mathbb{N}$, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(n)$, let $q = \lambda^{\omega(1)}$ be a prime power, let $t \in [n] = O(1)$ and let $\mathcal{F}$ be the family of t-slice (resp. an $(n-t)$-slice) access structures over n parties. For any $s \in \mathbb{F}_q$ there exists a hiding anonymous secret sharing scheme with statistical reconstruction realizing $\mathcal{F}$ with share size $O(n^t \log q)$.*

Proof. By construction, any access structure over n parties given by a t-slice (resp. a $(n-t)$-slice) with $t = O(1)$ only has $O(n^t)$ authorized subsets with size strictly less than t (resp. $O(n^t)$ authorized subsets). Thus, it suffices to apply the scheme of Remark 3.18 (resp. of Theorem 3.17). □

4 Traceable Secret Sharing for General Access Structures

In this section, we focus on generalizing the definitions of traceable secret sharing from [BPR24] to accommodate arbitrary access structures. First, we outline how we extend their tracing model to the general case. Later, we present a restricted notion of traceability, which we refer to as *projection traceability*. We then build upon this notion to define traceability for general access structures.

4.1 The Tracing Model

Our tracing framework builds on the model of [BPR24], which formalizes traceability in the context of threshold secret sharing, and extends it to support arbitrary monotone access structures. In their setting, the adversary uses the shares of an unauthorized subset of parties to construct a reconstruction box, which can later reconstruct the secret when provided with a sufficient number of valid shares. It is assumed that this reconstruction box is eventually obtained by the authorities, who are able to interact with it in a black-box manner in order to trace it back to the corrupted parties who built it from their shares.

Within this framework, a traceable secret sharing scheme must satisfy two main requirements: *traceability*, which ensures that the tracer can correctly identify at least one corrupted party whose share was used to build the reconstruction box; and *non-imputability*, which guarantees that no honest party is falsely blamed. To support the tracing procedure, the Share algorithm outputs not only the individual shares, but also a tracing key and a verification key. The tracing key is given to the tracer and is used to generate a proof that links the reconstruction box to one or more corrupted parties. The verification key is given to a verifier and enables to check the correctness of the proof presented by the tracer.

We generalize this model in two important directions. First, we incorporate the Projection algorithm already defined for anonymous schemes and introduce an additional randomized algorithm, denoted ShareGeneration. This algorithm models a local computation based on the tracing key that allows the tracer to generate projected *pseudoshares*, that is, artificial projected shares that simulate valid projected shares. These projected pseudoshares are used to interact with the reconstruction box during the tracing procedure while maintaining anonymity. This process is described in more detail later.

Second, we extend the notion of reconstruction boxes beyond the threshold setting. Rather than assuming that the reconstruction box operates on a fixed number of shares, as is the threshold case, we allow it to accept any subset of shares that satisfies a general access structure. As a result, the tracing procedure must be designed to accommodate this more flexible and expressive behavior.

In this context, together with the extended notion of traceability, we also introduce a restricted variant that we call *projection traceability*. At a high level, the two notions differ in the amount of information that parties must provide to the reconstruction box built by the adversary. Specifically, in the projected traceability setting, it is assumed that the reconstruction box only requires the projected shares of the parties to recover the secret. In contrast, in the traceability setting, the reconstruction box is allowed to reconstruct the secret only when it receives the full shares. As we will discuss in Section 6, assuming that the reconstruction box receives only projected shares simplifies the tracing procedure.

4.2 Projection Traceable Secret Sharing

We begin by introducing the definition of projection traceable secret sharing schemes for general access structures.

Definition 4.1 (Projection Traceable Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let Γ be an access structure over n parties. A projection traceable secret sharing scheme for Γ is a sextuple of the following algorithms:*

- $\mathsf{Share}(s, \Gamma) \mapsto (\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$ *is the sharing algorithm. It takes as input parameter the secret and the access structure.*[7] *It outputs a set of shares $\{\mathsf{sh}_i\}_{i\in[n]}$, a tracing key tk and a verification key vk.*
- $\mathsf{Proj}(\mathsf{sh}_i, k) \mapsto \mathsf{psh}_{i,k}$ *is the deterministic projection algorithm. It takes as input a share sh_i and a size $k \in [n]$. It outputs the projected share $\mathsf{psh}_{i,k}$ of level k.*
- $\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of projected shares $\mathsf{psh}_{i,k}$ of level k for an authorized set A with $|A| = k$ and the level k. It outputs the secret s.*
- $\mathsf{ShGen}(\mathsf{tk}, k, k', p) \mapsto \{\mathsf{psh}'_{i,k+k'}\}_{i\in[k']}$ *is the randomized share generation algorithm. It takes as input a traceable key $\mathsf{tk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$, two sizes $k, k' \in [n]$ such that $k + k' \leq n$, and an index p of one of the identifier's bits. It outputs a set of k' projected pseudoshares $\{\mathsf{psh}'_{i,k+k'}\}_{i\in[k']}$ of level $k + k'$.*
- $\mathsf{Trace}^R(\mathsf{tk}) \mapsto (\mathcal{I}, \pi)$ *is the randomized tracing algorithm. It takes as input a tracing key $\mathsf{tk} = (\mathsf{tk}_1, ..., \mathsf{tk}_n)$ and it gets oracle (black-box) access to a reconstruction box R. It outputs a subset $\mathcal{I} \subset [n]$ of identities of corrupted parties and an associated proof π.*
- $\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi) \mapsto \{0, 1\}$ *is the deterministic verification algorithm. It takes as input a verification key $\mathsf{vk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$, an alleged traitor subset $\mathcal{I}$, and an associated proof π. It outputs 1 (resp. 0) when accepting (resp. rejecting) the proof π that the parties $\mathcal{I}$ are guilty.*

[7] The requirement that the Share algorithm receives the access structure as input is not strictly necessary. However, we include it to maintain consistency with the definition of traceable secret sharing schemes presented later in Definition 4.8, where the access structure must be explicitly provided as input.

Apart from the standard notions of correctness and privacy, these algorithms must also satisfy the following requirements:

- ***Perfect Tracer Privacy.*** *For any secrets $s, s' \in \mathcal{S}$, any unauthorized set A, and any possible set of shares $\{\mathsf{sh}_i\}_{i\in A}$ and any possible tracing key tk, it holds that*

$$\mathbf{P}[(\{\mathsf{sh}_i\}_{i\in A}, \mathsf{tk}) = \mathsf{Share}(s, \Gamma)_{A,\mathsf{tk}}] = \mathbf{P}[(\{\mathsf{sh}_i\}_{i\in A}, \mathsf{tk}) = \mathsf{Share}(s', \Gamma)_{A,\mathsf{tk}}],$$

  where $\mathsf{Share}(s)_{A,\mathsf{tk}}$ corresponds to the restriction of the Share algorithm to the shares of A and the whole tracing key.
- ***Perfect Projection Traceability.*** *See Definition 4.4.*
- ***Perfect Projection Non-imputability.*** *See Definition 4.5.*

To define the notions of projection traceability and non-imputability, we first introduce the notion of projection good reconstruction boxes. Intuitively, a reconstruction box is considered projection good if, when provided with a sufficient number of valid projected shares, it reconstructs the secret with high probability. As previously discussed, we generalize the definition of [BPR24] to support general access structures and allow the reconstruction box to output only the minimal useful information, namely a single bit, rather than the full secret. In addition, since our framework incorporates the $\mathsf{Projection}$ algorithm, we further adapt the definition to require that the reconstruction box succeeds when it is given only the projected shares instead of the full shares.

Another important difference concerns how the correlation between the shares given to the corrupted parties and the shares used as input to the reconstruction box is modeled. In [BPR24], this correlation is made explicit through a correlation string, which is used by the sharing algorithm to ensure consistency between the two sets of shares. In our case, we instead require that the shares assigned to the corrupted parties coincide with the ones produced by the sharing procedure that generates the shares used as input to the reconstruction box. While the correlation-string mechanism could be adapted to our current setting, it does not generalize to the more general traceability model introduced later in Section 4.3. In contrast, our approach extends naturally to that broader framework. This distinction and its justification are further elaborated in Section 4.3.

Definition 4.2 (Projection Good Reconstruction Box). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, let Γ be an access structure over n parties, and let Σ be a projection traceable secret sharing for Γ. For $\epsilon \in [0, \frac{1}{2}]$, secrets $s_0, s_1 \in \mathcal{S}$, a bit $b \in \{0, 1\}$, an unauthorized subset $A \notin \Gamma$ with $k = |A|$, and a corresponding set of shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in A}$, a reconstruction box R is $(\Gamma, A, \mathbf{sh}, s_0, s_1, \epsilon)$-projection good if for any subset $B \subseteq [n]$ with $k' = |B|$ such that $A \cap B = \emptyset$ and $A \cup B \in \Gamma$, it holds that*

$$\mathbf{P}[R(\{\mathsf{psh}'_{i,k+k'}\}_{i\in B} = b)] \geq \frac{1}{2} + \epsilon,$$

where the probability is taken over $(\{\mathsf{sh}'_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$ conditioned on $\{\mathsf{sh}'_i\}_{i\in A} = \mathbf{sh}$, $\mathsf{psh}'_{i,k+k'} \leftarrow \mathsf{Proj}(\mathsf{sh}'_i, k + k')$ for $i \in B$, and the random coins of R.

Remark 4.3. Note that in the above definition, we require that $A \cup B$ is an authorized subset, rather than a minimal authorized subset, as done in the definition of [BPR24]. This change is necessary to accommodate general access structures, for which the minimality condition may be too restrictive. In particular, there are access structures in which there exists an unauthorized subset A that cannot be extended to a minimal authorized subset by adding any disjoint subset B. For example, consider a graph-based access structure, where the minimal authorized subsets correspond to pairs of vertices connected by an edge. If A is a pair of vertices not connected by an edge, then for any disjoint subset B, the union $A \cup B$ cannot form a minimal authorized subset.

Once the notion of projection good reconstruction boxes is in place, we proceed to formalize the definitions of projection traceability and projection non-imputability. We begin by presenting the projection traceable experiment in Fig. 5, followed by the formal definition of the projection traceability requirement in Definition 4.4.

Experiment $\mathbf{ExpProjTrace}_{\mathcal{A},\Sigma,\epsilon}$

1. $(\Gamma, J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $b \leftarrow_{\$} \{0,1\}$.
3. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$.
4. $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$.
5. $R \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
6. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
7. $\mathsf{GoodBox} \coloneqq (J \notin \Gamma) \wedge (R$ is $(\mathcal{F}, J, \mathbf{sh}, s_0, s_1, b, \epsilon)$-projection-good.
8. If $(J^* \not\subseteq J) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 1.
9. If $\neg\mathsf{GoodBox}$, then return 0.
10. If $(J^* \neq \emptyset) \wedge (J^* \subseteq J) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 0, else return 1.

Fig. 5: The projection tracing experiment for a traceable secret sharing scheme Σ for an access structure Γ.

Definition 4.4 (Perfect Projection Traceability). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let Σ be a projection traceable secret sharing for an access structure Γ over n parties. For any $\epsilon \in [0, \frac{1}{2}]$, Σ is perfect projection traceable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{proj-trac}}_{\mathcal{A},\Sigma,\epsilon} \coloneqq \left| \epsilon \cdot \mathbf{P}[\mathbf{ExpProjTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] - \frac{1}{|\mathcal{S}|} \right| = 0.$$

We now describe the projection non-imputability experiment in Fig. 6, and then formally define the projection non-imputability requirement in Definition 4.5.

Experiment $\mathbf{ExpProjNI}_{\mathcal{A},\Sigma}$

1. $(\Gamma, i^*, s, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$.
3. $(J^*, \pi) \leftarrow \mathcal{A}(\mathsf{state}, \{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}, \mathsf{tk})$.
4. If $(i^* \in J^*) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 1, else return 0.

Fig. 6: The non-imputability experiment for a projection traceable secret sharing scheme Σ for an access structure Γ.

Definition 4.5 (Perfect Projection Non-imputability). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let Σ be a projection traceable secret sharing for an access structure Γ over n parties. Then, Σ is projection non-imputable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{proj-ni}}_{\mathcal{A},\Sigma} := \mathbf{P}[\mathbf{ExpProjNI}_{\mathcal{A},\Sigma} = 1] = 0.$$

Remark 4.6. In contrast to the non-imputability definition of [BPR24], our definition does not grant the adversary access to the verification key, as doing so would allow it to falsely accuse an honest party in our construction. This restriction is inherent to the information-theoretic setting: if the verification key were public, non-imputability would fail in a trivial way. Indeed, the adversary could simply brute-force candidate proofs and test them against the verification key until a valid one is found. Such a proof is guaranteed to exist, since any single party is unauthorized and therefore traceable. In Section 5.4, we show that by relying on computational assumptions, this limitation can be overcome, and we explain how to modify our scheme to support a fully public verification key, thereby achieving public verifiability.

Furthermore, as in Remark 3.3, the requirements of projection traceable schemes can also be relaxed. In particular, instead of demanding perfect guarantees, one can define *statistical tracer privacy*, *statistical projection traceability* and *statistical projection non-imputability*. These relaxed variants lead to the definition of statistical projection traceable secret sharing schemes.

We end this subsection by introducing an additional requirement for projection traceable secret sharing schemes: *projection indistinguishability*. This notion strengthens the security guarantees of the scheme by ensuring that projected pseudoshares generated during the tracing procedure are indistinguishable from actual projected shares of honest parties.

More specifically, consider any two disjoint subsets $A, B \subseteq [n]$ such that $A \notin \Gamma$, and $A \cup B \in \Gamma$. Projection indistinguishability requires that, when the secret is sampled uniformly at random, the joint distribution of the shares of the parties in A, and the projections of the shares of the parties in B at level $|A \cup B|$ is equal to the joint distribution of the shares of the parties in A, and $|B|$

projected pseudoshares at level $|A \cup B|$ generated using the tracing key output by the Share algorithm.

Intuitively, this notion captures the idea that a projection good reconstruction box built from the shares of an unauthorized subset cannot tell whether it is being queried with actual projected shares or with projected pseudoshares. This indistinguishability plays a central role in our tracing experiment, as it ensures that the behavior of the reconstruction box remains consistent regardless of whether real or simulated projected shares are used, which enables the tracer to extract identifying information from the box in a sound and private manner. The formal definition follows.

Definition 4.7 (Perfect Projection Indistinguishability). *Let $m > 1$ be a power of two, let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| = m^2$, let Γ be an access structure over n parties, and let Σ be a projection traceable secret sharing for Γ. Then, Σ is projection indistinguishable if for any secret $s \in \mathcal{S}$, any $p \in [2 \log m]$, and any pair of disjoint subsets $A, B \subseteq [n]$ such that $A, B \notin \Gamma$, and $A \cup B \in \Gamma$, it holds that the distributions*

$$(\{\mathsf{sh}_i\}_{i\in A}, \{\mathsf{psh}_{i,k+k'}\}_{i\in B}) \text{ and } (\{\mathsf{sh}_i\}_{i\in A}, \{\mathsf{psh}'_{i,k+k'}\}_{i\in [k']})$$

are equal, where $k = |A|$, $k' = |B|$, $s \leftarrow_{\$} \mathcal{S}$, $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$, $\mathsf{psh}_{i,k+k'} \leftarrow \mathsf{Proj}(\mathsf{sh}_i, k+k')$ for $i \in B$, and $\{\mathsf{psh}'_{i,k+k'}\}_{i\in[k']} \leftarrow \mathsf{ShGen}(\mathsf{tk}, k, k', p)$.

4.3 Traceable Secret Sharing

The notion of projection traceability introduced in the previous subsection provides a useful stepping stone towards defining traceability in the general access structure setting. However, its applicability is inherently limited due to its reliance on a restricted model of reconstruction boxes. Specifically, it assumes that the reconstruction box only requires parties to input their projected shares to recover the secret. As discussed later in Section 6.1, this restriction makes it harder for the reconstruction box to distinguish fake queries issued by the tracer, since projected shares carry less information than full shares. As a result, while this setting simplifies the tracing procedure, it does not fully capture realistic scenarios. In particular, an adversary aware of this limitation could simply build a reconstruction box that expects full shares instead.

In this context, we introduce the notion of traceability for general access structures, which extends the projection traceability model by considering reconstruction boxes that operate on the full shares held by the parties, rather than just their projections. However, as we will discuss in Section 6, enabling tracing in this setting requires more than anonymity; the scheme must also hide the access structure itself. Consequently, since we will work with hiding anonymous schemes, it is convenient to define traceable schemes with respect to a family of access structures rather than a single one. The definition follows.

Definition 4.8 (Traceable Secret Sharing Scheme). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, and let $\mathcal{F}$ be a family of access structures over n parties. A traceable secret sharing scheme for $\mathcal{F}$ is a sextuple of the following algorithms:*

- $\mathsf{Share}(s, \Gamma) \mapsto (\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$ *is the sharing algorithm. It takes as input parameter the secret and the access structure. It outputs a set of shares* $\{\mathsf{sh}_i\}_{i\in[n]}$*, a tracing key* tk *and a verification key* vk*.*
- $\mathsf{Proj}(\mathsf{sh}_i, k) \mapsto \mathsf{psh}_{i,k}$ *is the deterministic projection algorithm. It takes as input a share* sh_i *and a size* $k \in [n]$*. It outputs the projected share* $\mathsf{psh}_{i,k}$ *of level* k*.*
- $\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of projected shares* $\mathsf{psh}_{i,k}$ *of level* k *for an authorized set* A *with* $|A| = k$ *and the level* k*. It outputs the secret* s*.*
- $\mathsf{ShGen}(\mathsf{tk}, k, k', p) \mapsto \{\mathsf{sh}'_i\}_{i\in[k']}$ *is the randomized share generation algorithm. It takes as input a traceable key* $\mathsf{tk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$*, two sizes* $k, k' \in [n]$ *such that* $k + k' \leq n$*, and an index* p *of one of the identifier's bits. It outputs a set of* k' *pseudoshares* $\{\mathsf{sh}'_i\}_{i\in[k']}$*.*[8]
- $\mathsf{Trace}^R(\mathsf{tk}) \mapsto (\mathcal{I}, \pi)$ *is the randomized tracing algorithm. It takes as input a tracing key* $\mathsf{tk} = (\mathsf{tk}_1, ..., \mathsf{tk}_n)$ *and it gets oracle (black-box) access to a reconstruction box* R*. It outputs a subset* $\mathcal{I} \subset [n]$ *of identities of corrupted parties and an associated proof* π*.*
- $\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi) \mapsto \{0, 1\}$ *is the deterministic verification algorithm. It takes as input a verification key* $\mathsf{vk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$*, an alleged traitor subset* $\mathcal{I}$*, and an associated proof* π*. It outputs* 1 *(resp.* 0*) when accepting (resp. rejecting) the proof* π *that the parties* $\mathcal{I}$ *are guilty.*

Apart from the standard notions of correctness and privacy, these algorithms must also satisfy the following requirements:

- ***Perfect Tracer Privacy.*** *For any secrets* $s, s' \in \mathcal{S}$*, any unauthorized set* A*, and any possible set of shares* $\{\mathsf{sh}_i\}_{i\in A}$ *and any possible tracing key* tk*, it holds that*

$$\mathbf{P}[(\{\mathsf{sh}_i\}_{i\in A}, \mathsf{tk}) = \mathsf{Share}(s, \Gamma)_{A,\mathsf{tk}}] = \mathbf{P}[(\{\mathsf{sh}_i\}_{i\in A}, \mathsf{tk}) = \mathsf{Share}(s', \Gamma)_{A,\mathsf{tk}}],$$

  where $\mathsf{Share}(s, \Gamma)_{A,\mathsf{tk}}$ *corresponds to the restriction of the* Share *algorithm to the shares of* A *and the whole tracing key.*
- ***Perfect Traceability.*** *See Definition 4.10.*
- ***Perfect Non-imputability.*** *See Definition 4.11.*

As in Section 4.2, to define the notions of traceability and non-imputability, we begin by introducing the notion of good reconstruction boxes. The key difference from the definition of projection good reconstruction boxes is that, in this setting, the reconstruction box is allowed to receive the full shares of the parties as input, rather than only their projected shares.

Additionally, since our traceable schemes are defined over a family of access structures, the definition of good reconstruction boxes must reflect this generality. In particular, we require the reconstruction box to be correct for all access

[8] In this setting, the $\mathsf{ShareGeneration}$ algorithm outputs full pseudoshares instead of projected pseudoshares because the reconstruction box takes as input the whole shares.

structures in the family that are compatible with the corrupted subset, that is, for all access structures in which the corrupted parties form an unauthorized set. For each of these access structures, the reconstruction box must successfully reconstruct the secret when it is given full shares generated under that access structure, as long as the shares corresponding to the corrupted parties coincide with those from the original sharing. This condition is justified by the fact that the reconstruction algorithm is independent of the access structure, which remains hidden from the parties.

This also motivates our modeling choice of requiring equality between the shares of the corrupted parties, instead of introducing a correlation string as done in [BPR24]. In our setting, the secret is shared under two potentially different access structures (though they agree on the behavior of the corrupted set), making it infeasible to use the same correlation string for both cases. Instead, requiring that the shares of the corrupted parties match in both sharing procedures provides a natural way to ensure consistency among the shares.

Definition 4.9 (Good Reconstruction Box). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, let $\mathcal{F}$ be a family of access structures over n parties, let Γ be an access structure in $\mathcal{F}$, and let Σ be a traceable secret sharing for $\mathcal{F}$. For $\epsilon \in [0, \frac{1}{2}]$, secrets $s_0, s_1 \in \mathcal{S}$, a bit $b \in \{0,1\}$, an unauthorized subset $A \notin \Gamma$, and a corresponding set of shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in A}$, a reconstruction box R is $(\mathcal{F}, A, \mathbf{sh}, s_0, s_1, b, \epsilon)$-good if for any access structure $\Gamma' \in \mathcal{F}$ that is consistent with A, any unauthorized subset $B \notin \Gamma'$ such that $A \cap B = \emptyset$ and $A \cup B \in \Gamma'$, it holds that*

$$\mathbf{P}[R(\{\mathsf{sh}'_i\}_{i\in B}) = b] \geq \frac{1}{2} + \epsilon,$$

where the probability is taken over $(\{\mathsf{sh}'_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma')$ conditioned on $\{\mathsf{sh}'_i\}_{i\in A} = \mathbf{sh}$, and the random coins of R.

Once the notion of good reconstruction boxes is established, we proceed to formalize the definitions of traceability and non-imputability. Broadly speaking, these definitions follow the same structure as those introduced in the projection traceability setting. The main difference is that, in this case, the adversary does not choose the access structure, as it is hidden from the parties. Instead, the adversary only knows the family of access structures for which the traceable scheme is defined. To highlight the differences introduced in this setting, all modifications are marked in blue.

We start by presenting the traceable experiment in Fig. 7, followed by the formal definition of the traceability requirement in Definition 4.10.

Definition 4.10 (Perfect Traceability). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, let $\mathcal{F}$ be a family of access structures over n parties, and let Σ be a traceable scheme for $\mathcal{F}$. For any $\epsilon \in [0, \frac{1}{2}]$, Σ is perfect traceable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{trac}}_{\mathcal{A},\Sigma,\epsilon} \coloneqq \mathbf{P}[\mathbf{ExpTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = 0.$$

Experiment $\mathbf{ExpTrace}_{\mathcal{A},\Sigma,\epsilon}$

1. $(J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\mathcal{F})$.
2. $\Gamma \leftarrow_{\$} \mathcal{F} \setminus \bigcup_{\substack{\Gamma' \in \mathcal{F} \\ J \in \Gamma'}} \Gamma'$. ▷ The subset J must be unauthorized in Γ.
3. $b \leftarrow_{\$} \{0,1\}$.
4. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$.
5. $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$.
6. $R \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
7. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
8. $\mathsf{GoodBox} := (J \notin \Gamma) \wedge (R \text{ is } (\mathcal{F}, J, \mathbf{sh}, s_0, s_1, b, \epsilon)\text{-good})$.
9. If $(J^* \not\subseteq J) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 1.
10. If $\neg\mathsf{GoodBox}$, then return 0.
11. If $(J^* \neq \emptyset) \wedge (J^* \subseteq J) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 0, else return 1.

Fig. 7: The tracing experiment for a traceable secret sharing scheme Σ for a family of access structures $\mathcal{F}$. The steps highlighted in blue indicate the differences compared to the tracing experiment of Fig. 5.

Experiment $\mathbf{ExpNI}_{\mathcal{A},\Sigma}$

1. $(\Gamma, i^*, s, \mathsf{state}) \leftarrow \mathcal{A}(\mathcal{F})$.
2. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$.
3. $(J^*, \pi) \leftarrow \mathcal{A}(\mathsf{state}, \{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}, \mathsf{tk})$.
4. If $(i^* \in J^*) \wedge (\mathsf{Verify}(\mathsf{vk}, J^*, \pi) = 1)$ then return 1, else return 0.

Fig. 8: The non-imputability experiment for a traceable secret sharing scheme Σ for a family of access structures $\mathcal{F}$. The steps highlighted in blue indicate the differences compared to the non-imputability experiment of Fig. 6.

We now describe the non-imputability experiment in Fig. 8, and then formally define the non-imputability requirement in Definition 4.11.

Definition 4.11 (Perfect Non-imputability). *Let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| \geq 2$, let $\mathcal{F}$ be a family of access structures over n parties, and let Σ be a traceable scheme for $\mathcal{F}$. Then, Σ is non-imputable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{ni}}_{\mathcal{A},\Sigma} := \mathbf{P}[\mathbf{ExpNI}_{\mathcal{A},\Sigma} = 1] = 0.$$

Remark 4.12. As in Remark 4.6, the requirements of traceable schemes can also be relaxed. In particular, instead of demanding perfect guarantees, one can define *statistical tracer privacy*, *statistical traceability* and *statistical non-imputability*.

These relaxed variants lead to the definition of statistical traceable secret sharing schemes.

We conclude this subsection by introducing an additional requirement for traceable secret sharing schemes: *minimal set indistinguishability*. This notion generalizes *projection indistinguishability* and further strengthens the security guarantees of the scheme. It ensures that the pseudoshares generated during the tracing procedure are indistinguishable from real shares held by honest parties, where the union of these parties and the corrupted subset forms a minimal authorized subset in some access structure within the same family.

More precisely, consider two disjoint subsets $A, B \subseteq [n]$ such that $A, B \notin \Gamma \cup \Gamma'$ for some $\Gamma, \Gamma' \in \mathcal{F}$, and $A \cup B$ is minimal authorized subset in Γ. Minimal set indistinguishability requires that, when the secret is sampled uniformly at random, the distribution of the shares of the parties in $A \cup B$ generated according to Γ is equal to the joint distribution of the shares of the parties in A generated according to Γ', and $|B|$ pseudoshares generated using the tracing key associated with Γ'.

Intuitively, this captures the idea that a reconstruction box built from the shares of an unauthorized subset cannot distinguish whether it is being queried with real shares generated under an access structure in which the parties from the query together with the corrupted subset form a minimal authorized subset, or with simulated pseudoshares generated according to a different access structure.

As with projection indistinguishability, this requirement is crucial for our tracing experiment because it guarantees that the reconstruction box behaves consistently for both real and simulated inputs, which allows the tracer to extract meaningful identifying information from it. However, as explained later in Section 6.1, to support effective tracing in this setting, the indistinguishability must hold specifically with respect to a minimal authorized subset. The formal definition follows.

Definition 4.13 (Perfect Minimal Set Indistinguishability). *Let $m > 1$ be a power of two, let $\mathcal{S}$ be a finite set of secrets with $|\mathcal{S}| = m^2$, let $\mathcal{F}$ be a family of access structures over n parties, and let Σ be a traceable scheme for $\mathcal{F}$. Then, Σ is minimal set indistinguishable if for any secret $s \in \mathcal{S}$, any $p \in [2 \log m]$, any pair of access structures $\Gamma, \Gamma' \in \mathcal{F}$, and any pair of disjoint subsets $A, B \subseteq [n]$ such that $A, B \notin \Gamma \cup \Gamma'$, and $A \cup B$ is a minimal authorized subset in Γ, it holds that the distributions*

$$(\{\mathsf{sh}_i\}_{i \in A \cup B}) \text{ and } (\{\mathsf{sh}'_i\}_{i \in A}, \{\mathsf{sh}''_i\}_{i \in [k']})$$

are equal, where $k = |A|$, $k' = |B|$, $s \leftarrow_{\$} \mathcal{S}$, $(\{\mathsf{sh}_i\}_{i \in [n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s, \Gamma)$, $(\{\mathsf{sh}'_i\}_{i \in [n]}, \mathsf{tk}', \mathsf{vk}') \leftarrow \mathsf{Share}(s, \Gamma')$, and $\{\mathsf{sh}''_i\}_{i \in [k']} \leftarrow \mathsf{ShGen}(\mathsf{tk}', k, k', p)$.

5 A Projection Traceable Scheme for General Access Structures

In this section, we present a projection traceable secret sharing scheme for general access structures in the information-theoretic setting. We begin by outlining

the intuition behind the role of anonymity in enabling traceability. We then describe the construction of the scheme in full detail and provide a formal proof of its correctness and security. Finally, we discuss further optimizations and show how the scheme can be made efficient for some specific families of access structures.

5.1 The Need for Anonymity

At first glance, anonymity and traceability may seem at odds: while anonymity hides the identities of the parties holding shares, traceability aims to reveal which corrupted parties contributed to constructing a malicious reconstruction box. However, as observed by [BPR24], anonymity is in fact a fundamental ingredient for achieving efficient traceability. Specifically, it is necessary to ensure that the tracing procedure can be carried out with a polynomial number of queries.

The core reason is that if shares were linked to party identities, then a malicious reconstruction box could recognize whether a given query comes from the tracer. In particular, a reconstruction box built by a set of corrupted parties A could inspect the identifiers associated with the shares in a query and check whether any of them belong to A. If so, since none of these parties would query their own reconstruction box, it could deduce that the query contains non-valid or simulated shares and refuse to respond meaningfully by returning $\perp$.

This forces the tracer to avoid querying the reconstruction box with any shares from the corrupted parties. However, since the tracer does not know which parties built the reconstruction box, ensuring that a query avoids all corrupted parties is non-trivial. In some cases, for access structures with a large number of unauthorized subsets, this can lead to an expected exponential number of queries before obtaining a useful response. This issue is illustrated in [BPR24] for the case of threshold access structures in which both the threshold and the number of corrupted parties are a constant fraction of the total number of parties.

5.2 The Scheme

For simplicity, throughout this subsection we assume that the tracer knows in advance the number of corrupted parties c. In Section 5.4, we explain how to modify our tracing procedure to remove this assumption and allow the tracing algorithm to work even when the number of corrupted parties is unknown.

Our construction combines the anonymous secret sharing scheme of Fig. 3 with a key observation: any two shares generated by the same scheme for the same party, even if they correspond to different secrets, are indistinguishable to that party.[9] The core ideas underlying each of the algorithms in our construction are as follows.

[9] This holds due to the scheme's satisfaction of the stronger anonymity notion known as *multi-dealer anonymity* [BGI$^+$25], as discussed in Section 3.

Share. We begin by sharing the secret s among the parties according to the access structure Γ using the anonymous scheme of Fig. 3, but now taking $\mathbb{Z}_m^2$ as the domain of the secret. Additionally, for each party i, we assign a secret identifier $\mathsf{id}_i = (\mathsf{id}_i^1, \mathsf{id}_i^2)$ sampled uniformly from the same domain. The first component id_i^1 is part of the tracing key and will be used by the tracer to identify corrupted parties, while the second component id_i^2 is part of the verification key and can be used by the verifier to validate the accusation issued by the tracer.

Next, for each unauthorized subset $A \notin \Gamma$, we find the smallest disjoint subset $B \subseteq [n]$ such that $A \cup B \in \Gamma$. We create $|B|$ dummy parties $d_1, \ldots, d_{|B|}$ and select uniformly at random one party $i \in A$. Then, for each bit $b \in \{0,1\}$ of the identifier id_i, we independently share the corresponding value s_b among the set $A \cup \{d_1, \ldots, d_{|B|}\}$ using the DNF additive scheme. The shares assigned to the dummy parties are included in the tracing key and will later allow the tracer to reconstruct bitwise the identifier id_i, thereby identifying the corrupted part i.

Intuitively, this construction fools the set A into believing that the shares obtained from the bitwise encoding of id_i correspond to shares of the secret s for the authorized set $A \cup B$. Since these shares are indistinguishable, the reconstruction box will not be able to tell whether they came from s or from the encoding of id_i. As a result, when for each bit b of id_i the tracer queries the box with the dummy parties' shares $d_1, \ldots, d_{|B|}$, it treats them as valid shares of s for $A \cup B$ and outputs the corresponding value s_b. By repeating this process for all bits, the tracer reconstructs the identifier id_i, exposing a corrupted party.

Finally, we assign as many dummy random shares to each party to ensure uniform share sizes across all parties.

Projection & Reconstruction. Since the secret is shared using the anonymous scheme of Fig. 3, the Projection and Reconstruction algorithms remain the same.

ShareGeneration. For given sizes k and k', the algorithm ShareGeneration is run by the tracer during the tracing procedure and constructs k' projected pseudoshares of level $k + k'$ for the p-th bit of the identifier. Each pseudoshare is created from dummy shares contained in the tracing key, which are generated by secret sharing the bitwise encoding of the identifiers across all subsets of the form $A \cup \{d_1, \ldots, d_{k'}\}$, where $A \subseteq [n]$ is an unauthorized subset of size k, and $d_1, \ldots, d_{k'}$ are dummy parties. To complete each projected pseudoshare, we add as many uniformly random values as necessary to match the size of an actual projected share of size $k + k'$.

In general terms, in this step, we use the tracing key to construct projected pseudoshares for the p-th bit of the identifier that are indistinguishable from actual projected shares, which are later used to fool the reconstruction box during the tracing procedure. Moreover, this approach allows us to simultaneously trace all unauthorized subsets of size k that require k' additional shares to become authorized.

Trace. The tracing procedure is run by the tracer and proceeds iteratively for each bit of id_i. Since the reconstruction box accepts projected shares of a given size, for each level $\ell \in [n-c]$ and for each identifier's bit b, we generate ℓ projected pseudoshares of level $c+\ell$ using the $\mathsf{ShareGeneration}$ algorithm, and we query the box with them. As previously explained, the reconstruction box cannot distinguish these projected pseudoshares from genuine shares of the secret. Therefore, when queried with them, a *good* reconstruction box will run the $\mathsf{Reconstruction}$ algorithm on the pseudoshares and output s_b, which reveals the bit b of the identifier id_i.

After completing the $2\log m$ queries needed to recover all bits of id_i, the tracer reconstructs the full identifier. It then compares the first component id_i^1 with the corresponding part of the identifiers present in the tracing key. This step ensures that the output is a valid identifier and determines the corresponding party i. Once this check is satisfied, the second recovered component id_i^2 is used as a proof π, which is then provided to the verifier.

Since the reconstruction box is not perfect, for each bit of the identifier the tracing procedure is repeated $\mathrm{poly}(\lambda)$ times, and the tracer outputs the majority value among the responses. By the Chernoff bound, this majority-vote rule ensures that, at the correct level[10], the reconstruction box returns all the correct bits of the identifier id_i with overwhelming probability.

Verify. The verification procedure is executed by the verifier and consists of a simple check: it verifies whether any of the values provided in the proof π matches the second component of the identifier from the verification key corresponding to the accused party. If so, it outputs 1; otherwise, it returns 0. Intuitively, for a malicious tracer to falsely accuse an honest party i, it would need to correctly guess the second component id_i^2 of that party's identifier. Thus, if this value is sampled uniformly from a large domain, the probability of such a guess succeeding is negligible.

We provide in Fig. 9 an outline of the scheme's construction. The full details of the algorithms are given in Fig. 10 and Fig. 11. The formal statement is presented in Theorem 5.1.

Theorem 5.1. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in [0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m^2$, the protocol Σ of Fig. 10 and Fig. 11 is a projection traceable secret sharing scheme with statistical reconstruction, projection traceability, and projection non-imputability realizing Γ with share size $O(2^n \mathrm{poly}(\log m))$ and $O(\mathrm{poly}(\lambda, n)\log m)$ queries.*

[10] That is, the level corresponding to the actual number of additional parties required for the corrupted parties to form an authorized set.

$\mathsf{Share}(s, \Gamma)$:

1. Share the secret $s \in \mathbb{Z}_m^2$ among Γ with the general anonymous scheme of Fig. 3.
2. Assign a uniformly random identifier $\mathsf{id}_i = (\mathsf{id}_i^1, \mathsf{id}_i^2) \in \mathbb{Z}_m^2$ to each party $i \in [n]$.
3. Append all the first parts of the identifiers id_i^1 in the tracing key.
4. Append all the second parts of the identifiers id_i^2 in the verification key.
5. For each $A \notin \Gamma$
   (a) Find the smallest subset B such that $A \cup B \in \Gamma$.
   (b) Consider $|B|$ dummy parties $d_1, ..., d_{|B|}$.
   (c) Select randomly a party $i \in A$.
   (d) For each bit $b \in \{0, 1\}$ of id_i
      i. Share s_b among $A \cup \{d_1, ..., d_{|B|}\}$ with the DNF additive scheme.
   (e) Ensure a uniform share size among all parties sending dummy random values to them.
   (f) Append the shares of the dummy parties $d_1, ..., d_{|B|}$ in the tracing key.

$\mathsf{Proj}(\mathsf{sh}_i)$: The same as in the general anonymous scheme of Fig. 3.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i \in A}, |A|)$: The same as in the general anonymous scheme of Fig. 3.

$\mathsf{ShGen}(\mathsf{tk}, k, k', p)$:

1. Use as the k' projected pseudoshares of level $k + k'$ the portions of the shares corresponding to the dummy parties $d_1, \ldots, d_{k'}$ that are obtained by secret sharing the p-th bit of the identifier with all unauthorized subsets of size k.
2. Ensure that all k' projected pseudoshares match the size of actual projected shares of level $k + k'$ by padding dummy random values to them.

$\mathsf{Trace}^R(\mathsf{tk}, c, \lambda)$:

1. For each size $\ell \in [n - c]$.
   (a) For each bit $b \in \{0, 1\}$ of id_i
      i. Repeat $\mathrm{poly}(\lambda)$ times
         A. Compute ℓ projected pseudoshares of level $c + \ell$ using the ShGen algorithm.
         B. Query the reconstruction box R with these projected pseudoshares.
         C. Store the output bit b^*.
      ii. Take as the bit b the majority value among all the $\mathrm{poly}(\lambda)$ outputs b^*.
   (b) If the first part of the reconstructed identifier matches any first part of the identifiers in the tracing key, terminate and output the second part of the reconstructed identifier as the proof π.

$\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi)$:

1. If the proof π matches any second part of the identifiers in the verification key, output 1.
2. Otherwise, output 0.

Fig. 9: Outline of a projection traceable secret sharing scheme for any access structure Γ.

Notation:

1. Let Γ_k be the family of authorized subsets in Γ of size k.
2. Let $\overline{\Gamma} = 2^{[n]} \setminus \Gamma$ be the family of unauthorized subsets in Γ.
3. Let $\overline{\Gamma}_k$ be the family of subsets in $\overline{\Gamma}$ of size k.
4. Let $\overline{\Gamma}_{k,\ell}$ be the family of subsets in $\overline{\Gamma}_k$ for which the smallest disjoint subset they require to form an authorized set in Γ has size ℓ.
5. Let $\Phi_t = \Gamma_t \cup \bigcup_{k+\ell=t} \overline{\Gamma}_{k,\ell}$.
6. Let P_t denote the set of permutations of $|\Phi_t|$ elements and P'_m the set of permutations of $2\log m$ elements.

$\mathsf{Share}(s, \Gamma)$:

1. Share $s \in \mathbb{Z}_m^2$ with the anonymous general scheme Σ of Fig. 3 among Γ, i.e.
$$\{\mathsf{sh}^s_{i,2\log m}\}_{i\in[n]} \leftarrow \Sigma(s, \Gamma).$$
2. Parse $\mathsf{sh}^s_{i,2\log m}$ as $\{\mathsf{sh}^s_{i,2\log m,k}\}_{k\in[n]}$ for $i \in [n]$.
3. Parse $\mathsf{sh}^s_{i,2\log m,k}$ as $\{\mathsf{sh}^*_{i,2\log m,k,A}\}_{A\in\Gamma_k}$ for $i \in [n]$, $k \in [n]$.
4. For $q \in [2\log m - 1]$
   1. Share a random value $r \in \mathbb{Z}_m^2$ with the anonymous general scheme Σ of Fig. 3 among Γ, i.e. $\{\mathsf{sh}^s_{i,q}\}_{i\in[n]} \leftarrow \Sigma(s, \Gamma)$.
5. Select a permutation $\sigma'_m \in P'_m$ uniformly at random, i.e. $\sigma'_m \leftarrow_\$ P'_m$.
6. For $q \in [2\log m]$
   1. Set $p \leftarrow \sigma'_m(q)$ and $\mathsf{sh}^s_{i,p} \leftarrow \mathsf{sh}^s_{i,q}$ for $i \in [n]$.
7. Set $\mathsf{id}_i = (\mathsf{id}^1_i, \mathsf{id}^2_i) \leftarrow_\$ \mathbb{Z}_m^2$ for $i \in [n]$.
8. Parse id^k_i as $\{\mathsf{id}^k_{i,j}\}_{j\in[m]}$ for $i \in [n], k \in [2]$.
9. For $k \in [n]$
   (a) For $\ell \in [n-k]$
      i. For $A \in \overline{\Gamma}_{k,\ell}$.
         A. Let d_1,...,d_ℓ be ℓ extra dummy parties, and let $B = A \cup \{d_1, ..., d_\ell\}$.
         B. Select a party $j \in A$ uniformly at random, i.e. $j \leftarrow_\$ A$.
         C. For $q \in [2\log m]$
            1. Set $b \leftarrow \mathsf{id}^p_{j,q}$.
            2. Share s_b with the anonymous additive scheme Σ' of Fig. 1 among B, i.e. $\{\mathsf{sh}^*_{i,k+\ell,q,A}\}_{i\in B} \leftarrow \Sigma'(s_b, B)$.
            3. Set as the traceable key for the q bit of the identifier for the set A the shares of the ℓ dummy parties in B, i.e. $\mathsf{tk}^*_{q,A} \leftarrow \{\mathsf{sh}^*_{i,q,k+\ell,A}\}_{i\in B\setminus A}$.
            4. Send two random elements $r_0, r_s \in \mathbb{Z}_m^2$ to each of the parties not in A, i.e. $\mathsf{sh}^*_{i,q,k+\ell,A} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $i \in [n] \setminus A$.
         D. Set $\mathsf{tk}^*_A \leftarrow \{\mathsf{tk}^*_{q,A}\}_{q\in[2\log m]}$ for $A \in \overline{\Gamma}$.
10. For $t \in [n]$
    (a) Select a permutation $\sigma_t \in P_t$ uniformly at random, i.e. $\sigma_t \leftarrow_\$ P_t$.
    (b) For $A \in \Phi_t$
       i. Set $B \leftarrow \sigma_t(A)$.
       ii. Set $\mathsf{sh}_{i,q,t,B} \leftarrow \mathsf{sh}^*_{i,q,t,A}$ for $i \in [n], q \in [2\log m]$.
       iii. If $A \notin \Gamma$, set $\mathsf{tk}_B \leftarrow \mathsf{tk}^*_A$.
    (c) Set $\mathsf{sh}_{i,q,t} \leftarrow \{\mathsf{sh}_{i,q,t,A}\}_{A\in\Phi_t}$ for $i \in [n], q \in [2\log m]$.
    (d) Set $\mathsf{tk}_t \leftarrow (\mathsf{id}^1_t, \sigma_t, \{\mathsf{tk}_B\}_{B\in\mathcal{J}})$, where $\mathcal{J} = \{\sigma_t(A) : A \in \Phi_t \setminus \Gamma_t\}$.
11. Set $\mathsf{sh}_{i,q} \leftarrow \{\mathsf{sh}_{i,q,k}\}_{k\in[n]}$ for $i \in [n], q \in [2\log m]$.
12. Set $\mathsf{tk} \leftarrow \{\mathsf{tk}_i\}_{i\in[n]}$.
13. Set $\mathsf{vk} \leftarrow \{\mathsf{id}^2_i\}_{i\in[n]}$.
14. Output $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$:

1. Parse sh_i as $\{\mathsf{sh}_{i,k}\}_{k\in[n]}$.
2. Set $\mathsf{psh}_{i,k} \leftarrow \mathsf{sh}_{i,k}$, i.e. the projection algorithm is the projection function.
3. Output $\mathsf{psh}_{i,k}$.

Fig. 10: Share and Projection algorithms of a projection traceable secret sharing scheme for any access structure Γ.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k)$:

1. Parse $\mathsf{psh}_{i,k}$ as $\{\mathsf{sh}_{i,q,k,B}\}_{B\in\Phi_k}$ for $i \in A, q \in [2\log m]$.
2. Set $s' \leftarrow_\$ \mathbb{Z}_m^2$.
3. For $B \in \Phi_k$
   1. For $q \in [2\log m]$
      (a) Parse $\mathsf{sh}_{i,q,k,B}$ as $(\mathsf{sh}_i^0, \mathsf{sh}_i^s)$ for $i \in A$.
      (b) Set $u \leftarrow \sum_{i\in A} \mathsf{sh}_i^0$.
      (c) If $u = 0$, set $s' \leftarrow \sum_{i\in A} \mathsf{sh}_i^s$ and **break**.
4. Output s'.

$\mathsf{ShGen}(\mathsf{tk}, k, k', p)$:

1. Parse tk as $\{\mathsf{tk}_i\}_{i\in[n]}$.
2. Parse tk_i as $(\mathsf{id}_i^1, \sigma_i, \{\mathsf{tk}_B\}_{B\in\mathcal{J}})$ for $i \in [n]$, where $\mathcal{J} = \{\sigma_i(A) : A \in \Phi_i \setminus \Gamma_i\}$.
3. Parse tk_B as $\{\mathsf{tk}_{q,B}\}_{q\in[2\log m]}$ for $B \in \{\sigma_{k+k'}(A) : A \in \overline{\Gamma}_{k,k'}\}$.
4. Parse $\mathsf{tk}_{q,B}$ as $\{\mathsf{sh}_{i,q,k+k',B}\}_{i\in[k']}$ for $q \in [2\log m]$.
5. For $i \in [k']$
   1. For $q \in [2\log m]$
      (a) For $B \in \Phi_{k+k'}$
         i. Set $A \leftarrow \sigma_i^{-1}(B)$.
         ii. If $A \in \overline{\Gamma}_{k,k'}$ and $q = p$
            A. Set $\mathsf{psh}'_{i,q,k+k',B} \leftarrow \mathsf{sh}_{i,q,k+k',B}$.
         iii. Else
            A. Send two random elements $r_0, r_B \in \mathbb{Z}_m^2$ to the party i, i.e.
            $$\mathsf{psh}'_{i,q,k+k',B} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2).$$
         iv. Set $\mathsf{psh}'_{i,q,k+k'} \leftarrow \{\mathsf{psh}'_{i,q,k+k',B}\}_{B\in\Phi_{k+k'}}$
      (b) Set $\mathsf{psh}'_{i,k+k'} \leftarrow \{\mathsf{psh}'_{i,q,k+k'}\}_{q\in[2\log m]}$.
6. Output $\{\mathsf{psh}'_{i,k+k'}\}_{i\in[k']}$.

$\mathsf{Trace}^R(\mathsf{tk}, c, \lambda)$:

1. Set three empty lists L, L', L''.
2. For $\ell \in [n-c]$
   1. For $q \in [2\log m]$
      (a) For $h \in [\mathrm{poly}(\lambda)]$
         i. Set $\{\mathsf{psh}'_{i,c\ell,h}\}_{i\in[\ell]} \leftarrow \mathsf{ShGen}(\mathsf{tk}, c, \ell)$.
         ii. Query R on $\{\mathsf{psh}'_{i,c\ell,h}\}_{i\in[\ell]}$. Let b_h be the response.
         iii. Store b_h in L''.
      (b) Set $\mathsf{id}_{\ell,q}$ the majority value in L'' and erase L''.
   2. Set $\mathsf{id}_{\ell,1} \leftarrow \{\mathsf{id}_{\ell,q}\}_{q\in[m]}$ and set $\mathsf{id}_{\ell,2} \leftarrow \{\mathsf{id}_{\ell,q}\}_{q\in[2\log m]\setminus[\log m]}$.
   3. If $\mathsf{id}_{h,\ell}^1 = \mathsf{id}_i^1$ for some $i \in [n]$
      (a) If $i \notin L$, add i to L.
      (b) If $\mathsf{id}_{h,\ell}^2 \notin L'$, add $\mathsf{id}_{h,\ell}^2$ to L'.
3. If $|L| \neq 1$, terminate and output $\perp$.
4. Set $\mathcal{I} \leftarrow \{i\}$, where i is the unique element in L.
5. Set $\pi \leftarrow L'$.
6. Output $(\mathcal{I}, \pi)$.

$\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi)$:

1. Parse vk as $\{\mathsf{id}_i^2\}_{i\in[n]}$.
2. Let i' be the unique element in $\mathcal{I}$.
3. If there exists $\mathsf{id}_{h,\ell}^2 \in \pi$ such that $\mathsf{id}_{h,\ell}^2 = \mathsf{id}_{i'}^2$, output 1. Otherwise output $\perp$.

Fig. 11: Reconstruction, ShareGeneration, Trace, and Verify algorithms of a projection traceable secret sharing scheme for any access structure Γ.

5.3 Proof of Theorem 5.1

To facilitate comprehension, we break down the proof of Theorem 5.1 into a sequence of propositions and lemmas. We begin by proving that the protocol described in Fig. 10 and Fig. 11 is an anonymous secret sharing scheme.

Proposition 5.2. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the protocol Σ of Fig. 10 and Fig. 11 is an anonymous scheme with statistical privacy realizing Γ with share size $O(2^n \mathrm{poly}(\log m))$.*

Proof. We begin by proving the privacy of the scheme. Observe that the Share algorithm distributes the secret using the same mechanism as the anonymous scheme described in Fig. 3. Moreover, the additional components of the shares either consist of uniformly random values or correspond to shares of fake secrets s_b that are used to recover bits b of the identifier. These fake secrets are shared using the anonymous additive scheme, and their unique authorized subset includes dummy parties whose shares are not available. Consequently, these shares are indistinguishable from random values and do not give any information about the secret. Therefore, the overall privacy of the scheme directly follows from the privacy guarantee of the construction in Fig. 3.

Regarding the statistical correctness of the scheme, note that the Projection and Reconstruction algorithms remain identical to those of the anonymous scheme in Fig. 3, with the only difference being the increased number of candidate share components that must be checked during reconstruction. However, this number is still bounded by $O(2^n \log m)$. Furthermore, as in the scheme of Fig. 3, for any authorized set of size k all except one of the candidate share components of level k are uniform random values because of the previous analysis on the Share algorithm. Consequently, by choosing a sufficiently large group size m, the statistical correctness follows by the same argument used in the proof of the anonymous scheme of Fig. 3.

Next, perfect projected correctness and perfect anonymous reconstruction follow from the above observations on the Share, Projection, and Reconstruction algorithms, combined with the fact that the anonymous scheme of Fig. 3 already satisfies both properties.

Finally, for the share size it suffices to notice that in each copy of the scheme, each party receives at most two share components of $\log m$ bits for each subset (authorized or not) of $2^{[n]}$. Since there are $2 \log m$ copies of the scheme, the total share size per party is $O(2^n \log^2 m)$. □

Remark 5.3. Indeed, the proof of Proposition 5.2 shows that the scheme of Theorem 5.1 satisfies the strongest notion of share anonymity, that is, *uniform anonymity*.

Next, we prove that the scheme satisfies perfect tracer privacy.

Proposition 5.4. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$,*

the anonymous scheme Σ of Fig. 10 and Fig. 11 realizing Γ has perfect tracer privacy.

Proof. To prove the perfect tracer privacy requirement, it suffices to combine the privacy requirement proved in Proposition 5.2 with the fact that the tracing key only contains the share components corresponding to fake secrets s_b that are used to recover bits b of the identifier. Since these are generated independently of the secret, they reveal no additional information about it. Consequently, the perfect tracer privacy requirement holds. □

Before proving that the scheme satisfies statistical projection traceability and non-imputability, we first establish a technical lemma showing that the scheme achieves perfect projection indistinguishability. This property will play a crucial role in the subsequent proof of the scheme's traceability.

Lemma 5.5. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the anonymous scheme Σ of Fig. 10 and Fig. 11 realizing Γ has perfect projection indistinguishability.*

Proof. Let $p \in [2 \log m]$, let $A, B \subseteq [n]$ be two subsets as described in the statement of Definition 4.7, and let $d_1, \ldots, d_{n-1}$ denote the additional $n-1$ dummy parties used in the Share algorithm.

First, by the privacy and reconstruction analysis provided in the proof of Proposition 5.2, all share components in $\{\mathsf{sh}_i\}_{i \in A} \cup \{\mathsf{psh}_{i,k+k'}\}_{i \in [k']}$ behave as uniformly random values, except for the unique share component corresponding to the authorized subset $A \cup B$, which consists of valid shares of the secret s.

Similarly, by the construction of the Share and $\mathsf{ShareGeneration}$ algorithms, an analogous argument shows that all share components in $\{\mathsf{sh}_i\}_{i \in A} \cup \{\mathsf{psh}'_{i,k+k'}\}_{i \in [k']}$ are also uniformly random, except for the share component corresponding to the subset $A \cup \{d_1, \ldots, d_{k'}\}$, which consists of shares of a fake secret s_b that are used to recover the p-th bit b of the identifier.

Since both s and s_b are sampled uniformly at random from $\mathbb{Z}_m^2$, it follows that the two distributions in the definition of projection indistinguishability are identical. Indeed, both distributions can be viewed as being generated by the same Share procedure, where the secret s is replaced by the fake secret s_b, and the parties in B are replaced by the dummy parties $d_1, \ldots, d_{k'}$. This concludes the proof of the perfect projection indistinguishability requirement. □

Once the lemma is proved, we use it to prove that the scheme satisfies efficient statistical projection traceability.

Proposition 5.6. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in [0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m^2$, the anonymous scheme Σ of Fig. 10 and Fig. 11 realizing Γ has statistical projection traceability and requires $O(\mathrm{poly}(\lambda, n) \log m)$ queries.*

Proof. Let $\mathcal{A}$ be any adversary participating in the $\mathsf{Adv}^{\mathrm{proj-trac}}_{\mathcal{A},\Sigma,\epsilon}$ security experiment, let A be the unauthorized subset of corrupted parties and s_0, s_1 be the secrets selected by $\mathcal{A}$ at the beginning of the experiment with $|A| = c$ (Fig. 5, Step 1), let $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in A}$ be the shares received by $\mathcal{A}$ from the challenger (Fig. 5, Step 4), and let tk be the tracing key received by the tracer from the challenger (Fig. 5, Step 3). Let id_{i^*} denote the shared identifier corresponding to the set A (Fig. 9, Step 5.(c) of Share), let $k' \in [n-c]$ be such that $A \in \overline{\Gamma}_{c,k'}$, and let $B \subseteq [n]$ be a subset of size k' such that $A \cup B \in \Gamma$.[11]

Let G be the event in which $\mathcal{A}$ outputs a reconstruction box R that is $(\Gamma, A, \mathbf{sh}, s_0, s_1, \epsilon(\lambda))$-projection good (Fig. 5, Step 7). Conditioned on $\neg\mathsf{G}$, the experiment outputs 0 with probability 1, so we condition the rest of the analysis on G.

For each $\ell \in [n-c]$, $q \in [2\log m]$, let $\mathsf{id}_{\ell,q}$ be the majority bit obtained from the $\mathrm{poly}(\lambda)$ queries to R at the (ℓ, q)-th iteration, and let $\mathsf{id}_\ell = (\mathsf{id}^1_\ell, \mathsf{id}^2_\ell)$ be the identifier constructed by padding all these bits. We define the following events:

- Let E_1 be the event in which $\mathsf{id}_{k',q} \neq \mathsf{id}_{i^*}$ for some $q \in [2\log m]$, i.e., any of the output bits at the correct level does not match that of id_{i^*}.
- For $\ell \in [n-c]$, let E^ℓ_2 denote the event in which there exists $i \in [n] \setminus \{i^*\}$ such that $\mathsf{id}^1_{h,\ell} = \mathsf{id}^1_i$, i.e., the first component of the constructed identifier at level ℓ matches that of a party different from i^*.
- Let E_3 be the event in which $\mathsf{id}^1_i = \mathsf{id}^1_{i'}$ for some distinct $i, i' \in [n]$.

Clearly, conditioned on G, the adversary wins the experiment (i.e., the output is 1) only if one of the events above occurs. We now bound the probability of each of these events.

We start by bounding E_1. For each $q \in [2\log m]$ and $h \in [\mathrm{poly}(\lambda)]$, let $b_{k',q,h}$ be the output of R to the (k', q, h)-th query. By assumption G holds, so:

$$\Pr[b = R(\{\mathsf{psh}_{i,c+k'}\}_{i\in B})] \geq \frac{1}{2} + \epsilon(\lambda),$$

where $\{\mathsf{psh}_{i,c+k'}\}_{i\in B}$ correspond to the actual projected shares of level $c + k'$ of the parties in B derived from the same sharing of s according to Γ as $\mathbf{sh}$.

Then, by perfect projection indistinguishability (Lemma 5.5), we have:

$$\Pr[b_{k',q,h} = b] \geq \frac{1}{2} + \epsilon(\lambda)$$

for each $q \in [2\log m]$ and $h \in [\mathrm{poly}(\lambda)]$.

From there, since R is stateless and queries are independent, by taking $\mathrm{poly}(\lambda)$ sufficiently large and using the Chernoff bound, we get that the probability that $\mathsf{id}_{k',q}$ does not match the q-th bit of id_{i^*} is negligible for each $q \in [2\log m]$.

Therefore, since $2\log m = O(\mathrm{poly}(\lambda))$ we conclude that $\Pr[\mathsf{E}_1]$ is negligible in λ.

Next, we move to bound E^ℓ_2 for $\ell \in [n-c]$. By the construction of the scheme, all fake secrets that allow to recover the identifiers are shared with

[11] Such a subset B exists by assumption, since $A \in \overline{\Gamma}_{c,k'}$.

the anonymous additive scheme of Fig. 1 and the reconstruction box is only able to recover the corresponding identifier for the queries of level k'. At this level, the shared identifier is precisely id_{i^*}, so these queries have no knowledge of any other identifier. For the remaining levels $\ell \in [n-c] \setminus \{k'\}$, the privacy property of the scheme implies that the output of R on these queries reveals no information about id_{i^*}. Thus, since id_{i^*} is sampled uniformly from a super-polynomial domain, we conclude that $\Pr[\mathsf{E}_2^\ell]$ is negligible in λ.

Now, we bound E_3. Since the identifiers $\{\mathsf{id}_i^1\}_{i\in[n]}$ are independently sampled from a domain of super-polynomial size, the probability that any two parties share them is negligible in λ.

Finally, we have proved that each of the events under which the adversary could win the experiment occurs with negligible probability in λ. Therefore, by the union bound, the overall success probability of $\mathcal{A}$ in the $\mathsf{Adv}^{\mathrm{trac}}_{\mathcal{A},\Sigma,\epsilon}$ experiment is negligible in λ, as desired. □

We conclude the proof of Theorem 5.1 by showing that the scheme satisfies statistical projection non-imputability.

Proposition 5.7. *Let Γ be any access structure over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the anonymous scheme Σ of Fig. 10 and Fig. 11 realizing Γ has statistical projection non-imputability.*

Proof. Let $\mathcal{A}$ be any adversary participating in the $\mathsf{Adv}^{\mathrm{proj-ni}}_{\mathcal{A},\Sigma}$ security experiment, let $i^* \in [n]$ be the honest party selected by $\mathcal{A}$ at the beginning of the experiment (Fig. 6, Step 1), and let $\mathsf{id}_{i^*} = (\mathsf{id}^1_{i^*}, \mathsf{id}^2_{i^*})$ denote the identifier of the party i^* (Fig. 9, Step 2 of Share). Let $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ be the shares received by $\mathcal{A}$ from the challenger (Fig. 5, Step 3).

Let E_{i^*} be the event in which $\mathcal{A}$ learns $\mathsf{id}^2_{i^*}$. Clearly, by construction, the experiment $\mathsf{Adv}^{\mathrm{ni}}_{\mathcal{A},\Sigma}$ outputs 1 only if E_{i^*} occurs. We now bound the probability of this event.

By construction of the Share algorithm, all components of $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ that relate to fake secrets that allow to recover the identifier id_{i^*} correspond to unauthorized subsets because id_{i^*} is only shared using the anonymous additive scheme of Fig. 1 with subsets that include i^*. Furthermore, all other components of $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ are generated independently of id_{i^*} and thus carry no information about it. Therefore, by the privacy property of the anonymous additive scheme from Fig. 1, the adversary $\mathcal{A}$ obtains no information about the value $\mathsf{id}^2_{i^*}$.

Combining this with the fact that $\mathsf{id}^2_{i^*}$ is sampled uniformly from a super-polynomial domain, we conclude that the probability of event E_{i^*} is negligible in λ, as desired. □

5.4 Further Optimizations

We present additional optimizations of the scheme from Theorem 5.1. First, we show how to reduce the share size for some access structures by leveraging the

anonymous threshold scheme introduced in Section 3.2. Then, we extend the general scheme to the setting where the tracer does not know the number of corrupted parties that contributed to the reconstruction box.

Efficient Schemes for Low Slices. The following remark describes how to optimize our construction in order to obtain an efficient projection traceable scheme for low slices.

Remark 5.8. As in the case of anonymous schemes, the scheme described above can be optimized by combining it with the anonymous threshold scheme from Fig. 2. Specifically, if Γ is an access structure in which all unauthorized subsets have size at most $t-1$, then the share size can be reduced by replacing the scheme used to share the secret in Step 1 of the Share algorithm in Fig. 10 with the optimized construction described in Remark 3.13. However, this optimization also requires adjusting the sharing procedure for identifiers associated with unauthorized subsets in $\overline{\Gamma}_{k,\ell}$ for which $k+\ell=t$, to ensure compatibility with the scheme used for sharing the secret.

Remark 5.8 leads to efficient projection traceable schemes with polynomial share size for a specific family of access structures: low slices. That is, access structures over n parties given by t-slices in which $t \ll n$.

Corollary 5.9. *Let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(n)$, let $q = \lambda^{\omega(1)}$ be a prime power, let $\epsilon(\lambda) \in [0, \frac{1}{2}]$ be a non-negligible function, let $t \in [n]$ with $t = O(1)$, and let Γ be a t-slice access structure over n parties. For any $s \in \mathbb{F}_q$ there exists a projection traceable secret sharing scheme with statistical reconstruction, projection traceability and projection non-imputability realizing Γ with share size $O(n^t \log q)$ and $O(\lambda)$ queries.*

Proof. First, it is straightforward to check that $\mathcal{F}$ is a covering family.

Then, by construction, any access structure over n parties given by a t-slice with $t = O(1)$ only has $O(n^t)$ subsets with size strictly less than t. Thus, it suffices to apply the scheme of Remark 5.8.

Since all subsets of size greater than t are authorized, the Trace algorithm can reduce the number of iterations from $n-c$ to $t-c$, each consisting of $\mathrm{poly}(\lambda)$ queries. □

Learning the Number of Corrupted Parties. The following remark explains how our scheme can be adapted to support projection traceability even when the number of corrupted parties involved in building the reconstruction box is not known in advance.

Remark 5.10. To handle the case where the number of corrupted parties c is not known in advance, we follow the approach introduced by [BPR24]. It suffices to invoke the Trace algorithm with successive values of $c = n-1, n-2, \ldots$ until it succeeds.

We now justify the soundness of the approach described in Remark 6.15. Let c^* denote the actual number of corrupted parties. By construction, for any $c > c^*$, the queries generated by the Trace algorithm consist of projected pseudoshares that only reconstruct to the fake secret s_b when combined with projected shares from a set of c real parties. By the privacy of the scheme, when $c > c^*$, these queries do not reveal any information about any bit of the identifier. As a result, the tracing procedure obtains a valid identifier with only negligible probability, and thus it succeeds only with negligible probability.

In contrast, when $c = c^*$, the tracing procedure corresponds exactly to the one analyzed in Proposition 6.11, which guarantees that the tracer fails in identifying a corrupted party with negligible probability. Thus, the tracing procedure can be adapted to the setting in which the amount of corrupted parties is not known in advance by simply testing all possible values of c in descending order.

6 A Traceable Scheme for General Access Structures

In this section, we construct a traceable secret sharing scheme for general access structures in the information-theoretic setting. We start by explaining why hiding the access structure is essential for achieving traceability. After that, we present the full construction of the scheme along with a proof of its security and correctness. We conclude by exploring optimizations that improve efficiency for particular families of access structures.

6.1 The Need for Hiding the Access Structure

In order to achieve traceability for general access structures in the setting introduced by [BPR24], it is essential to hide the access structure from the participating parties. To understand why, we first observe that all known tracing strategies in this model, including those from [BPR24] for threshold structures and our own generalization to arbitrary access structures, rely on a common approach.

These strategies exploit the assumption that the reconstruction box is *good*, meaning that it reconstructs the secret with high probability when given a valid set of shares as input. Then, the general tracing procedure involves querying the reconstruction box with carefully crafted shares. When these shares, combined with those internally held by the box, form a valid authorized subset, the box is expected to output the secret or, in the case of our construction, an identifier that can be used to trace one of the corrupted parties.

However, a subtle but crucial assumption underlies this strategy: the reconstruction box always uses all the shares it has at its disposal when computing the output. This assumption is justified only if the union of the corrupted subset and the query corresponds to a *minimal* authorized subset. If instead the union is a non-minimal authorized subset, the box may use only a subset of the available shares, those sufficient to form a minimal authorized subset, making it

difficult for the tracer to know which shares actually influenced the output. This uncertainty can break the tracing mechanism.

More specifically, when the combined set of shares includes more than one minimal authorized subset, the reconstruction box has several valid reconstruction paths. In the tracing technique described in Section 5, if these different paths produce inconsistent outputs (a situation that may occur when the input includes simulated shares used for tracing) the box can detect this inconsistency, realize that it is being queried by a tracer, and simply return $\perp$.

This issue is especially problematic for general access structures, as there are cases where an unauthorized subset is not contained in any minimal authorized subset. For example, in a graph access structure, a corrupted subset made up of two nodes with no edge between them cannot be extended to form a minimal authorized set. In such situations, existing tracing techniques break down. To address this issue, our approach hides the access structure from the parties, revealing only a family of access structures to which the actual one belongs. This allows the tracer to trick the reconstruction box into believing that the query shares, together with those of the corrupted parties, form a minimal authorized set in some structure of the family, enabling the tracing procedure to work.

Furthermore, if the goal is to trace all corrupted parties that contributed to the construction of the reconstruction box, rather than just a single one, hiding the access structure becomes essential. In particular, for access structures in which two or more parties play equivalent roles, corrupted parties can exploit this symmetry when building a reconstruction box to ensure that at least one of them cannot be traced. By hiding the access structure, such strategies are no longer available. The formal statement follows and the underlying intuition is made explicit in its proof.

Theorem 6.1 (Theorem 1.5 restated). *There does not exist any traceable secret sharing scheme for general access structures that traces back all the corrupted parties and keeps the access structure public.*

Proof. For any $n \geq 3$, let Σ be a traceable secret sharing scheme for general access structures, as in the statement of the theorem, and let Γ be an access structure over n parties whose expression as a monotone Boolean function is given by $f_\Gamma = f_{\Gamma'} \wedge (x_{n-1} \vee x_n)$, where Γ' is a non-empty access structure over the first $n-2$ parties.

First, observe that in Γ the parties $n-1$ and n play equivalent roles. That is, for any $A \subseteq \Gamma'$, it holds that $A \cup \{n-1\} \in \Gamma$ if and only if $A \cup \{n\} \in \Gamma$.

Now let $\mathcal{A} = \{n-1, n\}$ be the corrupted set that builds a good reconstruction box $R_\mathcal{A}$ for Σ. We argue that there exists a strategy for $R_\mathcal{A}$ that prevents the party n from being traced, leading to a contradiction.

Since $n-1$ and n have equivalent roles in Γ, $R_\mathcal{A}$ can always simulate internally another reconstruction box R_{n-1} that only uses the share of the party $n-1$. As a result, the observable behavior of $R_\mathcal{A}$ is identical to that of R_{n-1}.

Consequently, if Σ were able to trace back party n given black-box access to $R_\mathcal{A}$, then the same tracing procedure applied to R_{n-1} would falsely accuse

n, violating the non-imputability property of Σ. Therefore, Σ cannot trace all corrupted parties in this setting, reaching the desired contradiction. □

Later, in Remark 6.4 we discuss scenarios where it is reasonable to assume that parties do not know the access structure, and in Section 6.4 we show how to modify our traceable scheme to enable tracing all corrupted parties.

Covering Families of Access Structures. Unfortunately, the strategy described above only works under a specific condition: for every unauthorized subset in any access structure from the family, there must exist another access structure in the same family in which that subset is strictly included in a minimal authorized subset. This property is necessary to ensure that the tracer can always simulate a minimal authorized set during the tracing procedure.

Note that not all natural families of access structures satisfy this property. For instance, in the family of all graph access structures with n parties, some unauthorized subsets, such as a triple of vertices with no connecting edges, cannot be extended to a minimal authorized subset within any access structure of the family. In these cases, the proposed tracing technique fails, because the reconstruction box cannot be fooled into treating the input shares as belonging to a minimal authorized subset of an access structure of the family.

To formalize the class of families where our tracing approach is applicable, we introduce the notion of a *covering family of access structures*: a family in which every unauthorized subset of every access structure within the family is strictly contained in a minimal authorized subset of some access structure in the same family. The definition follows.

Definition 6.2. *A family of access structures over n parties $\mathcal{F}$ is a* covering family *if for every $\Gamma \in \mathcal{F}$ and every subset $A \notin \Gamma$, there exists $B \subseteq [n]$ and $\Gamma' \in \mathcal{F}$ such that $A \subsetneq B$ and B is minimal in Γ'.*

The following straightforward lemma states that the family of all access structures over n parties satisfies the covering property.

Lemma 6.3. *For every $n \in \mathbb{N}$, the family of all access structures over n parties is a covering family.*

Proof. It suffices to notice that for any subset $A \subsetneq [n]$ there exists an access structure over n parties in which A is a minimal authorized subset. □

Remark 6.4. Besides the family of all access structures, natural families such as k-slices, weighted threshold, and hierarchical are also covering. Thus, our tracing procedure applies to all of them, extending its relevance to settings where parties have only partial knowledge of the access structure. For example, in proof-of-stake blockchains, parties may know that the access structure is a weighted threshold, even if they lack full information about the exact weight distribution. In such cases, since the parties only know that the access structure belongs to a covering family, our tracing procedure remains applicable.

6.2 The Scheme

For simplicity, and following the approach in Section 5, we assume in this subsection that the tracer knows in advance the exact number of corrupted parties c. In Section 6.4, we explain how to relax this assumption by adapting the tracing mechanism so that it remains effective even when the number of corruptions is not known beforehand. Our construction builds upon the projection traceable scheme from Section 5, introducing slight modifications to adapt it to the traceability setting. At a high level, we replace the underlying anonymous scheme with the hiding anonymous scheme from Section 3.4 to ensure that the access structure remains hidden. Additionally, we modify the ShareGeneration and Trace algorithms to account for the fact that the reconstruction box now expects full shares as input, rather than just their projections.

We next outline the main ideas behind the construction and highlight its differences compared to the projection traceable scheme of Section 5.

Share. The overall structure of the Share algorithm in our construction closely mirrors the one used in the projection traceable scheme presented in Section 5. That is, the secret is first shared among the parties using an anonymous scheme, and for each unauthorized subset, we embed a party identifier bitwise in a way that later enables tracing. However, to accommodate the stronger requirements of our traceability model, particularly the need to hide the access structure, we introduce three key modifications.

First, instead of using the anonymous scheme from Section 3, we now share the secret using the hiding anonymous scheme described in Section 3.4. This ensures that the access structure remains hidden from the parties, a property that is crucial to enable tracing in the general setting, as discussed in Section 6.1.

Second, to address the issue posed by non-minimal authorized subsets, also described in Section 6.1, we modify how we share the identifiers for tracing. Rather than always sharing the identifier for an unauthorized subset using an extension to a minimal authorized subset within the same access structure, we allow the extension to occur within any access structure from the covering family $\mathcal{F}$ that contains the same unauthorized subset. This guarantees that the corrupted parties can always be made to appear as part of a minimal authorized subset from some access structure in the family, which is critical for the success of the tracing procedure.

Third, access structures in the covering family $\mathcal{F}$ may have different numbers of unauthorized subsets. Since we share a tracing identifier for every unauthorized subset of the selected access structure, the total number of components in the share could vary depending on which access structure was chosen. Thus, to prevent the share size from leaking information about the underlying access structure, we pad each party's share with dummy random values until the total size matches the maximum number of unauthorized subsets over all access structures in $\mathcal{F}$. This ensures that the share size is uniform and does not reveal which access structure from the covering family was used.

Projection & Reconstruction. Since the secret is shared using the hiding anonymous scheme of Fig. 4, the Projection and Reconstruction algorithms remain the same as the ones of the projection traceable scheme of Section 5.

ShareGeneration. The ShareGeneration algorithm in our construction follows the same overall approach as in the projection traceable scheme of Section 5. Namely, it takes the tracing key as input and uses it to simulate shares that will later be used as part of the tracer's queries to the reconstruction box. However, the key difference lies in the type of pseudoshares being generated.

In the projection traceable setting, the reconstruction box was assumed to operate on projected shares, so the ShareGeneration algorithm only needed to generate projected pseudoshares. In our setting, by contrast, the reconstruction box is allowed to request full shares from the parties. As a result, the ShareGeneration algorithm must now produce full pseudoshares.

To do this, we still use the tracing keys in the same way as in the projection traceable case: the pseudoshares are generated by taking the actual share components associated with the dummy parties introduced during the Share phase of the bitwise encoding of the identifiers for each unauthorized subset of the size given as input. The main difference is that, instead of padding these components with enough random values to match the size of a projected share, we now extend them with sufficient dummy random values to match the full share size. This ensures that the pseudoshares are indistinguishable from real shares.

Trace. The tracing procedure remains the same as in the projection traceable scheme of Section 5, but instead of querying the reconstruction box with projected pseudoshares, it now uses the full pseudoshares generated by the updated ShareGeneration algorithm.

Verify. The verification procedure remains the same as the one of the projection traceable scheme of Section 5.

We provide in Fig. 12 an outline of the scheme's construction. The full details of the algorithms are given in Fig. 13 and Fig. 14. The formal statement is presented in Theorem 6.5.

Theorem 6.5. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in [0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m^2$, the protocol Σ of Fig. 12 is a traceable secret sharing scheme with statistical reconstruction, traceability, and non-imputability realizing $\mathcal{F}$ with share size $O(2^n \mathrm{poly}(\log m))$ and $O(\mathrm{poly}(\lambda, n) \log m)$ queries.*

As a simple corollary of Theorem 6.5, we get our traceable scheme for general access structures.

Corollary 6.6 (Theorem 1.1 restated). *Let $\mathcal{F}$ be the family of all access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda =$*

$\mathsf{Share}(s, \Gamma)$:

1. Share the secret $s \in \mathbb{Z}_m^2$ among Γ with the general hiding anonymous scheme of Fig. 4.
2. Assign a uniformly random identifier $\mathsf{id}_i = (\mathsf{id}_i^1, \mathsf{id}_i^2) \in \mathbb{Z}_m^2$ to each party $i \in [n]$.
3. Append all the first parts of the identifiers id_i^1 in the tracing key.
4. Append all the second parts of the identifiers id_i^2 in the verification key.
5. For each $A \notin \Gamma$
   (a) Find a subset $B \notin \Gamma$ such that $A \cup B$ is a minimal authorized subset for some $\Gamma' \in \mathcal{F}$.
   (b) Consider $|B|$ dummy parties $d_1, ..., d_{|B|}$.
   (c) Select randomly a party $i \in A$.
   (d) For each bit $b \in \{0, 1\}$ of id_i
      i. Share s_b among $A \cup \{d_1, ..., d_{|B|}\}$ with the DNF additive scheme.
   (e) Ensure a uniform share size among all parties sending dummy random values to them.
   (f) Append the shares of the dummy parties $d_1, ..., d_{|B|}$ in the tracing key.
6. For each $A \in \Gamma$ such that there exists $\Gamma' \in \mathcal{F}$ for which $A \notin \Gamma'$
   (a) Ensure a uniform share size among all access structures sending dummy random values to the all the n parties.

$\mathsf{Proj}(\mathsf{sh}_i)$: The same as in the projection traceable scheme of Fig. 9.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i \in A}, |A|)$: The same as in the projection traceable scheme of Fig. 9.

$\mathsf{ShGen}(\mathsf{tk}, k, k', p)$:

1. Use as the k' pseudoshares the portions of the shares corresponding to the dummy parties $d_1, \ldots, d_{k'}$ that are obtained by secret sharing the p-th bit of the identifier with all unauthorized subsets of size k.
2. Ensure that all k' pseudoshares match the size of actual shares by padding dummy random values to them.

$\mathsf{Trace}^R(\mathsf{tk}, c, \lambda)$:

1. For each size $\ell \in [n - c]$.
   (a) For each bit $b \in \{0, 1\}$ of id_i
      i. Repeat poly(λ) times
         A. Compute ℓ pseudoshares using the ShGen algorithm.
         B. Query the reconstruction box R with these pseudoshares.
         C. Store the output bit b^*.
      ii. Take as the bit b the majority value among all the poly(λ) outputs b^*.
   (b) If the first part of the reconstructed identifier matches any first part of the identifiers in the tracing key, terminate and output the second part of the reconstructed identifier as the proof π.

$\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi)$:

1. The same as in the projection traceable scheme of Fig. 9.

Fig. 12: Outline of a traceable secret sharing scheme for any access structure Γ of a covering family of access structures $\mathcal{F}$.

$\Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$ there exists a traceable secret sharing scheme with statistical reconstruction, traceability, and non-imputability realizing $\mathcal{F}$ with share size $O(2^n \mathrm{poly}(\log m))$ and $O(\mathrm{poly}(\lambda, n) \log m)$ queries.

Proof. It suffices to combine Theorem 6.5 together with Lemma 6.3. □

6.3 Proof of Theorem 6.5

For clarity, we organize the proof of Theorem 6.5 as a sequence of propositions and lemmas. We begin by showing that the protocol described in Fig. 13 and Fig. 14 satisfies the properties of a hiding anonymous secret sharing scheme.

Proposition 6.7. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the protocol Σ of Fig. 13 and Fig. 14 is a hiding anonymous scheme with statistical privacy realizing $\mathcal{F}$ with share size $O(2^n \mathrm{poly}(\log m))$.*

Proof. We begin by proving the privacy of the scheme. Observe that the Share algorithm distributes the secret using the same mechanism as the hiding anonymous scheme described in Fig. 4. Moreover, the additional components of the shares either consist of uniformly random values or correspond to shares of fake secrets s_b that are used to recover bits b of the identifier. These fake secrets are shared using the anonymous additive scheme, and their unique authorized subset includes dummy parties whose shares are not available. Consequently, these shares are indistinguishable from random values and do not give any information about the secret. Therefore, the overall privacy of the scheme directly follows from the privacy guarantee of the construction in Fig. 4.

Regarding the statistical correctness of the scheme, note that the Projection and Reconstruction algorithms remain identical to those of the anonymous scheme in Fig. 3, with the only difference being the increased number of candidate share components that must be checked during reconstruction. However, this number is still bounded by $O(2^n \log m)$. More in detail, the construction uses $2 \log m$ independent copies of the base scheme, one for each bit of the identifiers, and in each copy every candidate share component corresponds to either an authorized or an unauthorized subset of an access structure in the family. In the worst case, this yields at most 2^{n+1} candidate shares per copy (two for each of the 2^n subsets), which results in the stated bound. Furthermore, as in the scheme of Fig. 4, for any authorized set of size k all except one of the candidate share components of level k are uniform random values because of the previous analysis on the Share algorithm. Consequently, by choosing a sufficiently large group size m, the statistical correctness follows by the same argument used in the proof of the anonymous scheme Fig. 3.

Next, perfect projected correctness and perfect anonymous reconstruction follow from the above observations on the Share, Projection, and Reconstruction algorithms, combined with the fact that the hiding anonymous scheme of Fig. 4 already satisfies both properties.

Notation:

1. Let $\overline{\Gamma} = 2^{[n]} \setminus \Gamma$ be the family of unauthorized subsets in Γ.
2. Let $\Delta = \bigcup_{\Gamma' \in \mathcal{F}} \Gamma'$ be the family of authorized subsets of access structures in $\mathcal{F}$.
3. Let Δ_k be the family of subsets in Δ of size k.
4. Let $\overline{\Delta} = \bigcup_{\Gamma' \in \mathcal{F}} \overline{\Gamma'}$ be the family of unauthorized subsets of access structures in $\mathcal{F}$.
5. Let $\overline{\Delta}_k$ be the family of subsets in $\overline{\Delta}$ of size k.
6. Let $\overline{\Delta}_{k,\ell}$ be the family of subsets in $\overline{\Delta}_k$ that require an unauthorized subset in $\overline{\Delta}$ of size ℓ to be a minimal authorized subset in some $\Gamma' \in F$.
7. Let $\Phi_t = \Delta_t \cup \bigcup_{k+\ell=t} \overline{\Delta}_{k,\ell}$.
8. Let P_t denote the set of permutations of $|\Phi_t|$ elements and P'_m the set of permutations of $2\log m$ elements.

$\mathsf{Share}(s, \Gamma)$:

1. Share $s \in \mathbb{Z}_m^2$ with the hiding anonymous scheme Σ of Fig. 4 among Γ, i.e. $\{\mathsf{sh}^s_{i,2\log m}\}_{i\in[n]} \leftarrow \Sigma(s, \Gamma)$.
2. Parse $\mathsf{sh}^s_{i,2\log m}$ as $\{\mathsf{sh}^s_{i,2\log m,k}\}_{k\in[n]}$ for $i \in [n]$.
3. Parse $\mathsf{sh}^s_{i,2\log m,k}$ as $\{\mathsf{sh}^*_{i,2\log m,k,A}\}_{A\in\Delta_k}$ for $i \in [n]$, $k \in [n]$.
4. For $q \in [2\log m - 1]$
   (a) For $k \in [n]$
      i. For $A \in \Delta_k$
         A. Send two random elements $r_0, r_s \in \mathbb{Z}_m^2$ to each of the parties in A, i.e. $\mathsf{sh}^s_{i,q,k,A} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $i \in A$.
      ii. Set $\mathsf{sh}^s_{i,q,k} \leftarrow \{\mathsf{sh}^s_{i,q,k,A}\}_{A\in\Delta_k}$ for $i \in [n]$.
   (b) Set $\mathsf{sh}^s_{i,q} \leftarrow \{\mathsf{sh}^s_{i,q,k}\}_{k\in[n]}$ for $i \in [n]$.
5. Select a permutation $\sigma'_m \in P'_m$ uniformly at random, i.e. $\sigma'_m \leftarrow_\$ P'_m$.
6. For $q \in [2\log m]$
   (a) Set $p \leftarrow \sigma'_m(q)$ and $\mathsf{sh}^s_{i,p} \leftarrow \mathsf{sh}^s_{i,q}$ for $i \in [n]$.
7. Set $\mathsf{id}_i = (\mathsf{id}_i^1, \mathsf{id}_i^2) \leftarrow_\$ \mathbb{Z}_m^2$ for $i \in [n]$.
8. Parse id_i^k as $\{\mathsf{id}^k_{i,j}\}_{j\in[m]}$ for $i \in [n], k \in [2]$.
9. For $k \in [n]$, for $\ell \in [n-k]$, and for $A \in \overline{\Delta}_{k,\ell}$
   (a) If $A \in \overline{\Gamma}$
      i. Let d_1,...,d_ℓ be ℓ extra dummy parties, and let $B = A \cup \{d_1, ..., d_\ell\}$.
      ii. Select a party $j \in A$ uniformly at random, i.e. $j \leftarrow_\$ A$.
      iii. For $q \in [2\log m]$
         1. Set $b \leftarrow \mathsf{id}^p_{j,q}$.
         2. Share s_b with the anonymous additive scheme Σ' of Fig. 1 among B, i.e. $\{\mathsf{sh}^*_{i,k+\ell,q,A}\}_{i\in B} \leftarrow \Sigma'(s_b, B)$.
         3. Set as the traceable key for the q bit of the identifier for the set A the shares of the ℓ dummy parties in B, i.e. $\mathsf{tk}^*_{q,A} \leftarrow \{\mathsf{sh}^*_{i,q,k+\ell,A}\}_{i\in B\setminus A}$.
         4. Send two random elements $r_0, r_s \in \mathbb{Z}_m^2$ to each of the parties not in A, i.e. $\mathsf{sh}^*_{i,q,k+\ell,A} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $i \in [n] \setminus A$.
      iv. Set $\mathsf{tk}^*_A \leftarrow \{\mathsf{tk}^*_{q,A}\}_{q\in[2\log m]}$ for $A \in \overline{\Gamma}$.
   (b) Otherwise, send $4\log m$ random elements $r_0^1, r_s^1, ..., r_0^{2\log m}, r_s^{2\log m} \in \mathbb{Z}_m^2$ to each of the n parties, i.e. $\mathsf{sh}^*_{i,q,k+\ell,A} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $i \in [n], q \in [2\log m]$.
10. For $t \in [n]$
   (a) Select a permutation $\sigma_t \in P_t$ uniformly at random, i.e. $\sigma_t \leftarrow_\$ P_t$.
   (b) For $A \in \Phi_t$
      i. Set $B \leftarrow \sigma_t(A)$ and $\mathsf{sh}_{i,q,t,B} \leftarrow \mathsf{sh}^*_{i,q,t,A}$ for $i \in [n], q \in [2\log m]$.
      ii. If $A \notin \Gamma$, set $\mathsf{tk}_B \leftarrow \mathsf{tk}^*_A$.
   (c) Set $\mathsf{sh}_{i,q,t} \leftarrow \{\mathsf{sh}_{i,q,t,A}\}_{A\in\Phi_t}$ for $i \in [n], q \in [2\log m]$.
   (d) Set $\mathsf{tk}_t \leftarrow (\mathsf{id}_t^1, \sigma_t, \{\mathsf{tk}_B\}_{B\in\mathcal{J}})$, where $\mathcal{J} = \{\sigma_t(A) : A \in \Phi_t \cap \overline{\Gamma}\}$.
11. Set $\mathsf{sh}_{i,q} \leftarrow \{\mathsf{sh}_{i,q,k}\}_{k\in[n]}$ for $i \in [n], q \in [2\log m]$.
12. Set $\mathsf{sh}_i \leftarrow \{\mathsf{sh}_i\}_{k\in[n]}$ for $i \in [n]$.
13. Set $\mathsf{tk} \leftarrow \{\mathsf{tk}_i\}_{i\in[n]}$.
14. Set $\mathsf{vk} \leftarrow \{\mathsf{id}_i^2\}_{i\in[n]}$.
15. Output $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$.

Fig. 13: Share algorithm of a traceable secret sharing scheme for any access structure Γ of a covering family of access structures $\mathcal{F}$.

$\mathsf{Proj}(\mathsf{sh}_i, k)$: The same as in the projection traceable scheme of Fig. 10.

$\mathsf{Rec}(\{\mathsf{psh}_{i,k}\}_{i\in A}, k)$: The same as in the projection traceable scheme of Fig. 11.

$\mathsf{ShGen}(\mathsf{tk}, k, k', p)$:

1. Parse tk as $\{\mathsf{tk}_i\}_{i\in[n]}$.
2. Parse tk_i as $(\mathsf{id}_i^1, \sigma_i, \{\mathsf{tk}_B\}_{B\in\mathcal{J}})$ for $i \in [n]$, where $\mathcal{J} = \{\sigma_i(A) : A \in \Phi_i \cap \overline{\Gamma}\}$.
3. Parse tk_B as $\{\mathsf{tk}_{q,B}\}_{q\in[2\log m]}$ for $B \in \{\sigma_{k+k'}(A) : A \in \overline{\Delta}_{k,k'} \cap \overline{\Gamma}\}$.
4. Parse $\mathsf{tk}_{q,B}$ as $\{\mathsf{sh}_{i,q,k+k',B}\}_{i\in[k']}$ for $q \in [2\log m]$.
5. For $i \in [k']$
   (a) For $q \in [2\log m]$
      i. For $j \in [n]$
         A. For $B \in \Phi_j$
            I. Set $A \leftarrow \sigma_i^{-1}(B)$.
            II. If $A \in \overline{\Delta}_{k,k'} \cap \overline{\Gamma}$ and $q = p$
               (1) Set $\mathsf{sh}'_{i,q,j,B} \leftarrow \mathsf{sh}_{i,q,j,B}$.
            III. Else
               (1) Send two random elements $r_0, r_B \in \mathbb{Z}_m^2$ to the party i, i.e.
               $$\mathsf{sh}'_{i,q,j,B} \leftarrow_{\$} (\mathbb{Z}_m^2, \mathbb{Z}_m^2).$$
         B. Set $\mathsf{sh}'_{i,q,j} \leftarrow \{\mathsf{sh}'_{i,q,j,B}\}_{B\in\Phi_j}$.
      ii. Set $\mathsf{sh}'_{i,q} \leftarrow \{\mathsf{sh}'_{i,q,j}\}_{j\in[n]}$.
   (b) Set $\mathsf{sh}'_i \leftarrow \{\mathsf{sh}'_{i,q}\}_{q\in[2\log m]}$.
6. Output $\{\mathsf{sh}'_i\}_{i\in[k']}$.

$\mathsf{Trace}^R(\mathsf{tk}, c, \lambda)$:

1. Set three empty lists L, L', L''.
2. For $\ell \in [n-c]$
   (a) For $q \in [2\log m]$
      i. For $h \in [\mathrm{poly}(\lambda)]$
         A. Set $\{\mathsf{sh}'_{i,h,\ell}\}_{i\in[\ell]} \leftarrow \mathsf{ShGen}(\mathsf{tk}, c, \ell, q)$.
         B. Query R on $\{\mathsf{sh}'_{i,\ell,q,h}\}_{i\in[\ell]}$. Let b_h be the response.
         C. Store b_h in L''.
      ii. Set $\mathsf{id}_{\ell,q}$ the majority value in L'' and erase L''.
   (b) Set $\mathsf{id}_{\ell,1} \leftarrow \{\mathsf{id}_{\ell,q}\}_{q\in[m]}$ and set $\mathsf{id}_{\ell,2} \leftarrow \{\mathsf{id}_{\ell,q}\}_{q\in[2\log m]\setminus[\log m]}$.
   (c) If $\mathsf{id}_{h,\ell}^1 = \mathsf{id}_i^1$ for some $i \in [n]$
      i. If $i \notin L$, add i to L.
      ii. If $\mathsf{id}_{h,\ell}^2 \notin L'$, add $\mathsf{id}_{h,\ell}^2$ to L'.
3. If $|L| \neq 1$, terminate and output $\perp$.
4. Set $\mathcal{I} \leftarrow \{i\}$, where i is the unique element in L.
5. Set $\pi \leftarrow L'$.
6. Output $(\mathcal{I}, \pi)$.

$\mathsf{Verify}(\mathsf{vk}, \mathcal{I}, \pi)$: The same as in the projection traceable scheme of Fig. 11.

Fig. 14: Projection, Reconstruction, ShareGeneration, Trace, and Verify algorithms of a traceable secret sharing scheme for any access structure Γ of a covering family of access structures $\mathcal{F}$.

Regarding the perfect hideout requirement, it suffices to argue that the secret is shared using the hiding anonymous scheme of Fig. 3 and that then sufficiently many dummy random values are padded to guarantee that the shares are of a uniform size across all access structures in the family even after sharing the party's identifiers.

Finally, for the share size it suffices to notice that in each copy of the scheme, each party receives at most two share components of $\log m$ bits for each subset (authorized or not) of $2^{[n]}$. Since there are $2\log m$ copies of the scheme, the total share size per party is $O(2^n \log^2 m)$. □

Remark 6.8. Indeed, the proof of Proposition 6.7 shows that the scheme of Theorem 6.5 satisfies the strongest notion of share anonymity, that is, *uniform anonymity*.

Next, we prove that the scheme satisfies perfect tracer privacy.

Proposition 6.9. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the hiding anonymous scheme Σ of Fig. 13 and Fig. 14 realizing $\mathcal{F}$ has perfect tracer privacy.*

Proof. To prove the perfect tracer privacy requirement, it suffices to combine the privacy requirement proved in Proposition 6.7 with the fact that the tracing key only contains the share components corresponding to fake secrets s_b that are used to recover bits b of the identifier. Since these are generated independently of the secret, they reveal no additional information about it. Consequently, the perfect tracer privacy requirement holds. □

Prior to establishing statistical traceability and non-imputability, we present a technical lemma demonstrating that the scheme ensures perfect minimal set indistinguishability. This property will be fundamental in the proof of the scheme's traceability.

Lemma 6.10. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the hiding anonymous scheme Σ of Fig. 13 and Fig. 14 realizing $\mathcal{F}$ has perfect minimal set indistinguishability.*

Proof. Let $p \in [2\log m]$, let A, B be two unauthorized subsets in any two access structures $\Gamma, \Gamma' \in \mathcal{F}$, with $k = |A|$, $k' = |B|$, and $A \cup B$ being a minimal authorized subset in Γ as described in the statement of Definition 4.13, and let $d_1, \ldots, d_{n-1}$ denote the additional $n-1$ dummy parties used in the Share algorithm.

First, by the privacy and reconstruction analysis provided in the proof of Proposition 6.7, all share components in $\{\mathsf{sh}_i\}_{i \in A \cup B}$ behave as uniformly random values, except for the unique share component corresponding to the minimal authorized subset $A \cup B$, which consists of valid shares of the secret s.

Similarly, by the construction of the Share and $\mathsf{ShareGeneration}$ algorithms, an analogous argument shows that all share components in $\{\mathsf{sh}'_i\}_{i\in A} \cup \{\mathsf{sh}''_i\}$ are also uniformly random, except for the share component corresponding to the subset $A \cup \{d_1, \ldots, d_{k'}\}$, which consists of shares of a fake secret s_b that are used to recover the p-th bit b of the identifier.

Since both s and s_b are sampled uniformly at random from $\mathbb{Z}_m^2$, it follows that the two distributions in the definition of minimal set indistinguishability are identical.

Indeed, both distributions can be viewed as being generated by the same Share procedure, where the secret s is replaced by the fake secret s_b, and the parties in B are replaced by the dummy parties $d_1, \ldots, d_{k'}$. This concludes the proof of the perfect projection indistinguishability requirement. □

After proving the lemma, we employ it to establish that the scheme achieves efficient statistical traceability.

Proposition 6.11. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in [0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m^2$, the hiding anonymous scheme Σ of Fig. 13 and Fig. 14 realizing Γ has statistical traceability and requires $O(\mathrm{poly}(\lambda, n) \log m)$ queries.*

Proof. Let $\mathcal{A}$ be any adversary participating in the $\mathsf{Adv}^{\mathrm{trac}}_{\mathcal{A},\Sigma,\epsilon}$ security experiment, let A be the unauthorized subset of corrupted parties and s_0, s_1 be the secrets selected by $\mathcal{A}$ at the beginning of the experiment with $|A| = c$ (Fig. 7, Step 1), let $\Gamma \in \mathcal{F}$ be the selected access structure (Fig. 7, Step 2), let $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in A}$ be the shares received by $\mathcal{A}$ from the challenger (Fig. 7, Step 6), and let tk be the tracing key received by the tracer from the challenger (Fig. 7, Step 4). Let id_{i^*} denote the identifier corresponding to the set A (Fig. 12, Step 5.(c) of Share), let $k' \in [n-c]$ be such that $A \in \overline{\Delta}_{c,k'}$, and let $B \subseteq [n]$ be the unauthorized subset of size k' such that $A \cup B$ is a minimal authorized subset in some $\Gamma' \in F$.[12]

Let G be the event in which $\mathcal{A}$ outputs a reconstruction box R that is $(\mathcal{F}, A, \mathbf{sh}, s_0, s_1, \epsilon(\lambda))$-good (Fig. 7, Step 8). Conditioned on $\neg\mathsf{G}$, the experiment outputs 0 with probability 1, so we condition the rest of the analysis on G.

For each $\ell \in [n-c]$, $q \in [2\log m]$, let $\mathsf{id}_{\ell,q}$ be the majority bit obtained from the $\mathrm{poly}(\lambda)$ queries to R at the (ℓ, q)-th iteration, and let $\mathsf{id}_\ell = (\mathsf{id}^1_\ell, \mathsf{id}^2_\ell)$ be the identifier constructed by padding all these bits. We define the following events:

- Let E_1 be the event in which $\mathsf{id}_{k',q} \neq \mathsf{id}_{i^*}$ for some $q \in [2\log m]$, i.e., any of the output bits at the correct level does not match that of id_{i^*}.
- For $\ell \in [n-c]$, let E^ℓ_2 denote the event in which there exists $i \in [n] \setminus \{i^*\}$ such that $\mathsf{id}^1_{h,\ell} = \mathsf{id}^1_i$, i.e., the first component of the constructed identifier at level ℓ matches that of a party different from i^*.
- Let E_3 be the event in which $\mathsf{id}^1_i = \mathsf{id}^1_{i'}$ for some distinct $i, i' \in [n]$.

[12] Such a subset B exists by assumption, since $\mathcal{F}$ is a covering family.

Clearly, conditioned on G, the adversary wins the experiment (i.e., the output is 1) only if one of the events above occurs. We now bound the probability of each of these events.

We start by bounding E_1. For each $q \in [2 \log m]$ and $h \in [\mathrm{poly}(\lambda)]$, let $b_{k',q,h}$ be the output of R to the (k', q, h)-th query. By assumption G holds, so:

$$\Pr[b = R(\{\mathsf{sh}'_i\}_{i \in B})] \geq \frac{1}{2} + \epsilon(\lambda),$$

where $\{\mathsf{sh}'_i\}_{i \in B}$ correspond to shares of s_b according to Γ' conditioned on $\{\mathsf{sh}'_i\}_{i \in A} = \mathbf{sh}$. Then, by perfect minimal set indistinguishability (Lemma 6.10), we have:

$$\Pr[b_{k',q,h} = b] \geq \frac{1}{2} + \epsilon(\lambda)$$

for each $q \in [2 \log m]$ and $h \in [\mathrm{poly}(\lambda)]$.

From there, since R is stateless and queries are independent, by taking $\mathrm{poly}(\lambda)$ sufficiently large and using the Chernoff bound, we get that the probability that $\mathsf{id}_{k',q}$ does not match the q-th bit of id_{i^*} is negligible for each $q \in [2 \log m]$.

Therefore, since $2 \log m = O(\mathrm{poly}(\lambda))$ we conclude that $\Pr[\mathsf{E}_1]$ is negligible in λ.

Next, we move to bound E_2^ℓ for $\ell \in [n - c]$. By the construction of the scheme, all fake secrets that allow to recover the identifiers are shared with the anonymous additive scheme of Fig. 1 and the reconstruction box is only able to recover the corresponding identifier for the queries of level k'. At this level, the shared identifier is precisely id_{i^*}, so these queries have no knowledge of any other identifier. For the remaining levels $\ell \in [n - c] \setminus \{k'\}$, the privacy property of the scheme implies that the output of R on these queries reveals no information about id_{i^*}. Thus, since id_{i^*} is sampled uniformly from a super-polynomial domain, we conclude that $\Pr[\mathsf{E}_2^\ell]$ is negligible in λ.

Now, we bound E_3. Since the identifiers $\{\mathsf{id}_i^1\}_{i \in [n]}$ are independently sampled from a domain of super-polynomial size, the probability that any two parties share them is negligible in λ.

Finally, we have proved that each of the events under which the adversary could win the experiment occurs with negligible probability in λ. Therefore, by the union bound, the overall success probability of $\mathcal{A}$ in the $\mathsf{Adv}^{\mathrm{trac}}_{\mathcal{A},\Sigma,\epsilon}$ experiment is negligible in λ, as desired. □

We conclude the proof of Theorem 6.5 by proving that the scheme satisfies statistical non-imputability.

Proposition 6.12. *Let $\mathcal{F}$ be a covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, and let $m = 2^{\Theta(\lambda)}$. For any $s \in \mathbb{Z}_m^2$, the hiding anonymous scheme Σ of Fig. 13 and Fig. 14 realizing $\mathcal{F}$ has statistical non-imputability.*

Proof. Let $\mathcal{A}$ be any adversary participating in the $\mathsf{Adv}^{\mathrm{ni}}_{\mathcal{A},\Sigma}$ security experiment, let $i^* \in [n]$ be the honest party selected by $\mathcal{A}$ at the beginning of the experiment (Fig. 8, Step 1), and let $\mathsf{id}_{i^*} = (\mathsf{id}_{i^*}^1, \mathsf{id}_{i^*}^2)$ denote the identifier of the party

i^* (Fig. 9, Step 2 of Share). Let $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ be the shares received by $\mathcal{A}$ from the challenger (Fig. 8, Step 3).

Let E_{i^*} be the event in which $\mathcal{A}$ learns $\mathsf{id}^2_{i^*}$. Clearly, by construction, the experiment $\mathsf{Adv}^{\mathrm{ni}}_{\mathcal{A},\Sigma}$ outputs 1 only if E_{i^*} occurs. We now bound the probability of this event.

By construction of the Share algorithm, all components of $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ that relate to fake secrets that allow to recover the identifier id_{i^*} correspond to unauthorized subsets because id_{i^*} is only shared using the anonymous additive scheme of Fig. 1 with subsets that include i^*. Furthermore, all other components of $\{\mathsf{sh}_i\}_{i\in[n]\setminus\{i^*\}}$ are generated independently of id_{i^*} and thus carry no information about it. Therefore, by the privacy property of the anonymous additive scheme from Fig. 1, the adversary $\mathcal{A}$ obtains no information about the value $\mathsf{id}^2_{i^*}$.

Combining this with the fact that $\mathsf{id}^2_{i^*}$ is sampled uniformly from a super-polynomial domain, we conclude that the probability of event E_{i^*} is negligible in λ, as desired.

□

6.4 Further Optimizations

As in the projection traceability setting, we present additional optimizations of the scheme from Theorem 6.5. First, we show how to reduce the share size for some access structures by employing the anonymous threshold scheme introduced in Section 3.2. Then, we extend the general scheme to the setting where the tracer does not know the number of corrupted parties that contributed to the reconstruction box. Later, we provide a modification that allows tracing multiple corrupted parties simultaneously. Finally, we describe how to modify our scheme so that even a malicious verifier, or a malicious party holding both the tracing and verification keys, cannot falsely accuse an honest party.

Efficient Schemes for Low Slices. The following remark describes how to optimize our construction in order to obtain an efficient traceable scheme for low slices.

Remark 6.13. The construction described above can be optimized by incorporating the anonymous threshold scheme from Fig. 2. Specifically, if $\mathcal{F}$ is a covering family of access structures in which all unauthorized subsets have size at most $t-1$, then the share size can be reduced by replacing the scheme used to share the secret in Step 1 of the Share algorithm in Fig. 13 with the optimized construction from Remark 3.13.[13] This optimization, however, also requires adjusting the sharing procedure for identifiers corresponding to unauthorized subsets in $\overline{\Delta}_{k,\ell} \cap \overline{\Gamma}$ with $k + \ell = t$, to ensure compatibility with the scheme used to share the secret.

[13] A minor modification of the scheme is also required to ensure it satisfies the hiding property, but we omit these details.

Remark 6.13 leads to efficient traceable schemes with polynomial share size for a specific family of access structures: low slices. That is, access structures over n parties given by t-slices in which $t \ll n$.

Corollary 6.14 (Corollary 6.14 restated). *Let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\mathrm{poly}(n))$, let $q = \lambda^{\omega(1)}$ be a prime power, let $t \in [n]$ with $t = O(1)$, and let $\mathcal{F}$ be the family of t-slice access structures over n parties. For any $s \in \mathbb{F}_q$ there exists a traceable secret sharing scheme with statistical reconstruction, traceability and non-imputability realizing $\mathcal{F}$ with share size $O(\mathrm{poly}(\lambda, \log q))$ and $O(\mathrm{poly}(\lambda, \log q))$ queries.*

Proof. The proof follows the same argument as that of Corollary 5.9. □

Learning the Number of Corrupted Parties. The following remark describes how our scheme can be adapted to support traceability even when the number of corrupted parties involved in constructing the reconstruction box is unknown.

Remark 6.15. To handle the case where the number of corrupted parties c is not known in advance, we follow the approach as in Remark 6.15: we invoke the Trace algorithm with successive values of $c = n-1, n-2, \ldots$ until it succeeds.

The correctness of Remark 6.15 follows the same reasoning as that of Remark 5.10.

Tracing Multiple Corrupted Parties Simultaneously. In the traitor tracing literature [CFNP00,GSS21,BPR24], identifying a single corrupted party is typically considered sufficient, as this discourages others from misbehaving. However, our scheme can be adapted to trace multiple corrupted parties (potentially all of them), at the cost of increasing the size of the shares and tracing and the running time of the tracing procedure by a linear factor.

More precisely, to enable tracing of all corrupted parties rather than just one, we modify the sharing procedure as follows. Instead of sharing the bitwise encoding of one random identifier per unauthorized subset, we do it independently for all parties in the subset. Concretely, for every unauthorized subset of size k, we run k independent instances of the base scheme in parallel, one for each party, where the j-th instance is used to embed the identifier of the j-th party. This creates n parallel instances of the base scheme, allowing the tracer to identify all corrupted parties by running the original tracing procedure at most n times, one per party.

Stronger Non-imputability. As defined in Definition 4.11, the adversary in our non-imputability experiment is given access only to the tracing key and not to the verification key. As discussed in Remark 4.6, this restriction is inherent to the information-theoretic setting. We now explain why this model is still meaningful in practice, and we show how, by moving to the computational setting, our

scheme can be adapted to achieve a stronger notion of *public non-imputability*, in which the verification key is public and the verification procedure can be carried out by any party.

We consider this model reasonable because the verifier is assumed to be a trusted and independent authority whose role is limited to validating the proof produced by the tracer, rather than performing the tracing itself or participating in the access structure. Accordingly, the verifier does not hold the tracing key, which is required to carry out the tracing procedure, nor any shares of the scheme, and therefore cannot act as either a (honest or malicious) tracer or a party. In practice, this captures a natural separation of roles: for instance, the police may act as the tracer, while an independent judge, who is not involved with the parties, serves as the verifier and checks the validity of the evidence presented by the tracer.

Nevertheless, a slight modification of our scheme yields *public non-imputability* at the cost of assuming the existence of one-way functions. Concretely, we adopt and slightly adapt the non-imputability technique of [BPR24]. Instead of using the second component of each party's identifier directly as the verification key, we proceed as follows: for each party, we sample an additional random value, which is included in the tracing key. The verification key then consists of the images, under a one-way function, of the XOR of this random value with the second component of the party's identifier. Now, the proof output by the tracer is the XOR of the random value and the second component of the recovered identifier.

Intuitively, a malicious party that has access to the public verification key would now need to invert the one-way function in order to forge a valid proof, which is infeasible under the assumption that one-way functions are secure. Moreover, corrupted parties that build the reconstruction box cannot use the verification key to detect tracing attempts, since they do not know the random value that is XORed with their identifier. At the same time, an honest verifier can efficiently validate a proof by simply checking that applying the one-way function to it matches the corresponding verification key entry. We refer to [BPR24] for a detailed security analysis, as the arguments carry over to our setting with only minor adaptations.

Acknowledgments

We thank Yuval Ishai for providing us with early access to [BGI+25] and for valuable discussions, and Eurocrypt reviewers for their suggestions. The authors are supported by the projects ACITHEC PID2021-124928NB-I00 and MATSE PID2024-156636NB-C22 funded by MCIN/AEI/10.13039/501100011033, and by the project HERMES, funded by the European Union NextGenerationEU/PRTR via INCIBE.

References

ABF+19. Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 441–471. Springer-Verlag, 2019.

ABNP20. Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. In *STOC 2020*, pages 280–293, 2020.

AN21. Benny Applebaum and Oded Nir. Upslices, downslices, and secret-sharing with complexity of 1.5^n. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021*, volume 12827 of *LNCS*, pages 627–655. Springer, 2021.

Bei25. Amos Beimel. Secret-sharing schemes for general access structures: An introduction. Cryptology ePrint Archive, Paper 2025/518, 2025.

BEMS25. Karim Baghery, Ehsan Ebrahimi, Omid Mirzamohammadi, and Mahdi Sedaghat. Traceable verifiable secret sharing and applications. Cryptology ePrint Archive, Paper 2025/318, 2025.

BGI+25. Allison Bishop, Matthew Green, Yuval Ishai, Abhishek Jain, and Paul Lou. Fully anonymous secret sharing. In *CRYPTO 2025*. Springer International Publishing, 2025.

BL88. Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shaffi Goldwasser, editor, *CRYPTO '88*, volume 403 of *LNCS*, pages 27–35. Springer-Verlag, 1988.

Bla79. George Robert Blakley. Safeguarding cryptographic keys. In *Proc. of the 1979 AFIPS National Computer Conference*, volume 48, pages 313–317, 1979.

BPR24. Dan Boneh, Aditi Partap, and Lior Rotem. Traceable secret sharing: Strong security and efficient constructions. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024*, pages 221–256, Cham, 2024. Springer Nature Switzerland.

BPR25. Dan Boneh, Aditi Partap, and Lior Rotem. Traceable verifiable random functions. In *EUROCRYPT*. Springer International Publishing, 2025.

BS97. Carlo Blundo and D. R. Stinson. Anonymous secret sharing schemes. *Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 77:13–28, 1997.

CDK+25. Ran Canetti, Ivan Damgård, Sebastian Kolby, Divya Ravi, and Sophia Yakoubov. Deniable secret sharing. Cryptology ePrint Archive, Paper 2025/525, 2025.

CFNP00. Benny Chor, Amos Fiat, M. Naor, and Benny Pinkas. Tracing traitors. *IEEE Trans. Information Theory*, 46(3):893–910, 2000.

Con25. Roni Con. Anonymous shamir's secret sharing via reed-solomon codes against permutations, insertions, and deletions. In *ISIT*, 2025.

Csi96. László Csirmaz. The dealer's random bits in perfect secret sharing schemes. *Studia Sci. Math. Hungar.*, 32(3–4):429–437, 1996.

Csi97. László Csirmaz. The size of a share must be large. *J. of Cryptology*, 10(4):223–231, 1997.

DFLM24. Stefan Dziembowski, Sebastian Faust, Tomasz Lizurej, and Marcin Mielniczuk. Secret sharing with snitching. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 840–853, New York, NY, USA, 2024. Association for Computing Machinery.

EBG+24. Harry Eldridge, Gabrielle Beck, Matthew Green, Nadia Heninger, and Abhishek Jain. Abuse-resistant location tracking: balancing privacy and safety in the offline finding ecosystem. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA, 2024. USENIX Association.

FG26. Oriol Farràs and Miquel Guiot. Traceable secret sharing schemes for general access structures. To appear in Eurocrypt 2026, 2026.

GJP25. Vipul Goyal, Abhishek Jain, and Aditi Partap. Traceable secret sharing revisited. Cryptology ePrint Archive, Paper 2025/1980, 2025.

GL89. O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. STOC '89, page 25–32, New York, NY, USA, 1989. Association for Computing Machinery.

GMO03. Mida Guillermo, Keith Martin, and Christine O'Keefe. Providing anonymity in unconditionally secure secret sharing schemes. *Des. Codes Cryptography*, 28:227–245, 04 2003.

GSS21. Vipul Goyal, Yifan Song, and Akshayaram Srinivasan. Traceable secret sharing and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021*, pages 718–747, Cham, 2021. Springer International Publishing.

Hof24. Charlotte Hoffmann. Traceable secret sharing based on the chinese remainder theorem. Cryptology ePrint Archive, Paper 2024/811, 2024.

ISN87. Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structure. In *Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1), 15-20, 1993.

LV18. Tianren Liu and Vinod Vaikuntanathan. Breaking the circuit-size barrier in secret sharing. In *50th STOC*, pages 699–708, 2018.

LVW17. Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017*, volume 10401 of *LNCS*, pages 758–790. Springer-Verlag, 2017.

LVW18. Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 567–596. Springer-Verlag, 2018.

Mig83. Maurice Mignotte. How to share a secret? In Thomas Beth, editor, *Cryptography*, pages 371–375, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

PCO20. Anat Paskin-Cherniavsky and Ruxandra F. Olimid. On cryptographic anonymity and unpredictability in secret sharing. *Information Processing Letters*, 161:105965, 2020.

PP92. S. J. Phillips and N. C. Phillips. Strongly ideal secret sharing schemes. *J. Cryptology*, 5(3):185–191, 1992.

Sha79. Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

SV88. D. R. Stinson and S. A. Vanstone. A combinatorial approach to threshold schemes. *SIAM J. Discrete Math.*, 1(2):230–236, 1988.