Information-theoretic Strong Traceable Secret Sharing Schemes

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. More precisely, 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. Recently, two independent works extended traceable secret sharing to general access structures.

Goyal, Jain, and Partap [EC'26] introduced a model in which R is augmented with a label $I \subseteq [n]$ and is only required to distinguish between two secrets when queried with the shares of parties in I. Based on how this label relates to the corrupted set J that built the box, they defined two notions of traceability. If $I \cap J = \emptyset$, the model is called $\emptyset$-*strong traceability*, for which they presented a construction based on indistinguishability obfuscation (iO). Otherwise, their model is called *strong traceability*, for which they proved an impossibility result. Farràs and Guiot [EC'26] proposed a different model, which we call *hidden traceability*, where R has no label and the access structure is hidden from the parties.

In this work, we improve traceable secret sharing for general access structures in three directions. First, we present a fully information-theoretic scheme for the $\emptyset$-strong traceability model, eliminating the need for strong cryptographic assumptions. This resolves an open question posed by Goyal, Jain, and Partap, who asked what are the minimal assumptions needed in the $\emptyset$-strong traceability model.

Second, motivated by the impossibility of strong traceability, we introduce a relaxed notion called *hidden mildly strong traceability*. This model is relevant in practice and bridges the strong and hidden models. For this setting, we present an information-theoretic scheme for general access structures.

Finally, we consider the more general model of *stateful traceability*, where reconstruction boxes may keep state across queries, and we prove an impossibility result for this setting.

Keywords: Secret sharing scheme · traceability · information-theoretic security

Table of Contents

1 Introduction **3**
- 1.1 Our Results . . . 4
- 1.2 Our Techniques . . . 6
- 1.3 Related Work . . . 11
- 1.4 Open Questions . . . 12
- 1.5 Organization . . . 13

2 Preliminaries **14**
- 2.1 Notation . . . 14
- 2.2 Secret Sharing Schemes . . . 14
- 2.3 Hiding Secret Sharing . . . 15
- 2.4 Traceable Secret Sharing Schemes . . . 16
- 2.5 Strong Traceable Secret Sharing Schemes . . . 18
- 2.6 Hidden traceable Secret Sharing Schemes . . . 20

3 An Information-theoretic $\emptyset$-Strong Traceable Scheme **23**
- 3.1 An Overview of the Algorithms . . . 23
- 3.2 The Scheme . . . 27
- 3.3 Proof of Theorem 3.1 . . . 27

4 An Information-theoretic Hiding Mildly Strong Traceable Scheme **37**
- 4.1 The Mildly Strong Traceability Model . . . 37
- 4.2 The Hidden Mildly Strong Traceability Model. . . . 40
- 4.3 Minimal Covering Families of Access Structures . . . 42
- 4.4 An Overview of the Algorithms . . . 42
- 4.5 The Scheme . . . 45
- 4.6 Proof of Theorem 4.12 . . . 45
- 4.7 Further Optimizations . . . 53

5 Impossibility of Stateful Tracing **54**
- 5.1 Definitions . . . 54
- 5.2 Reduction to a Single Query Reconstruction Box . . . 56
- 5.3 An Impossibility Result . . . 57

1 Introduction

A secret sharing scheme is a cryptographic primitive in which a dealer distributes a secret among a set of parties so that only specific subsets, called authorized, can reconstruct it. The family of authorized subsets defines the access structure of the scheme, and its key efficiency measure is the size of the shares given to the parties.

Secret sharing was independently introduced by Shamir [Sha79] and Blakley [Bla79] in 1979. Both constructions considered threshold access structures, where any subset of parties can recover the secret as long as its size exceeds a fixed threshold. Since then, secret sharing has become a standard tool in cryptography, with applications such as secure multiparty computation and threshold cryptography. While threshold schemes cover many of these use cases, applications including weighted cryptography and attribute-based encryption require more expressive access structures.

This motivated the study of secret sharing for general access structures, i.e. for any monotone collection of subsets of parties. Such schemes were first proposed by Ito, Saito, and Nishizeki [ISN87], and later work focused on improving their efficiency [BL88,LV18,ABF+19,ABNP20,AN21,Nir26]. Despite this progress, the best known information-theoretic constructions for general access structures still require shares of exponential size, in the worst case.

Secret sharing schemes provide privacy guarantees against unauthorized subsets, but, by default, do not address other forms of dishonest behavior. In particular, they do not prevent parties from copying or selling their shares, which can be problematic in settings where such behavior is beneficial and goes unpunished.

To circumvent this limitation, Goyal, Song, and Srinivasan [GSS21], and later Boneh, Partap, and Rotem [BPR24], introduced traceable secret sharing schemes. In those schemes, it is assumed that an adversary uses the shares of an unauthorized subset to build a reconstruction box that can recover the secret when given enough valid shares. This box is assumed to be obtained by the authorities, who interact with it in a black-box manner to trace it back to its creators. In this context, a traceable secret sharing scheme must guarantee *traceability*, meaning that at least one party involved in building the box can be identified, and *non-imputability*, which ensures that honest parties are not falsely accused, while keeping the secret hidden during the tracing process. To support the tracing procedure, the sharing 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 linking the reconstruction box to one or more corrupted parties. The verification key is given to a verifier and allows checking the validity of the proof produced by the tracer.

Recently, two independent works extended traceable secret sharing to general access structures. On the one hand, Goyal, Jain, and Partap [GJP26] introduced a model for general access structures in which a reconstruction box is augmented with a label $I \subseteq [n]$. The box is only required to distinguish between two possible secrets when queried with all the shares of parties in I. Based on how this label relates to the corrupted set J that built the box, they defined two trace-

ability notions. If $I \cap J = \emptyset$, the model is called $\emptyset$-*strong traceability*, for which they presented an efficient construction for general access structures relying on indistinguishability obfuscation (iO). If $I \cap J \neq \emptyset$, the model is called *strong traceability*, for which they showed an impossibility result.

On the other hand, Farràs and Guiot [FG26] proposed a different generalization in which the reconstruction box has no label. Instead, the box is required to distinguish between two fixed secrets when queried with the shares of any subset of parties that, together with the ones that built the box, form an authorized set. This model, which we refer to as *hidden traceability*, has the additional feature that the access structure of the scheme is hidden from the parties. Rather than knowing the exact access structure, parties are only given a family of access structures that contains it. Within this setting, they presented an information-theoretic traceable secret sharing scheme.

Although the two generalizations are close in spirit, they differ in key aspects. This paper builds on both approaches with the aim of improving them and providing a common framework that makes their connections explicit.

1.1 Our Results

This work improves traceable secret sharing schemes for general access structures in three main directions. First, we provide a fully information-theoretic scheme for the $\emptyset$-strong traceability model, eliminating the need to rely on strong cryptographic assumptions. Then, we introduce a relaxation of strong traceability, called *hidden mildly strong traceability*, that is relevant in practice and bridges the strong traceability model of [GJP26] and the hidden traceability model of [FG26]. For this model, we provide an information-theoretic scheme for general access structures. Finally, we study the more general setting of *stateful tracing*, where reconstruction boxes can store information about previous queries, and we prove an impossibility result for this setting.

All our contributions, previous work on traceable secret sharing for general access structures, and their relations are presented in Fig. 1.

Information-theoretic $\emptyset$-Strong Traceable Schemes. Our first contribution is the construction of an information-theoretic scheme for general access structures in the $\emptyset$-strong traceability model. Unlike the previous construction in this model [GJP26], our scheme does not rely on iO or any other cryptographic assumption, demonstrating that no computational assumption is needed. This resolves an open question posed in [GJP26] regarding the assumptions required to build schemes in this tracing model. We prove the following result.

Theorem 1.1 (Informal). *For any access structure Γ over n parties, there exists a traceable secret sharing scheme with perfect privacy, statistical correctness, statistical $\emptyset$-strong traceability, and statistical strong non-imputability. The share size is linear in the DNF size of Γ, and tracing requires a number of queries that is polynomial in n.*

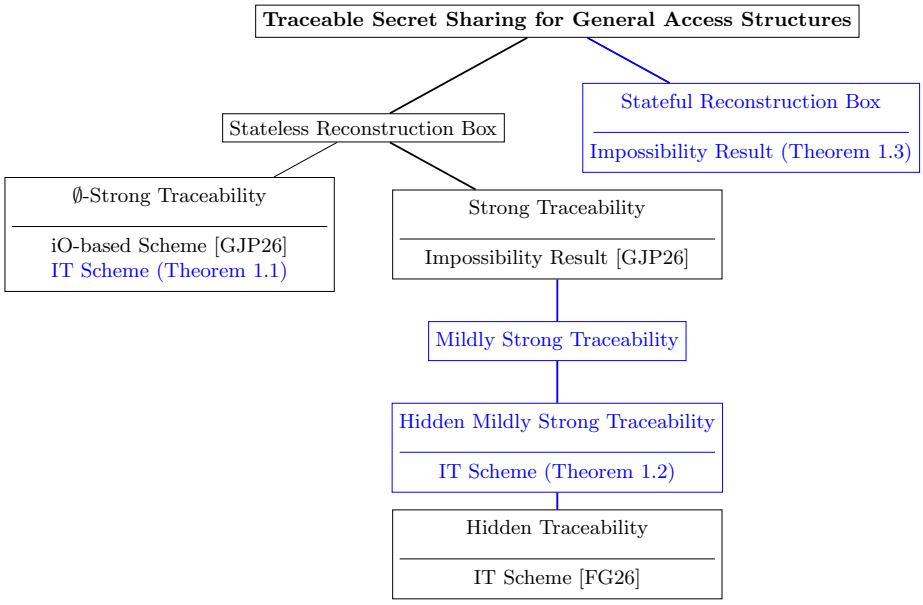

Fig. 1: Taxonomy of traceable secret sharing schemes for general access structures. The contributions of this work are highlighted in blue.

Indeed, our construction can be viewed as a compiler: starting from the DNF-based information-theoretic secret sharing scheme for general access structures, it upgrades it with traceability at the cost of a polynomial increase in the share size. Moreover, the size of the resulting tracing key is comparable to that of the shares.

Information-theoretic Hidden Mildly Strong Traceable Schemes. Motivated by the impossibility result for the strong traceability model [GJP26], we introduce a relaxed version, which we call *mildly strong traceability*, that circumvents this limitation. In this relaxation, the reconstruction box is required to recover the secret not only when given all the shares of the parties in I, but also when given shares from any subset $I' \subseteq I$ that, together with the corrupted parties, forms an authorized set. This model is relevant in practice, as it allows adversaries to build a reconstruction box that works for multiple potential buyers (all authorized subsets $I' \subseteq I$), rather than being tailored to a single specific buyer (the full subset I). Further details are provided in Section 4.1.

Then, we point out an inherent limitation of the mildly strong model: traditional schemes cannot provide any efficient tracing strategy. To address this, we take advantage of the hiding technique of [FG26], which motivates the definition of the *hidden mildly strong model*. This model is identical to the mildly strong one, except that now the access structure is hidden from the parties. For this

model, we present an information-theoretic traceable scheme for general access structures, stated below.

Theorem 1.2 (Informal). *For any access structure over n parties, there exists a traceable secret sharing scheme with perfect privacy, statistical correctness, statistical hidden mildly strong traceability, and statistical hidden non-imputability. Tracing requires a number of queries that is polynomial in n.*

Impossibility of Stateful Tracing. Finally, we consider the more general setting in which the reconstruction box built by the corrupted parties is stateful, meaning it can maintain internal state and adapt its behavior based on the sequence of input queries. In this context, we define the *stateful traceability model*, which captures the weakest meaningful notion of stateful reconstruction boxes: boxes labeled by I that are required to output the full secret for any input query containing shares of a subset $I' \subseteq I$ that, together with the corrupted parties, forms an authorized set. For this model, we prove the following impossibility result.

Theorem 1.3 (Informal). *There exist access structures for which there is no traceable secret sharing scheme achieving stateful traceability and non-imputability.*

1.2 Our Techniques

In this section, we outline the main ideas behind our results. We first explain how traceability can be achieved in the $\emptyset$-strong model without relying on cryptographic assumptions. We then discuss the limitations of the mildly strong traceability model and show how they can be addressed by hiding the access structure. Finally, we sketch the arguments that lead to our impossibility result for stateful tracing.

Information-theoretic $\emptyset$-Strong Traceable Schemes. Our tracing strategy is similar to that of Goyal, Jain, and Partap [GJP26], but replaces the use of iO with information-theoretic techniques. We begin by briefly recalling their construction and the main challenges faced, and then explain how we adapt it to the information-theoretic setting.

The tracing strategy of [GJP26] is based on observing *jumps* in the behavior of the reconstruction box for distinct input queries. First, the tracer queries the reconstruction box with crafted shares for the label I that reconstruct to a fake secret s^*. Then, the tracer repeatedly queries the box with carefully chosen inputs that selectively *disable* parties outside I in a way that is hidden from the box. In this context, disabling a party j means that the tracing input shares are set so that the share of the party j cannot contribute to the reconstruction. As the tracer gradually disables parties outside I one by one, there must be a point where the behavior of the box changes noticeably. This change, the so called jump, indicates that the disabled party was necessary for the box to work, and therefore belongs to the corrupted subset of parties that built it.

Once a corrupt party is identified, the tracer must still produce a verifiable proof of guilt. To this end, each party is assigned a random hidden identifier. The tracer then prepares inputs that selectively disable the corrupted party depending on the value of individual bits of its identifier. By repeatedly applying the same jump-based tracing strategy with these tailored inputs, the tracer can extract the identifier bit by bit, which can be used as a proof against that corrupted party.

Realizing this tracing strategy poses three main challenges. First, the tracer must be able to generate fake shares that reconstruct to a tracer-chosen fake secret. Second, the tracer must be able to modify the input shares so as to selectively disable parties one by one. Third, the tracer must be able to craft input shares that conditionally disable a party based on its identifier. To overcome these obstacles, Goyal, Jain, and Partap [GJP26] rely on indistinguishability obfuscation. Specifically, they design a modified reconstruction algorithm that natively supports these additional functionalities, obfuscate it using iO, and ensure that any reconstruction box must invoke this obfuscated procedure to recover the secret. We refer to [GJP26] for further details in their construction.

We next show how information-theoretic techniques can be used to overcome these challenges. In particular, our construction relies on the additive scheme for the DNF representation of an access structure. Recall that this scheme works by additively sharing the secret for each minimal authorized subset of the access structure. As we will see, this scheme possesses several properties that are particularly useful for our purposes.

Generating Fake Shares. First, we show how to enable the tracer to craft shares that reconstruct to a fake secret of its choice. The idea is simple: perform independent sharings of different secrets. Instead of only sharing the real secret s according to the access structure Γ, the dealer also independently shares a uniformly random value f according to Γ, and gives a copy of all these shares, together with f, to the tracer as part of the tracing key. Since the shares of f are indistinguishable from those of s, and the tracer knows f, it can turn them into shares of any fake secret s^* by adding $s^* - f$ to the share of one party of I in each minimal authorized subset. Then, the tracer can use the resulting shares of the parties in I as input to the reconstruction box, which will then output the desired fake secret s^*.

However, this introduces a problem for honest parties: during reconstruction, they would now reconstruct both the real and the fake secret, with no obvious way to distinguish the real one. To solve this, we use the indicator technique of Farràs and Guiot [FG26]. Specifically, together with each secret, we also share a value that acts as an indicator for the correct secret to reconstruct: the indicator 0 for the real secret, and a uniformly random indicator ω for the fake one, which is also given to the tracer. This ensures that, during reconstruction, only the indicators corresponding to the real secret sum to 0, allowing honest parties to identify it. In turn, when crafting its queries, the tracer also adds $-\omega$ to the indicator share that it modifies, so that the indicators of the crafted shares sum to 0 and the reconstruction box takes them as valid ones.

Disabling Parties. Next, we observe that the DNF-based scheme greatly simplifies the tracer's task of disabling parties outside I. To disable a party j, the tracer only needs to replace with uniformly random values the share components of the crafted shares for parties in I that correspond to minimal authorized subsets also containing the party j. During reconstruction, the indicators for these subsets will no longer sum to 0 and therefore will be ignored, effectively removing party j from the reconstruction process.

Conditionally Disabling Parties. Finally, it remains to enable the tracer to conditionally disable a party j depending on a bit of its identifier id_j. To do so, we combine the two previously used tricks. For each bit k of id_j, we independently share a uniformly random value and a uniformly random indicator according to the minimal authorized subsets containing j. In this case, we store in the tracing key these two random values and all the shares except for that of the party j. Additionally, if the k-th bit of id_j is 0, we replace the share of party j with a uniformly random value.

Using this setup, to extract the k-th bit of id_j, the tracer shifts the shares of the parties in I as before, so that they share a fake secret s^* of its choice, queries the reconstruction box with them, and observes the output. If the box returns the fake secret s^*, then the share of the party j was not replaced and $\mathsf{id}_j^k = 1$. Otherwise, if the box does not return the secret, then $\mathsf{id}_j^k = 0$.

Once all these challenges are addressed, we can apply the jump-based tracing strategy of [GJP26] to this modified DNF-based scheme to obtain our information-theoretic $\emptyset$-strong traceable scheme. To consolidate the ideas and make the construction more tangible, we next illustrate how it works with a toy example.

Example 1.4. Consider the access structure Γ on 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. For simplicity, we consider the case that the corrupted party p_3 constructed a reconstruction box R with label $I = \{p_2\}$, and we only illustrate the sharing and tracing steps needed for this particular case.

First, we show how to share the secret s according to Γ:

1. Share the secret s and the indicator 0 independently with the DNF scheme for Γ. That is, for each minimal authorized subset in Γ, share the secret s and the indicator 0 independently with the additive scheme:

   $p_1 \ll \mathsf{sh}_1^s = \{(r_{1,2}^0, r_{1,2}^s)\}$, $\quad p_2 \ll \mathsf{sh}_2^s = \{(-r_{1,2}^0, s - r_{1,2}^s), (r_{2,3}^0, r_{2,3}^s)\}$,
   $p_3 \ll \mathsf{sh}_3^s = \{(-r_{2,3}^0, s - r_{2,3}^s)\}$.

2. For each minimal authorized subset in Γ, share a uniformly random indicator ω and a uniformly random fake secret f, and store all the shares and the values ω, f in the tracing key tk:

   $p_1, \mathsf{tk} \ll \mathsf{sh}_1^f = \{(r_{1,2}^\omega, r_{1,2}^f)\}$, $\quad p_3, \mathsf{tk} \ll \mathsf{sh}_3^f = \{(\omega_{2,3} - r_{2,3}^\omega, f_{2,3} - r_{2,3}^f)\}$,
   $p_2, \mathsf{tk} \ll \mathsf{sh}_2^f = \{(\omega_{1,2} - r_{1,2}^\omega, f_{1,2} - r_{1,2}^f), (r_{2,3}^\omega, r_{2,3}^f)\}$, $\quad \mathsf{tk} \ll \omega_{1,2}, f_{1,2}, \omega_{2,3}, f_{2,3}$.

3. Assign a uniformly random 2-bit identifier to p_3 and store it in the verification key vk:
$$\mathsf{id}_3 = 01, \qquad \mathsf{vk} \ll \mathsf{id}_3.$$

4. For each bit of id_3, i.e. for the bits $\mathsf{id}_3^1 = 0$ and $\mathsf{id}_3^2 = 1$, do the following: share a uniformly random indicator $\omega_{3,b}$ and a uniformly random fake secret $f_{3,b}$ for the minimal authorized subset in Γ containing p_3, i.e. $\{p_2, p_3\}$, store the share of p_2 and the values $\omega_{3,b}, f_{3,b}$ in the tracing key, and replace the share of p_3 by uniformly random values if $\mathsf{id}_3^b = 0$:
$$\begin{aligned}
&p_2, \mathsf{tk} \ll \mathsf{sh}_2^{3,1} = \{(r_{3,1}^{\omega}, r_{3,1}^{f})\}, \quad p_3 \ll \mathsf{sh}_3^{3,1} = \{(\omega_{3,1} - r_{3,1}^{\omega}, f_{3,1} - r_{3,1}^{f})\},\\
&p_2, \mathsf{tk} \ll \mathsf{sh}_2^{3,2} = \{(r_{3,2}^{\omega}, r_{3,2}^{f})\}, \quad p_3 \ll \mathsf{sh}_3^{3,2} = \{(\omega_{3,2} - r_{3,2}^{\omega}, f_{3,2} - r_{3,2}^{f})\},\\
&\mathsf{tk} \ll \omega_{3,1}, f_{3,1}, \omega_{3,2}, f_{3,2},
\end{aligned}$$
where the blue color indicates that these shares are not the result of any sharing process, but are instead sampled uniformly at random from $\mathbb{Z}_m$.

The sharing steps to enable traceability against the parties p_1 and p_2 are analogous. Next, we show how to trace back the corrupted subset $\{p_3\}$:

1. Select a uniformly random bit $\beta \in \{0, 1\}$ and shift the share sh_2^f stored in tk by adding $(-\omega_{1,2}, s_\beta - f_{1,2})$ and $(-\omega_{2,3}, s_\beta - f_{2,3})$ to its components, so that they share the indicator 0 and the fake secret s_β:
$$\mathsf{sh}_2^{s_\beta} = \{(-r_{1,2}^{\omega}, s_\beta - r_{1,2}^{f}), (r_{2,3}^{\omega} - \omega_{2,3}, s_\beta + r_{2,3}^{f} - f_{2,3})\}.$$
Query the reconstruction box R with $\mathsf{sh}_2^{s_\beta}$. Assuming that R is good, it will recover the fake secret s_β:
$$s_\beta \leftarrow R(\mathsf{sh}_2^{s_\beta}).$$
2. Observe that the box is able to recover the fake secret.
3. Now, disable p_1 by replacing the corresponding share component of $\mathsf{sh}_2^{s_\beta}$ with uniformly random values:
$$\mathsf{sh}_2^{s_\beta} = \{(-r_{1,2}^{\omega}, s_\beta - r_{1,2}^{f}), (r_{2,3}^{\omega} - \omega_{2,3}, s_\beta + r_{2,3}^{f} - f_{2,3})\}.$$
4. Select a new uniformly random bit β, compute $\mathsf{sh}_2^{s_\beta}$ as in Step 1, and query the reconstruction box R with it:
$$s_\beta \leftarrow R(\mathsf{sh}_2^{s_\beta}).$$
5. Since the behavior of the box has not changed, keep disabling parties.
6. Disable p_3 by replacing the corresponding share component of $\mathsf{sh}_2^{s_\beta}$ with (new) uniformly random values:
$$\mathsf{sh}_2^{s_\beta} = \{(-r_{1,2}^{\omega}, s_\beta - r_{1,2}^{f}), (r_{2,3}^{\omega} - \omega_{2,3}, s_\beta + r_{2,3}^{f} - f_{2,3})\}.$$

7. Select a new uniformly random bit β and query the reconstruction box R with the modified share $\mathsf{sh}_2^{s_\beta}$:

$$\perp \leftarrow R(\mathsf{sh}_2^{s_\beta}).$$

8. Since the behavior of the box has changed, identify p_3 as the corrupted party and move to recover its identifier, which will be used as the accusation proof.
9. Select a uniformly random bit β, compute the share $\mathsf{sh}_2^{3,1}$ as in Step 1, and query the reconstruction box R with the resulting share $\mathsf{sh}_2^{3,1,s_\beta}$:

$$\perp \leftarrow R(\mathsf{sh}_2^{3,1,s_\beta}).$$

10. Since the box does not return the fake secret s_β, set $\mathsf{id}_3^1 = 0$.
11. Select a uniformly random bit β, compute the share $\mathsf{sh}_2^{3,2}$ as in Step 1, and query the reconstruction box R with the resulting share $\mathsf{sh}_2^{3,2,s_\beta}$:

$$s_\beta \leftarrow R(\mathsf{sh}_2^{3,2,s_\beta}).$$

12. Since the box returns the fake secret s_β, set $\mathsf{id}_3^2 = 1$.
13. Use the identifier $\mathsf{id}_3 = 01$ as a proof π against p_3.
14. The verifier checks whether the proof π coincides with the identifier of p_3 stored in the verification key vk.

Information-theoretic Hidden Mildly Strong Traceable Schemes. Before discussing the mildly strong and hidden mildly strong models, we recall the impossibility result from [GJP26] for strong traceability. It states that no traceable secret sharing scheme for general access structures can simultaneously satisfy strong traceability and tracer secrecy (or strong non-imputability). The core reason is simple: since $I \cap J \neq \emptyset$ and the reconstruction box requires the shares of all parties in I, any tracing procedure must be able to reconstruct the exact shares of the parties in $I \cap J$ from the tracing key alone, which violates either tracer secrecy or non-imputability.

Notice that this impossibility no longer applies in the mildly strong traceability model. In this setting, the tracer is not required to use all the shares of the parties in I as input queries, which in principle allows it to avoid the shares of parties in $I \cap J$. However, this raises a new challenge: without any additional knowledge of I, how can the tracer know which parties to avoid? This difficulty is precisely what shows that traditional secret sharing schemes are insufficient for tracing in the mildly strong model. In fact, for certain choices of Γ, I, and J, the tracer may need an exponential number of queries on average to find a valid input query. See Example 4.5 for further details.

We can overcome this issue with the following approach: by hiding the access structure from the parties, the tracer can make the reconstruction box believe that the subset $J \cup \{i\}$ is authorized for some $i \in I \setminus J$.[1] This reduces the

[1] Notice that $I \setminus J \neq \emptyset$ because $I \cup J$ is authorized but J alone is not.

complexity dramatically: instead of potentially requiring an exponential number of queries, the tracer now needs to perform at most $|I|$ queries to find a valid one.

Once the difficulty caused by the intersection $I \cap J$ is addressed by hiding the access structure, the hidden mildly strong traceability model naturally follows. The remaining steps of the tracing procedure then adopt the approach of Farràs and Guiot [FG26], which leads to a traceable scheme with exponential share size but only requiring a polynomial number of queries. See Section 4 for details.

Impossibility of Stateful Tracing. Our impossibility result is based in the following key insights. First, observe that since there is only one real secret, a stateful reconstruction box can, after receiving a valid input query and reconstructing the secret, ignore subsequent queries and continue outputting the same secret.[2]

This behavior remains consistent with that of a good reconstruction box, because if the first valid query consists of real shares, the box will always output the correct secret. This observation implies that any tracing procedure for a stateful reconstruction box must rely on just a single valid input query, since the box's behavior may depend exclusively on that first query.

Once this observation is in place, notice that to satisfy non-imputability, any two reconstruction boxes R_{p_1} and R_{p_2} built by parties p_1 and p_2 must behave differently on any valid crafted input query q for both parties. Otherwise, the tracer would be able to craft a proof for the party p_1 (resp. p_2) from a box built by p_2 (resp. p_1), which breaks non-imputability.

Consequently, if two parties p_1 and p_2 are equivalent, i.e. they play the same role in the access structure, the corrupted subset $\{p_1, p_2\}$ can construct a reconstruction box R_{p_1,p_2} that effectively evades tracing. Specifically, R_{p_1,p_2} internally simulates R_{p_1} and R_{p_2}, compares their outputs, and outputs $\perp$ whenever a discrepancy is detected.

This strategy becomes particularly problematic for access structures where six parties have an equivalent role. In such cases, any disjoint pair of these parties can construct a reconstruction box as described above. As a result, the tracer would not be able to detect which pair of those parties built the box, violating the traceability of the scheme.

1.3 Related Work

The notion of traceable secret sharing was first studied by Goyal, Song, and Srinivasan [GSS21]. They considered threshold access structures and proposed a construction based on the Goldreich–Levin decoding algorithm [GL89]. Boneh, Partap, and Rotem [BPR24] later revisited this notion and introduced a different traceability model. They presented two constructions for threshold access structures, based respectively on Shamir's secret sharing scheme [Sha79] and Blakley's

[2] For a valid input query, we refer to a query consisting of shares (genuine or crafted by the tracer) for a subset $I' \subseteq I$ such that $I' \cup J \in \Gamma$.

scheme [Bla79]. A key feature of their constructions is that they incur only a constant-factor overhead in share size compared to standard secret sharing. Notably, their construction based on Blakley's scheme is, to date, the only known traceable secret sharing scheme that works against stateful adversaries. Their work also provides a careful comparison between the two traceability models, which we refer to for further details.

Since then, several works have explored different aspects of traceable secret sharing. Hoffmann [Hof24] proposed a construction based on the Chinese Remainder Theorem for weighted threshold access structures, while Leslie and Dutta [LD25] focused on standard threshold access structures. Baghery et al. [BEMS25] introduced traceable verifiable secret sharing schemes for threshold access structures. Other extensions include traceable bottom-up secret sharing by Hajra et al. [HKMP25], tracing schemes with public verifiability by Luong et al. [LPLK25], and constructions supporting public tracing by Dey et al. [DHKP26].

More recently, two works are particularly relevant to this paper, as they extend traceable secret sharing to general access structures. Goyal, Jain, and Partap [GJP26] introduced the notions of $\emptyset$-strong and strong traceability. They presented an efficient construction for the $\emptyset$-strong model based on indistinguishability obfuscation, and proved an impossibility result for the strong traceability model. In contrast, Farràs and Guiot [FG26] proposed an information-theoretic construction for general access structures in a related but distinct model, and their tracing strategy relies on hiding the access structure from the parties. The similarities, differences, and connections between these two models are discussed in more detail in Section 2.4.

Finally, other works proposed notions closely related to traceable secret sharing, including secret sharing with snitching [DFLM24,BDF+25], traceable verifiable random functions [BPR25], and deniable secret sharing [CDK+26].

1.4 Open Questions

Our results point to several open problems in the study of traceable secret sharing schemes. Below, we highlight some directions in which future work could lead to notable improvements over the current state of the art.

Information-theoretic Traceable Secret Sharing. The only information-theoretic secret sharing primitives for tracing in general access structures are the ones presented in this work and in [FG26]. In both cases, the share size is exponential in the number of participants in the worst case. We cannot hope for a traceable secret sharing scheme with sub-exponential share size without improving the current best secret sharing constructions, which also have exponential share size in the worst case [LV18,ABF+19,ABNP20,AN21,Nir26]. However, since our construction acts as a compiler for the DNF-based scheme, it is natural to ask whether a similar transformation can be obtained for information-theoretic secret sharing schemes built from more efficient representations of the access structure, such as monotone formulas or monotone span programs.

Lower Bounds on Traceable Secret Sharing. Regarding tracing limitations, there are no known lower bounds on the share size of information-theoretic tracing secret sharing schemes besides the ones for standard schemes, which are exponential for linear schemes [BF20,BGW99,PR18,RBG01] and linear for general constructions [Wil26].

Bridging the Gap Between Information-theoretic and Computational $\emptyset$-Strong Traceable Secret Sharing. Currently, there are two distinct constructions for the $\emptyset$-strong tracing model. The scheme presented in this work is information-theoretic, with share size polynomial in the DNF size of the access structure, whereas the construction of Goyal, Jain, and Partap [GJP26] achieves share size proportional to the circuit size of the access structure at the cost of relying on iO. An open problem is whether this gap can be bridged with computational tracing schemes that are secure under weaker assumptions (such as OWF or PK) and still retain efficiency.

Mildly Strong Traceable Secret Sharing Schemes. In this work, we introduced mildly strong traceability as a relaxation of the strong traceability model of Goyal, Jain, and Partap [GJP26], for which an impossibility result was shown. We presented a traceable scheme for this relaxed model, at the cost of hiding the access structure from the parties. It is still an open question to determine if hiding the access structure is indeed necessary, or if there are less restrictive properties that allow tracing. In this context, understanding the precise relationship between anonymous [BGI$^+$25] and traceable schemes is particularly interesting.

Verifiable Traceable Secret Sharing Schemes. Another interesting direction is to construct verifiable traceable schemes for general access structures. This setting appears particularly challenging, as all existing tracing techniques, including ours, inherently rely on querying reconstruction boxes with fake or invalid shares, whereas verifiable schemes are designed precisely to prevent such behavior. This was already identified as an open problem in [BPR24].

1.5 Organization

In Section 2, we present the notation used in this work and the preliminaries on traceable secret sharing schemes. In Section 3, we present the information-theoretic $\emptyset$-strong traceable scheme of Theorem 1.1. In Section 4, we introduce the mildly strong and hidden mildly strong models, discuss their limitations, and present the information-theoretic hidden mildly strong traceable scheme of Theorem 1.2. Later, in Section 5, we present the stateful traceability model and the impossibility result of Theorem 1.3.

2 Preliminaries

2.1 Notation

We notate $\mathbb{N}$ and $\mathbb{Z}$ for the sets of natural and integer 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 $\{1, \ldots, n\}$ as $[n]$. For a set S, we denote its cardinal by $|S|$, its power set by 2^S, and we denote by $s \leftarrow_\$ S$ 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 $\mathbf{1}^n \in \{0,1\}^n$.

2.2 Secret Sharing Schemes

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

Definition 2.1 (Access Structure). *Let $P = [n]$ be a set of n parties. A collection $\Gamma \subseteq 2^P$ is monotone if $B \in \Gamma$ and $B \subseteq C$ imply that $C \in \Gamma$. An n-party access structure is a monotone collection $\Gamma \subseteq 2^P$ of non-empty subsets of P. 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.[3]

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* Reconstruction *such that*

- ***Perfect Correctness.*** *For any secret $s \in \mathcal{S}$ and any authorized set $A \in \Gamma$, it holds that*
$$\mathbf{Pr}[s = \mathsf{Rec}(\mathsf{Share}(s,\Gamma)_A, A)] = 1,$$
where $\mathsf{Share}(s,\Gamma)_A$ denotes the restriction of the output of $\mathsf{Share}(s,\Gamma)$ to the parties in A.

[3] 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 $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$.

– ***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*

$$\mathbf{Pr}[\{\mathsf{sh}_i\}_{i\in B} = \mathsf{Share}(s, \Gamma)_B] = \mathbf{Pr}[\{\mathsf{sh}_i\}_{i\in B} = \mathsf{Share}(s')_B].$$

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* Reconstruction *such that*

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

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

where $\mathsf{Share}(\mathbf{1}^\lambda, s, \Gamma)_A$ 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*

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

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.

2.3 Hiding Secret Sharing

In a recent work, Farràs and Guiot [FG26] introduced the notion of *hiding secret sharing*, which strengthens standard secret sharing by requiring that the access structure itself remain hidden from the parties. Concretely, the access structure Γ used to share the secret is not revealed; instead, the parties are only given a family of access structures $\mathcal{F}$ such that $\Gamma \in \mathcal{F}$.

In this setting, any subset of parties A, given their shares, cannot distinguish the actual access structure Γ from any other $\Gamma' \in \mathcal{F}$ that is consistent with A.

That is, Γ and Γ' are indistinguishable to A whenever they agree on all subsets of A: a subset of A is authorized in Γ if and only if it is authorized in Γ'.

This notion of hiding access structures will play a crucial role later in Section 4, where it is used to construct a mildly strong traceable secret sharing scheme for general access structures in the information-theoretic setting. We now give the formal definition.

Definition 2.5 (Hiding Secret Sharing Scheme,[FG26]). *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 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 $\{\mathsf{sh}_i\}_{i\in[n]}$.*
- $\mathsf{Rec}(\{\mathsf{sh}_i\}_{i\in A}) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of shares for an authorized subset A of some $\Gamma \in \mathcal{F}$. It outputs the secret s.*

These algorithms must satisfy the following requirements:

- ***Restricted Secret Sharing Schemes.*** *For any $\Gamma \in \mathcal{F}$, the restriction of the* Share *algorithm to Γ together with the* Reconstruction *algorithm form a 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*

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

In their work, Farràs and Guiot [FG26] introduced an information-theoretic hiding secret sharing scheme for general access structures, defined over a special class of families called *covering families*. Informally, a family of access structures is a covering family if it satisfies the following property: for every unauthorized subset in any access structure of the family, there exists another access structure in the same family in which this subset is strictly contained in a minimal authorized set. Its definition follows.

Definition 2.6. *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 exist $B \subseteq [n]$ and $\Gamma' \in \mathcal{F}$ such that $A \subsetneq B$ and B is minimal in Γ'.*

Remark 2.7. As noted in [FG26], for any $n \in \mathbb{N}$, the family of all access structures over n parties is a covering family.

2.4 Traceable Secret Sharing Schemes

The notion of *traceable secret sharing* extends traditional secret sharing by adding the ability to identify parties who illegally redistribute their shares. More

precisely, a coalition of corrupted parties is assumed to construct a *reconstruction box* R that, given sufficiently many valid shares as input, is able to reconstruct the secret. The goal is to trace R back to at least one of the corrupted parties using only black-box access to it.

Recently, Goyal, Jain, and Partap [GJP26], and Farràs and Guiot [FG26] introduced two closely related, but not equivalent, models of traceable secret sharing for general access structures. We first present the common syntax of [FG26,GJP26] and a core definition of information-theoretic traceable secret sharing schemes for general access structures. We then describe the two models in detail, highlighting both their similarities and their key differences.

Definition 2.8 (Traceable Secret Sharing Scheme,[FG26,GJP26]). *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 traceable secret sharing scheme for Γ is a quintuple 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 parameters the secret s, and the access structure Γ. It outputs a set of shares $\{\mathsf{sh}_i\}_{i\in[n]}$, a tracing key tk and a verification key vk.*
- $\mathsf{Rec}(\{\mathsf{sh}_i\}_{i\in A}) \mapsto s$ *is the deterministic reconstruction algorithm. It takes as input a set of shares sh_i for an authorized set $A \in \Gamma$. It outputs the secret s.*
- $\mathsf{ShGen}(\mathsf{tk}, k, \mathsf{param}) \mapsto \{\mathsf{sh}'_i\}_{i\in[k]}$ *is the randomized share generation algorithm. It takes as input a traceable key $\mathsf{tk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$, a size $k \in [n]$ and some extra parameters param, and outputs a set of pseudoshares $\{\mathsf{sh}'_i\}_{i\in[k]}$.*[4]
- $\mathsf{Trace}^R(\mathsf{tk}) \mapsto (I, \pi)$ *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 $I \subset [n]$ of identities of leaking parties and an associated proof π.*[5]
- $\mathsf{Verify}(\mathsf{vk}, I, \pi) \mapsto \{0, 1\}$ *is the deterministic verification algorithm. It takes as input a verification key $\mathsf{vk} = (\mathsf{vk}_1, ..., \mathsf{vk}_n)$, an alleged traitor subset 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*

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

[4] The extra parameters param vary depending on the tracing model considered, and will be explicitly defined later for each of the models.

[5] Depending on the tracing model considered, the tracing algorithm also receives as input the public values of the tracing experiment, such as the label of the reconstruction box, the two secrets s_0, s_1, the parameter ϵ, and the security parameter λ. To simplify the notation, we omit these inputs when calling the tracing algorithm in the experiments.

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.*** *Depends on the tracing model considered.*
- ***Perfect Non-imputability.*** *Depends on the tracing model considered.*

Remark 2.9. Being precise, the ShareGeneration algorithm is not strictly required, as it can be viewed as a subroutine of the Trace algorithm. Nevertheless, for clarity of exposition, we present ShareGeneration separately. This choice allows for a more modular and compartmentalized presentation of the schemes in the subsequent sections.

The main differences between the traceability models of [FG26] and [GJP26] arise in the definition of reconstruction box considered, as well as in the game-based definitions of traceability and non-imputability.

2.5 Strong Traceable Secret Sharing Schemes

In the model of Goyal, Jain, and Partap [GJP26], they present the notion of minimally-useful reconstruction box. In this definition, the reconstruction box contains a label consisting of a subset of parties $I \subseteq [n]$, and given the shares of all the parties in I, the box can distinguish whether the shares correspond to one of two possible secrets, s_0 or s_1.

Definition 2.10 (Minimally-useful 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 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* $J \notin \Gamma$*, a corresponding set of shares* $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$*, and a label* $I \subseteq [n]$*, a reconstruction box* R *is* $(\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)$*-minimally-useful if*

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

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

Depending on whether $I \cap J = \emptyset$ or not, they present two different notions of traceability: $\emptyset$*-strong* and *strong traceability*. For each of these concepts, we present both the tracing experiment and the formal definition. We start by presenting the notion of $\emptyset$-strong traceability, in which it is required that $I \cap J = \emptyset$.

Definition 2.11 (Perfect $\emptyset$-Strong Traceability). *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 traceable scheme for* Γ*. For any* $\epsilon \in [0, \frac{1}{2}]$*,* Σ *is perfect* $\emptyset$*-strong traceable if for every adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\emptyset-\mathrm{strongtrac}}_{\mathcal{A},\Sigma,\epsilon} := \mathbf{P}[\mathbf{Exp\emptyset-StrongTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = \frac{1}{2}.$$

In contrast, by allowing $I \cap J \neq \emptyset$, we obtain the notion of *strong traceability*, whose corresponding experiment and formal definition are given next.

Experiment $\mathbf{Exp\emptyset-StrongTrace}_{\mathcal{A},\Sigma,\epsilon}$

1. $(\Gamma, J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $b \leftarrow_{\$} \{0,1\}$.
3. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$.
4. $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$.
5. $(R, I) \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
6. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
7. $\mathsf{GoodBox} \coloneqq (I \cap J = \emptyset) \wedge (J \notin \Gamma) \wedge (I \notin \Gamma) \wedge (R$ is $(\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)$-minimally-useful).
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. 2: The tracing experiment for an $\emptyset$-strong traceable secret sharing scheme Σ for an access structure Γ.

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

1. $(\Gamma, J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $b \leftarrow_{\$} \{0,1\}$.
3. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$.
4. $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$.
5. $(R, I) \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
6. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
7. $\mathsf{GoodBox} \coloneqq (J \notin \Gamma) \wedge (I \notin \Gamma) \wedge (R$ is $(\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)$-minimally-useful).
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. 3: The tracing experiment for a strong traceable secret sharing scheme Σ for an access structure Γ. The steps highlighted in blue indicate the differences compared to the tracing experiment of Fig. 2.

Definition 2.12 (Perfect Strong Traceability). *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 traceable scheme for Γ. For any $\epsilon \in [0, \frac{1}{2}]$, Σ is perfect strong traceable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{strongtrac}}_{\mathcal{A},\Sigma,\epsilon} := \mathbf{P}[\mathbf{ExpStrongTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = \frac{1}{2}.$$

With respect to non-imputability, both the $\emptyset$-strong and strong traceability models rely on the same experiment and definition, which we state next.

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

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. 4: The non-imputability experiment a traceable secret sharing scheme Σ with access structure Γ.

Definition 2.13 (Perfect Strong Non-imputability). *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 traceable scheme for $\mathcal{F}$. Then, Σ is strong non-imputable if for every adversary $\mathcal{A}$, it holds that*

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

Remark 2.14. Note that strong traceability is strictly stronger than $\emptyset$-strong traceability. Nevertheless, both models are of independent interest. Goyal, Jain, and Partap [GJP26] constructed an $\emptyset$-strong traceable scheme based on indistinguishability obfuscation, while also proving an impossibility result for the strong traceability model. Specifically, they showed that no traceable secret sharing scheme for general access structures can simultaneously achieve strong traceability and tracer secrecy or non-imputability.

2.6 Hidden traceable Secret Sharing Schemes

The model introduced by Farràs and Guiot [FG26], which we refer to in this work as *hidden traceability*, differs from the model in [GJP26] primarily in that it is built on a hiding secret sharing scheme. As a result, neither the parties nor the adversary are aware of the actual access structure used to share the secret; they only know the family of access structures to which it belongs. This

additional hiding property is the key source of the differences with respect to the strong traceability setting, and it is precisely what enables traceability in this model. In Section 4, we further clarify the relationship between the two models by showing how the hidden traceability setting can be related to, and embedded within, the strong traceability framework.

We start by introducing the notion of *hiding-good reconstruction box*. In this case, hidden traceable schemes are defined with respect to a family of access structures, and the notion of reconstruction boxes must be adapted accordingly. Specifically, we require a hiding-good reconstruction box to behave correctly for all access 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 such access structure, the reconstruction box must successfully reconstruct the secret when it is given full shares generated according to that access structure, provided that the shares of the corrupted parties are the same as in the original sharing. This requirement is natural, since the reconstruction algorithm does not depend on the access structure, which remains hidden from the parties.

Definition 2.15 (Hiding-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 $J \notin \Gamma$, and a corresponding set of shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$, a reconstruction box R is $(\mathcal{F}, \mathbf{sh}, s_0, s_1, b, \epsilon)$-hiding-good if for any access structure $\Gamma' \in \mathcal{F}$ that is consistent with J, any unauthorized subset $I \notin \Gamma'$ such that $J \cap I = \emptyset$ and $J \cup I \in \Gamma'$, it holds that*

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

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

Once the notion of hiding-good reconstruction boxes is established, we present the definitions of hidden traceability and hidden non-imputability. In general terms, these definitions closely follow those of the strong traceability setting. The main distinction is that the access structure is now hidden from the parties: the adversary neither chooses nor learns the specific access structure used, and instead only knows the family of access structures for which the scheme is defined. To highlight the differences introduced in this setting, all modifications are marked in blue. We start with the notion of hidden traceability.

Definition 2.16 (Perfect Hidden 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 hidden traceable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\text{hiddentrac}}_{\mathcal{A},\Sigma,\epsilon} := \mathbf{P}[\mathbf{ExpHiddenTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = \frac{1}{2}.$$

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

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. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$.
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. $\mathsf{GoodBox} := (J \notin \Gamma) \wedge (R \text{ is } (\mathcal{F}, \mathbf{sh}, s_0, s_1, b, \epsilon)\text{-hiding-good})$.
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. 5: The tracing experiment for a hidden 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. 2.

Finally, we present the experiment and the definition of hidden non-imputability.

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

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. 6: The non-imputability experiment for a hidden 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. 4.

Definition 2.17 (Perfect Hidden 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*

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

Farràs and Guiot [FG26] presented an information-theoretic hidden traceable scheme for general access structures. We refer the reader to their work for full details of the construction.

3 An Information-theoretic ∅-Strong Traceable Scheme

In this section, we construct an ∅-strong traceable secret sharing scheme for general access structures in the information-theoretic setting. We begin by outlining the high-level ideas behind its algorithms, and then present the full scheme.

3.1 An Overview of the Algorithms

Our scheme combines the jump-based tracing technique of [GJP26] with the technique of independently sharing multiple fake secrets introduced in [FG26]. The core ideas underlying each of the algorithms in our construction are as follows.

Share. We begin by sharing the secret and the indicator $s, 0 \in \mathbb{Z}_m$ among the parties according to the access structure Γ using the additive scheme for the DNF representation of Γ.[6] Additionally, for each party i, we assign a secret identifier $\mathsf{id}_i = (\mathsf{id}_i^1, ..., \mathsf{id}_i^{\log m}) \in \mathbb{Z}_m$ sampled uniformly at random. These identifiers are stored as the verification key and are later used by the verifier to validate the accusation issued by the tracer.

Next, for every minimal authorized subset $A \in \Gamma$, we perform an additional sharing of a uniformly random indicator together with a uniformly random fake secret $\omega, f \in \mathbb{Z}_m$ among A. All the resulting shares, together with the values ω and f, are stored in the tracing key. Since $\omega \neq 0$ except with probability $1/m$, the indicator shares of this sharing do not sum to 0, which guarantees that the subset A cannot accidentally reconstruct the fake secret f in place of the genuine secret s during the reconstruction process. At the same time, since the tracer knows ω and f, it can turn this sharing into a sharing of the indicator 0 and of any fake secret of its choice by shifting the share of a single party, as described in the ShareGeneration algorithm below.

Intuitively, this step (Fig. 7, Step 4 (a)) enables the tracer to observe *jumps* in the behavior of the reconstruction box without ever exposing the real secret s. More precisely, from the point of view of the corrupted parties, the share components of these additional sharings are uniformly random, exactly as those of the genuine sharing of s. As a result, when the tracer queries the reconstruction box using the shifted shares taken from the tracing key, the box believes it is operating on valid shares of the secret. If the tracer has enabled a sufficient number of parties, the reconstruction box will therefore reconstruct and output the fake secret chosen by the tracer. Conversely, if the tracer has not enabled

[6] In the tracing experiment, $s \in \{s_0, s_1\}$. Nevertheless, the sharing algorithm only receives s, and it does not need to know the other secret.

enough parties, the reconstruction box will fail to reconstruct it and its output behavior will change. This observable change reveals a jump in functionality, which allows the tracer to identify the last party disabled as a corrupted one.

Later, for each minimal authorized subset $A \in \Gamma$, each party $j \in [n]$, and each bit position $k \in [\log m]$, we generate a sharing of a uniformly random indicator and fake secret $\omega, f \in \mathbb{Z}_m$ among the parties in A. The values ω, f and all share components except that of party j are included in the tracing key and will later be used by the tracer to recover the identifier id_j bit by bit. Moreover, if $j \in A$, the share of the party j is replaced with a uniformly random value whenever $\mathsf{id}_j^k = 0$. As a result, once the tracer shifts the shares as before, the fake secret can be reconstructed through the minimal authorized subsets containing j if and only if the k-th bit of id_j equals 1, which enables the tracer to recover id_j one bit at a time. For the minimal authorized subsets that do not contain j, these are ordinary fake sharings, which we include so that all minimal authorized subsets have the same number of share components.

The purpose of this part of the sharing procedure (Fig. 7, Step 4 (b)) is to allow the tracer to recover the identifier id_j of the previously detected corrupted party j, one bit at a time. The mechanism is as follows. For a given bit position k, if $\mathsf{id}_j^k = 1$, by construction the reconstruction box can successfully recover the fake secret whenever it is provided with sufficiently many shares. In contrast, if $\mathsf{id}_j^k = 0$, the reconstruction fails, since the share of party j has been replaced with a uniformly random value and thus carries no information about the fake secret. Crucially, the tracing key contains no information about the share component actually received by party j: although the tracer can compute the genuine share component of j from ω, f, and the remaining components, it cannot tell whether j received it or a uniformly random value. This design choice is necessary to ensure non-imputability: whether id_j^k equals 0 or 1 is determined precisely by whether the share of j is genuine or random, and the tracer must not be able to distinguish these cases directly. As a result, the tracer can learn the value of id_j^k only indirectly, through the behavior of the reconstruction box, which has access to the true share held by the corrupted party j.

Each of the previous fake sharings is performed N times independently, where N is polynomial in n, and in the inverse of the distinguishing advantage ϵ of the reconstruction box. The reason is that the reconstruction box may be imperfect, so the tracer can only learn its behavior by estimating the fraction of queries that it answers correctly. These estimates are reliable only if the queries behind each of them are independent, and therefore the tracer builds the h-th query of each estimate from the h-th copy of the fake sharings. If, instead, the same fake sharing were used in all the queries, the estimates would only capture the behavior of the box on that particular sharing, which may differ significantly from its behavior on genuine shares.

Finally, we randomly permute the share components of each minimal authorized subset $A \in \Gamma$, using the same permutation for all of them. This permutation ensures that the reconstruction box cannot distinguish which share components belong to the sharing of the real secret s and which correspond to the fake shar-

ings. Moreover, using the same permutation for all of them ensures that the N copies used in each estimate of the tracer occupy a uniformly random set of positions, so a reconstruction box whose behavior depends on these positions cannot bias the estimates.

Reconstruction. Each minimal authorized subset $A \in \Gamma$ tests the indicator shares associated with the share components of A to find the unique combination that reconstructs the indicator value 0.[7] This combination identifies the specific sharing from which the parties should reconstruct the actual secret s. Since every fake sharing shares a uniformly random indicator, which remains uniformly random even if the share of a party is replaced, assuming m is sufficiently large, any incorrect combination of indicator shares leads to the value 0 only with negligible probability. This ensures the statistical correctness of the reconstruction process.

ShareGeneration. The algorithm generates two types of pseudoshares: one used to detect a corrupted party j, and another used later to recover its identifier id_j bit by bit. The type of pseudoshare produced is controlled by the input parameter t: setting $t = 0$ produces pseudoshares for detecting the party, while setting $t = 1$ produces pseudoshares for recovering its identifier. In both cases, the algorithm also receives the fake secret s^* that the pseudoshares must share, and the index $h \in [N]$ of the copy of the fake sharings to be used.

To construct pseudoshares for detecting a corrupted party, we first take from the tracing key the components of the parties in I in the h-th copy of the fake sharings of every minimal authorized subset $A \in \Gamma$ containing at least one party in I. Then, in each such subset A, we shift the components of one party in $I \cap A$ by $(-\omega, s^* - f)$, where ω and f are the random indicator and fake secret of the corresponding sharing. Since the components of all the parties in A add up to (ω, f), after this shift they add up to $(0, s^*)$, so A can recover the fake secret s^* exactly once using these pseudoshares. To match the size of a real share, we place these components in the position given by the permutation and pad the pseudoshares with uniformly random values. Then, for each party in $[j] \setminus I$, we replace the share components corresponding to minimal authorized subsets containing these parties with random values. This ensures that the minimal authorized subsets that include any party in $[j] \setminus I$ can not recover the fake secret s^*, effectively disabling those parties.

The construction of pseudoshares for recovering the k-th bit of the identifier id_j of a corrupted party j is analogous to that for detecting the party. The difference is that the tracing key components used in this case are those of the h-th copy of the fake sharings generated for the party j and the bit position k, which are shifted in the same way. In the minimal authorized subsets not

[7] As is typically implicit in secret sharing, reconstruction requires each party to provide its share components in the order in which they were received. Otherwise, the correlation between share components across parties is lost.

containing j, these are ordinary fake sharings, while in those containing j the shifted components share s^* only if the share of j was not replaced, that is, if $\mathsf{id}_j^k = 1$. Additionally, only the parties in $[j-1] \setminus I$ are disabled, because it is precisely the share of the party j that allows the tracer to determine whether the k-th bit of its identifier id_j is 0 or 1.

Trace. The tracing procedure is run by the tracer, which receives as input the tracing key, the label I of the reconstruction box, the two secrets s_0, s_1, and the advantage ϵ. It proceeds iteratively in two main steps. The first step (Fig. 8, Trace, Step 2.) detects a corrupted party j, and the second step (Fig. 8, Trace, Step 4.) recovers its identifier id_j.

To detect a corrupted party j, the tracer first uses the ShareGeneration algorithm to generate pseudoshares with all parties enabled and queries the reconstruction box R with them. In each query, the tracer selects a fresh uniformly random bit β, generates pseudoshares of the fake secret s_β, and checks whether R outputs β. Since the h-th query uses the h-th copy of the fake sharings, the N queries are built from independent sharings, and the Chernoff bound guarantees that the observed behavior of R closely approximates its true distribution. The tracer then repeats the procedure iteratively, disabling parties outside I cumulatively one by one. In each iteration, it checks whether the fraction of correct outputs changes by more than $\epsilon/(2n)$ compared to the previous step. When such a change is detected, the tracer stops and identifies the last disabled party as the corrupted one.

Intuitively, a change in the behavior of the reconstruction box occurs only at a step in which a corrupted party is disabled. This is because when an honest party is disabled, from the point of view of the box, the distribution of the pseudoshares presented to it is indistinguishable to that of the pseudoshares from the previous iteration. As a result, its behavior must remain unaltered.

Once the corrupted party j is identified, the tracer proceeds to recover its identifier id_j in a similar iterative manner. For each bit $k \in [\log m]$, the tracer uses the ShareGeneration algorithm to generate pseudoshares aimed at recovering the k-th bit of id_j, with all parties in $[j-1]\setminus I$ disabled, and queries the reconstruction box R with them, again selecting a fresh bit β in each query and using the h-th copy of the corresponding fake sharings in the h-th query. Then, it compares the resulting fraction of correct outputs with those observed in the first step right before and right after disabling j. If it is closer to the former, the tracer sets $\mathsf{id}_j^k = 1$; otherwise, it sets $\mathsf{id}_j^k = 0$.

The idea behind this second step is that, as described in the Share algorithm, at the k-th iteration the fake secret s_β can be reconstructed if and only if the k-th bit of id_j is 1. Equivalently, this corresponds to the reconstruction box not exhibiting a sudden change in its behavior.

Verify. The verification procedure is executed by the verifier and consists of a simple check: it verifies whether the value provided in the proof π matches the

identifier from the verification key corresponding to the accused party. If so, it outputs 1; otherwise, it returns 0.

As explained in the Share algorithm, the tracing key does not give any information about the identifier of any honest party i. Therefore, for a malicious tracer to falsely accuse an honest party, it would need to correctly guess its identifier. If this value is sampled uniformly from a large domain, the probability of such a guess succeeding is negligible.

3.2 The Scheme

We provide in Fig. 7 and Fig. 8 an outline of the scheme's construction, while its full description is given in Fig. 9 and Fig. 10. The formal statement is presented in Theorem 3.1, and its proof is deferred to Section 3.3.

Theorem 3.1 (Theorem 1.1 restated). *Let Γ be an access structure over n parties, let d be the size of its DNF representation, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in (0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m$, the protocol of Fig. 7 and Fig. 8 is an information theoretic $\emptyset$-strong traceable secret sharing scheme with statistical reconstruction, traceability, and non-imputability realizing Γ with a share size of $O(d \cdot n^3 \log^3 m/\epsilon^2)$ and $O(n^2(n + \log m) \log m/\epsilon^2)$ queries.*

3.3 Proof of Theorem 3.1

For clarity, we organize the proof of Theorem 3.1 as a sequence of propositions. We begin by showing that the protocol described in Fig. 9 and Fig. 10 is a secret sharing scheme.

Proposition 3.2. *Let Γ be an access structure over n parties, let d be the size of its DNF representation, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in (0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m$, the protocol of Fig. 9 and Fig. 10 is an information theoretic secret sharing scheme with statistical reconstruction realizing Γ with a share size of $O(d \cdot n^3 \log^3 m/\epsilon^2)$.*

Proof. We begin by proving the privacy of the scheme. Let $B \notin \Gamma$ be an unauthorized subset. Since every minimal authorized subset $A \in \Gamma$ contains a party outside B, the components of the parties in $A \cap B$ in the sharing of the real secret are uniformly random and independent by the privacy of the additive scheme. Moreover, each additional sharing shares a uniformly random pair $(\omega_{A,u}, f_{A,u})$, so all its components are uniformly random and independent even after replacing the share of a party with uniformly random values. Since all these sharings are performed independently, the shares of B consist of independent and uniformly random elements of $\mathbb{Z}_m$ regardless of the secret, the identifiers, and the permutation σ. In particular, the scheme has perfect privacy.

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

1. Share the indicator $0 \in \mathbb{Z}_m$ and the secret $s \in \mathbb{Z}_m$ according to Γ with the DNF scheme.
2. Assign a uniformly random identifier $\mathsf{id}_i = (\mathsf{id}_i^1, ..., \mathsf{id}_i^{\log m}) \in \mathbb{Z}_m$ to each party $i \in [n]$.
3. Set all the identifiers $\{\mathsf{id}_i\}_{i\in[n]}$ as the verification key.
4. For each minimal $A \in \Gamma$
   (a) Repeat N times
      i. Share a uniformly random indicator and a uniformly random fake secret $\omega, f \in \mathbb{Z}_m$ among A with the additive scheme.
      ii. Append all the shares and the values ω, f in the tracing key.
   (b) For each party $j \in [n]$ and each $k \in [\log m]$, repeat N times
      i. Share a uniformly random indicator and a uniformly random fake secret $\omega, f \in \mathbb{Z}_m$ among A with the additive scheme.
      ii. Append the values ω, f and all the shares except the one of the party j in the tracing key.
      iii. If $j \in A$ and $\mathsf{id}_j^k = 0$, replace the share of the party j by two uniformly random values $r_0, r_s \in \mathbb{Z}_m$.
5. Select uniformly at random a permutation of the $1 + N(1 + n \log m)$ share components of a minimal authorized subset, permute with it the share components corresponding to each minimal $A \in \Gamma$, and append such permutation in the tracing key.

$\mathsf{Rec}(\{\mathsf{sh}_i\}_{i\in A})$:

1. For each set of share components $\{(\mathsf{sh}_{i,A}^0 \mathsf{sh}_{i,A}^s)\}_{i\in A}$ shared for the authorized set A
   (a) If $\sum_{i\in A} \mathsf{sh}_{i,A}^0 = 0$, set $s = \sum_{i\in A} \mathsf{sh}_{i,A}^s$ and **break**.
2. Output s.

Fig. 7: Outline of the Share and Reconstruction algorithms of an $\emptyset$-strong traceable secret sharing scheme for any access structure Γ, where $N = \mathrm{poly}(n, 1/\epsilon)$.

$\mathsf{ShGen}(\mathsf{tk}, I, j, t, k, s^*, h)$:

1. If $t = 0$
   (a) Use as the $|I|$ pseudoshares the components of the parties in I from the h-th copy of the sharings of Fig. 7, Share, Step 4. a), which are stored in the tracing key.
   (b) Replace the components corresponding to minimal authorized subsets containing parties from $[j] \setminus I$ by two uniformly random values $r_0, r_s \in \mathbb{Z}_m$.
2. Otherwise
   (a) Use as the $|I|$ pseudoshares the components of the parties in I from the h-th copy of the sharings of the (j,k)-th iteration of Fig. 7, Share, Step 4. b), which are stored in the tracing key.
   (b) Replace the components corresponding to minimal authorized subsets containing parties from $[j-1] \setminus I$ by two uniformly random values $r_0, r_s \in \mathbb{Z}_m$.
3. For each minimal $A \in \Gamma$ with $A \cap I \neq \emptyset$ whose components have not been replaced, add $(-\omega, s^* - f)$ to the components of one party in $I \cap A$, where ω, f are the random values of the corresponding sharing, so that the indicator sums to 0 and the fake secret to s^*.
4. Place these components in the positions given by the permutation in the tracing key, and ensure that all $|I|$ pseudoshares match the size of actual shares by padding dummy random values to them.

$\mathsf{Trace}^R(\mathsf{tk}, I, s_0, s_1, \epsilon, \lambda)$:

1. Let $j_1 < \cdots < j_M$ be the parties in $[n] \setminus I$, and set $j_0 = 0$.
2. For $r \in \{0, \ldots, M\}$
   (a) For $h \in [N]$
      i. Select uniformly at random a bit $\beta \in \{0, 1\}$.
      ii. Compute $|I|$ pseudoshares by calling $\mathsf{ShGen}(\mathsf{tk}, I, j_r, 0, 0, s_\beta, h)$.
      iii. Query the reconstruction box R with these pseudoshares, and store whether its output bit equals β.
   (b) Let p_r be the fraction of correct outputs.
   (c) If $r \geq 1$ and $|p_r - p_{r-1}| > \frac{\epsilon}{2n}$, set $j = j_r$ and **break**.
3. If no party j has been set, output $(\emptyset, \perp)$.
4. For each bit index $k \in [\log m]$
   (a) For $h \in [N]$
      i. Select uniformly at random a bit $\beta \in \{0, 1\}$.
      ii. Compute $|I|$ pseudoshares by calling $\mathsf{ShGen}(\mathsf{tk}, I, j, 1, k, s_\beta, h)$.
      iii. Query the reconstruction box R with these pseudoshares, and store whether its output bit equals β.
   (b) Let $p_{j,k}$ be the fraction of correct outputs.
   (c) If $|p_{j,k} - p_{r-1}| \leq |p_{j,k} - p_r|$, set $\mathsf{id}_j^k = 1$.
   (d) Otherwise, set $\mathsf{id}_j^k = 0$.
5. Set $\pi = (\mathsf{id}_j^1, ..., \mathsf{id}_j^{\log m}) \in \mathbb{Z}_m$ and output $(\{j\}, \pi)$, where π is the proof for the party j.

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

1. If the identifier in the proof π matches that of the party in I contained in the verification key vk, output 1.
2. Otherwise, output 0.

Fig. 8: Outline of the ShareGeneration, Trace, and Verify algorithms of an $\emptyset$-strong traceable secret sharing scheme for any access structure Γ.

Notation:

1. Let $N = \lceil 512\, n^2(\lambda + log(4n(1+\log m)))/\epsilon^2 \rceil$ and $L = 1 + N(1 + n \log m)$.
2. Let $\mathcal{T} = \{0\} \cup \{(0,h)\}_{h\in[N]} \cup \{(j,k,h)\}_{j\in[n],k\in[\log m],h\in[N]}$ be the set of labels of the L sharings performed for each minimal authorized subset.
3. Let P denote the set of bijections from $\mathcal{T}$ to $[L]$.
4. Let $\Gamma_i = \{A \in \Gamma : A \text{ is minimal and } i \in A\}$.
5. Let $\Gamma_I = \{A \in \Gamma : A \text{ is minimal and } i \in A \text{ for some } i \in I\}$.

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

1. Share the indicator $0 \in \mathbb{Z}_m$ and the secret $s \in \mathbb{Z}_m$ according to Γ with the DNF scheme for Γ, i.e $\{(\mathsf{sh}_i^0, \mathsf{sh}_i^s)\}_{i\in[n]} \leftarrow (\Sigma(0,\Gamma), \Sigma(s,\Gamma))$.
2. Parse $(\mathsf{sh}_i^0, \mathsf{sh}_i^s)$ as $\{(\mathsf{sh}_{i,A,0}^0, \mathsf{sh}_{i,A,0}^s)\}_{A\in\Gamma_i}$ for $i \in [n]$.
3. Set $\mathsf{sh}_{i,A,0}^* \leftarrow (\mathsf{sh}_{i,A,0}^0, \mathsf{sh}_{i,A,0}^s)$ for $i \in [n]$ and $A \in \Gamma_i$.
4. Set $\mathsf{id}_i = (\mathsf{id}_i^1, ..., \mathsf{id}_i^{\log m}) \leftarrow_\$ \mathbb{Z}_m$ for $i \in [n]$.
5. Set $\mathsf{vk} \leftarrow \{\mathsf{id}_i\}_{i\in[n]}$.
6. For each minimal $A \in \Gamma$
   (a) For $h \in [N]$
      i. Set $u \leftarrow (0,h)$.
      ii. Select a random indicator and a random fake secret, i.e. $(\omega_{A,u}, f_{A,u}) \leftarrow_\$ (\mathbb{Z}_m, \mathbb{Z}_m)$.
      iii. Share them among A, i.e. $\{(\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)\}_{\ell\in A} \leftarrow (\Sigma(\omega_{A,u}, A), \Sigma(f_{A,u}, A))$.
      iv. Set $\mathsf{tk}_{A,u} \leftarrow (\omega_{A,u}, f_{A,u}, \{(\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)\}_{\ell\in A})$.
      v. Set $\mathsf{sh}_{\ell,A,u}^* \leftarrow (\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)$ for $\ell \in A$.
   (b) For $j \in [n]$, $k \in [\log m]$, and $h \in [N]$
      i. Set $u \leftarrow (j,k,h)$.
      ii. Select a random indicator and a random fake secret, i.e. $(\omega_{A,u}, f_{A,u}) \leftarrow_\$ (\mathbb{Z}_m, \mathbb{Z}_m)$.
      iii. Share them among A, i.e. $\{(\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)\}_{\ell\in A} \leftarrow (\Sigma(\omega_{A,u}, A), \Sigma(f_{A,u}, A))$.
      iv. Set $\mathsf{tk}_{A,u} \leftarrow (\omega_{A,u}, f_{A,u}, \{(\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)\}_{\ell\in A\setminus\{j\}})$.
      v. If $j \in A$ and $\mathsf{id}_j^k = 0$, replace the share of the party j by two random values $r_0, r_s \in \mathbb{Z}_m$, i.e. $(\mathsf{sh}_{j,A,u}^0, \mathsf{sh}_{j,A,u}^s) \leftarrow_\$ (\mathbb{Z}_m, \mathbb{Z}_m)$.
      vi. Set $\mathsf{sh}_{\ell,A,u}^* \leftarrow (\mathsf{sh}_{\ell,A,u}^0, \mathsf{sh}_{\ell,A,u}^s)$ for $\ell \in A$.
7. Select a bijection $\sigma \in P$ uniformly at random, i.e. $\sigma \leftarrow_\$ P$.
8. For each minimal $A \in \Gamma$
   (a) For $u \in \mathcal{T}$
      i. Set $d \leftarrow \sigma(u)$.
      ii. Set $\mathsf{sh}_{i,A,d} \leftarrow \mathsf{sh}_{i,A,u}^*$ for $i \in A$.
   (b) Set $\mathsf{sh}_{i,A} \leftarrow \{\mathsf{sh}_{i,A,d}\}_{d\in[L]}$ for $i \in A$.
9. Set $\mathsf{sh}_i \leftarrow \{\mathsf{sh}_{i,A}\}_{A\in\Gamma_i}$ for $i \in [n]$.
10. Set $\mathsf{tk} \leftarrow (\sigma, \{\mathsf{tk}_{A,u}\}_{A,u})$, where A ranges over the minimal authorized subsets of Γ and $u \in \mathcal{T} \setminus \{0\}$.
11. Output $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$.

$\mathsf{Rec}(\{\mathsf{sh}_i\}_{i\in A})$:

1. Select a minimal authorized subset $B \subseteq A$.
2. Parse sh_i as $\{\mathsf{sh}_{i,B'}\}_{B'\in\Gamma_i}$ for $i \in B$.
3. Parse $\mathsf{sh}_{i,B}$ as $\{\mathsf{sh}_{i,B,d}\}_{d\in[L]}$ for $i \in B$.
4. Parse $\mathsf{sh}_{i,B,d}$ as $(\mathsf{sh}_{i,B,d}^0, \mathsf{sh}_{i,B,d}^s)$ for $i \in B$ and $d \in [L]$.
5. For $d \in [L]$
   1. If $\sum_{i\in B} \mathsf{sh}_{i,B,d}^0 = 0$, set $s \leftarrow \sum_{i\in B} \mathsf{sh}_{i,B,d}^s$ and **break**.
6. Output s.

Fig. 9: Share and Reconstruction algorithms of an $\emptyset$-strong traceable secret sharing scheme for any access structure Γ.

$\mathsf{ShGen}(\mathsf{tk}, I, j, t, k, s^*, h)$:

1. Parse tk as $(\sigma, \{\mathsf{tk}_{A,u}\}_{A,u})$.
2. If $t = 0$, set $u \leftarrow (0, h)$ and $D \leftarrow [j] \setminus I$. Otherwise, set $u \leftarrow (j, k, h)$ and $D \leftarrow [j-1] \setminus I$.
3. Set $d \leftarrow \sigma(u)$.
4. For $A \in \Gamma_I$
   (a) If $A \cap D \neq \emptyset$, send two random values $r_0, r_s \in \mathbb{Z}_m$, i.e. $(\mathsf{sh}^0_{\ell,A,d}, \mathsf{sh}^s_{\ell,A,d}) \leftarrow_\$ (\mathbb{Z}_m, \mathbb{Z}_m)$ for $\ell \in I \cap A$.
   (b) Otherwise
      i. Parse $\mathsf{tk}_{A,u}$ as $(\omega_{A,u}, f_{A,u}, \{(\mathsf{sh}^0_{\ell,A,u}, \mathsf{sh}^s_{\ell,A,u})\}_\ell)$.
      ii. Set $(\mathsf{sh}^0_{\ell,A,d}, \mathsf{sh}^s_{\ell,A,d}) \leftarrow (\mathsf{sh}^0_{\ell,A,u}, \mathsf{sh}^s_{\ell,A,u})$ for $\ell \in I \cap A$.
      iii. Set $\ell_A \leftarrow \min(I \cap A)$ and shift its components so that the indicator sums to 0 and the fake secret to s^*, i.e. $(\mathsf{sh}^0_{\ell_A,A,d}, \mathsf{sh}^s_{\ell_A,A,d}) \leftarrow (\mathsf{sh}^0_{\ell_A,A,d} - \omega_{A,u}, \mathsf{sh}^s_{\ell_A,A,d} + s^* - f_{A,u})$.
   (c) For $d' \in [L] \setminus \{d\}$ and $\ell \in I \cap A$, send two random values $r_0, r_s \in \mathbb{Z}_m$, i.e. $(\mathsf{sh}^0_{\ell,A,d'}, \mathsf{sh}^s_{\ell,A,d'}) \leftarrow_\$ (\mathbb{Z}_m, \mathbb{Z}_m)$.
   (d) Set $\mathsf{sh}_{\ell,A} \leftarrow \{(\mathsf{sh}^0_{\ell,A,d'}, \mathsf{sh}^s_{\ell,A,d'})\}_{d' \in [L]}$ for $\ell \in I \cap A$.
5. Set $\mathsf{sh}_\ell \leftarrow \{\mathsf{sh}_{\ell,A}\}_{A \in \Gamma_\ell}$ for $\ell \in I$.
6. Output $\{\mathsf{sh}_\ell\}_{\ell \in I}$.

$\mathsf{Trace}^R(\mathsf{tk}, I, s_0, s_1, \epsilon, \lambda)$:

1. Let $j_1 < \cdots < j_M$ be the parties in $[n] \setminus I$, and set $j_0 \leftarrow 0$.
2. For $r \in \{0, \ldots, M\}$
   (a) Set $c \leftarrow 0$.
   (b) For $h \in [N]$
      i. Set $\beta \leftarrow_\$ \{0, 1\}$.
      ii. Set $\{\mathsf{sh}_i\}_{i \in I} \leftarrow \mathsf{ShGen}(\mathsf{tk}, I, j_r, 0, 0, s_\beta, h)$.
      iii. Query R on $\{\mathsf{sh}_i\}_{i \in I}$. Let b_h be its response.
      iv. If $b_h = \beta$, set $c \leftarrow c + 1$.
   (c) Set $p_r \leftarrow \frac{c}{N}$.
   (d) If $r \geq 1$ and $|p_r - p_{r-1}| > \frac{\epsilon}{2n}$, set $j \leftarrow j_r$ and **break**.
3. If no party j has been set, output $(\emptyset, \perp)$.
4. For $k \in [\log m]$
   (a) Set $c \leftarrow 0$.
   (b) For $h \in [N]$
      i. Set $\beta \leftarrow_\$ \{0, 1\}$.
      ii. Set $\{\mathsf{sh}_i\}_{i \in I} \leftarrow \mathsf{ShGen}(\mathsf{tk}, I, j, 1, k, s_\beta, h)$.
      iii. Query R on $\{\mathsf{sh}_i\}_{i \in I}$. Let b_h be its response.
      iv. If $b_h = \beta$, set $c \leftarrow c + 1$.
   (c) Set $p_{j,k} \leftarrow \frac{c}{N}$.
   (d) If $|p_{j,k} - p_{r-1}| \leq |p_{j,k} - p_r|$, set $\mathsf{id}^k_j \leftarrow 1$.
   (e) Otherwise, set $\mathsf{id}^k_j \leftarrow 0$.
5. Set $\mathsf{id}'_j \leftarrow (\mathsf{id}^1_j, ..., \mathsf{id}^{\log m}_j)$.
6. Set $\pi \leftarrow (j, \mathsf{id}'_j)$.
7. Output $(\{j\}, \pi)$.

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

1. Parse vk as $\{\mathsf{id}_i\}_{i \in [n]}$, I as j, and π as (j, id'_j).
2. If $\mathsf{id}'_j = \mathsf{id}_j$, output 1.
3. Otherwise, output 0.

Fig. 10: $\mathsf{ShareGeneration}$, Trace, and Verify algorithms of an $\emptyset$-strong traceable secret sharing scheme for any access structure Γ.

We now analyze statistical correctness. Let $A \in \Gamma$ be an authorized subset, and let $B \subseteq A$ be the minimal authorized subset selected by the reconstruction algorithm. By construction, we have

$$0 = \sum_{i \in B} \mathsf{sh}^0_{i,B,d} \quad \text{and} \quad s = \sum_{i \in B} \mathsf{sh}^s_{i,B,d},$$

where $d = \sigma(0)$. Therefore, the reconstruction algorithm fails only if there exists another $d' \in [L]$ with $d' \neq d$ such that $\sum_{i \in B} \mathsf{sh}^0_{i,B,d'} = 0$, mistakenly identifying d' as the position of the sharing of the real secret. However, the indicator components at each such position add up to the uniformly random value $\omega_{B,u}$ where $u = \sigma^{-1}(d')$, or to a uniformly random value if the share of a party has been replaced. Hence, the probability of this event is at most L/m, which is negligible in λ since L is polynomial in λ and $m = 2^{\Theta(\lambda)}$.

Finally, for the share size it suffices to notice that each party receives L pairs of elements of $\mathbb{Z}_m$ for each minimal authorized subset containing it, and that the number of minimal authorized subsets coincides with the size d of the DNF representation of Γ. Since $\lambda = \Theta(\log m)$ and $\lambda = \Omega(\mathrm{poly}(n))$, we have that $N = O(n^2 \log m/\epsilon^2)$ and $L = O(n^3 \log^2 m/\epsilon^2)$, so the total share size per party is $O(d \cdot L \log m) = O(d \cdot n^3 \log^3 m/\epsilon^2)$. □

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

Proposition 3.3. *Let Γ be an 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 $s \in \mathbb{Z}_m$, the secret sharing scheme of Fig. 9 and Fig. 10 has tracer privacy.*

Proof. The tracing key consists of the permutation σ and, for each additional sharing, of the random values $\omega_{A,u}$ and $f_{A,u}$ and of some of its share components, which are all generated independently of the secret. In particular, it does not contain any component of the sharing of the real secret. Therefore, for any unauthorized subset B, the components of the parties in B in the sharing of the real secret are uniformly random and independent of the tracing key and of the remaining components of their shares, as shown in the proof of Proposition 3.2. Consequently, the joint distribution of the shares of B and the tracing key does not depend on the secret and the perfect tracer privacy requirement holds. □

Now, we move to prove that the scheme achieves statistical traceability.

Proposition 3.4. *Let Γ be an 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 $s \in \mathbb{Z}_m$, the secret sharing scheme of Fig. 9 and Fig. 10 has statistical traceability and requires $O(n^2(n + \log m) \log m/\epsilon^2)$ queries.*

Proof. The upper bound on the number of queries follows from the algorithm description, since Trace makes at most $(n + 1 + \log m)N$ queries and $N = O(n^2 \log m/\epsilon^2)$.

Consider an execution of the experiment in Fig. 2. The adversary can win in one of two ways: given a $\mathsf{GoodBox}$, the tracer either outputs a valid accusation against a party outside J, or an invalid accusation against a party in J. We bound the probability of each event separately.

We start by bounding the probability that the tracer outputs a valid accusation against a party outside J. Let $j \in [n] \setminus J$. By construction, the identifier id_j only affects the components of j in the sharings of Step 6 (b) of Share generated for j, which are included neither in the tracing key nor in the shares of J. Since R only depends on the shares of J and the random coins of the adversary, the output of the tracer is independent of id_j. Hence, the tracer outputs a valid accusation against a party outside J with probability at most $1/m$.

Next, we bound the probability that $\mathsf{GoodBox}$ holds and the tracer does not output a valid accusation against a party in J. To this end, we fix the shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i \in J}$ of the corrupted parties and the random coins of the adversary, so that the reconstruction box R and its label I are also fixed, and we assume that $I \cap J = \emptyset$, $I \notin \Gamma$, and $J \notin \Gamma$. As in Fig. 10, we denote by $j_1 < \cdots < j_M$ the parties in $[n] \setminus I$, and for $r \in \{0, \ldots, M\}$ we let $D_r = \{j_1, \ldots, j_r\}$ be the set of parties that are disabled in the pseudoshares $\mathsf{ShGen}(\mathsf{tk}, I, j_r, 0, 0, s^*, h)$.

First, notice that the corrupted parties cannot learn anything about the permutation σ nor about the additional sharings that they do not hold. Indeed, since J is unauthorized, every minimal authorized subset contains a party outside J. Hence, as shown in the proof of Proposition 3.2, the components of J are uniformly random and independent, both in the sharing of the real secret and in the additional sharings, and this is also the case for the components of a party $j \in J$ that have been replaced with uniformly random values. Therefore, the shares of J are independent of the secret, the identifiers, and the permutation σ, which only determines the positions of the components. Moreover, conditioned on $\mathbf{sh}$ and σ, the genuine components (that is, before any replacement) of the additional sharings that are not included in $\mathbf{sh}$ are uniformly random and independent, since each additional sharing is an independent additive sharing of a uniformly random pair among a minimal authorized subset.

Second, we characterize the distribution of the pseudoshares generated by the tracer. For a subset $D \subseteq [n] \setminus I$, a position $d \in [L]$, and a bit $\beta \in \{0, 1\}$ let $\mathcal{X}_{D,d,\beta}$ be the distribution of $|I|$ pseudoshares whose components for each minimal authorized subset $A \in \Gamma_I$ are sampled independently as follows:

- At position d, if $A \cap D = \emptyset$ and $A \subseteq I \cup J$, the components of the parties in $I \cap A$ are uniformly random subject to adding up, together with the components of the parties in $J \cap A$ at position d, to $(0, s_\beta)$.
- At any other position, and also at position d if $A \cap D \neq \emptyset$ or $A \not\subseteq I \cup J$, the components are uniformly random.

Note that $J \cap A \neq \emptyset$ for every $A \in \Gamma_I$ with $A \subseteq I \cup J$, since I is unauthorized. We claim that, conditioned on $\mathbf{sh}$ and σ, for every $h \in [N]$ and every $\beta \in \{0, 1\}$, the output of $\mathsf{ShGen}(\mathsf{tk}, I, j_r, 0, 0, s_\beta, h)$ is distributed according to $\mathcal{X}_{D_r,d,\beta}$, where $d = \sigma((0, h))$, for every $r \in \{0, \ldots, M\}$. Moreover, for every $r \in [M]$ and $k \in [\log m]$, the output of $\mathsf{ShGen}(\mathsf{tk}, I, j_r, 1, k, s_\beta, h)$ is distributed

according to $\mathcal{X}_{D_{r-1},d,\beta}$ if $\mathsf{id}^k_{j_r} = 1$, and according to $\mathcal{X}_{D_r,d,\beta}$ if $\mathsf{id}^k_{j_r} = 0$, where $d = \sigma((j_r, k, h))$.

We start with the pseudoshares used to detect a corrupted party, and we let $u = (0, h)$. By construction, for each $A \in \Gamma_I$, the pseudoshares contain uniformly random values at every position different from d, and also at position d if $A \cap D_r \neq \emptyset$. Otherwise, they contain the genuine components of the parties in $I \cap A$ in the sharing of $(\omega_{A,u}, f_{A,u})$, where the component of ℓ_A is shifted by $(-\omega_{A,u}, s_\beta - f_{A,u})$. Since the genuine components of all the parties in A add up to $(\omega_{A,u}, f_{A,u})$, after the shift they add up to $(0, s_\beta)$. Hence, if $A \subseteq I \cup J$, the components of the parties in $I \cap A$ are uniformly random subject to adding up, together with the components of $J \cap A$ at position d, to $(0, s_\beta)$. In contrast, if $A \not\subseteq I \cup J$, there exists a party $v \in A \setminus (I \cup J)$ whose genuine component is uniformly random and independent of all the other values, and since the shifted component of ℓ_A depends on it, the components of the parties in $I \cap A$ are uniformly random and independent. This is precisely the distribution $\mathcal{X}_{D_r,d,\beta}$.

The pseudoshares used to recover the identifier are analyzed with the same argument, applied to the label $u = (j_r, k, h)$ and the set of disabled parties $[j_r - 1] \setminus I = D_{r-1}$, together with the following observation. For every $A \in \Gamma_I$ with $j_r \in A$, if $\mathsf{id}^k_{j_r} = 0$, the share of j_r in this sharing has been replaced with uniformly random values, so its genuine component is not included in $\mathbf{sh}$, even if $j_r \in J$. Hence, the shifted component of ℓ_A is uniformly random and independent of all the other values, exactly as if j_r was disabled. If $\mathsf{id}^k_{j_r} = 1$, the sharing of A is an ordinary additional sharing, and the same holds for every A with $j_r \notin A$. Notice also that if $j_r \notin J$, then $\mathcal{X}_{D_{r-1},d,\beta}$ and $\mathcal{X}_{D_r,d,\beta}$ coincide, since every A containing j_r satisfies $A \not\subseteq I \cup J$. Finally, since different copies correspond to independent sharings, pseudoshares generated from different copies are independent and the uniformly random values used by $\mathsf{ShareGeneration}$ are sampled independently.

Third, we relate the behavior of R on these distributions with its behavior on genuine shares. We denote by $\mu_D(d)$ the probability that $R(X) = \beta$, where $\beta \leftarrow_\$ \{0,1\}$ and $X \leftarrow \mathcal{X}_{D,d,\beta}$, and the probability is also taken over the random coins of R. Moreover, we let $\overline{\mu}_D = \frac{1}{L}\sum_{d\in[L]} \mu_D(d)$, and, for $\beta \in \{0,1\}$ we let a_β be the probability that $R(\{\mathsf{sh}'_i\}_{i\in I}) = \beta$, where the probability is taken over $(\{\mathsf{sh}'_i\}_{i\in[n]}, \mathsf{tk}', \mathsf{vk}') \leftarrow \mathsf{Share}(s_\beta, \Gamma)$ conditioned on $\{\mathsf{sh}'_i\}_{i\in J} = \mathbf{sh}$, and the random coins of R. We claim that the following holds.

1. $\overline{\mu}_{D_0} = \frac{a_0+a_1}{2}$.
2. $\mu_{D_M}(d) = \frac{1}{2}$ for every $d \in [L]$.
3. If $j_r \notin J$, then $\mu_{D_r}(d) = \mu_{D_{r-1}}(d)$ for every $d \in [L]$.

Intuitively, the first item states that R behaves the same when queried on fake shares generated by the tracer with all parties enabled as it does on genuine shares. The second item states that, when queried on fake shares generated with all parties disabled, R has no distinguishing advantage. Finally, the third item states that disabling an honest party cannot, by itself, cause a change in the behavior of the box.

For proving the first item, consider a sharing $(\{\mathsf{sh}'_i\}_{i\in[n]}, \mathsf{tk}', \mathsf{vk}') \leftarrow \mathsf{Share}(s_\beta, \Gamma)$ conditioned on $\{\mathsf{sh}'_i\}_{i\in J} = \mathbf{sh}$, and let σ' be its permutation. As shown above, σ'

is uniformly random and independent of **sh**, so the position $\sigma'(0)$ of the sharing of the secret is uniformly random in $[L]$. Moreover, conditioned on $\sigma'(0) = d$, the shares of I are distributed according to $\mathcal{X}_{\emptyset,d,\beta}$. Indeed, at position d, the components of the parties in $I \cap A$ add up to $(0, s_\beta)$ together with those of $J \cap A$ if $A \subseteq I \cup J$, and they are uniformly random otherwise, by the privacy of the additive scheme. At any other position, the components belong to additional sharings, so they are uniformly random. Hence, $a_\beta = \frac{1}{L}\sum_{d\in[L]} \mathbf{P}[R(X) = \beta]$, where $X \leftarrow \mathcal{X}_{\emptyset,d,\beta}$, and, since $D_0 = \emptyset$, averaging over $\beta \in \{0,1\}$ gives the first item.

For the second item, notice that, since I is unauthorized, every $A \in \Gamma_I$ contains a party outside I. Hence, $A \cap D_M \neq \emptyset$ for every $A \in \Gamma_I$, so $\mathcal{X}_{D_M,d,\beta}$ consists of uniformly random values and does not depend on β. Therefore, R outputs β with probability exactly $\frac{1}{2}$, regardless of its strategy.

The third item follows from the fact that, as shown above, $\mathcal{X}_{D_{r-1},d,\beta}$ and $\mathcal{X}_{D_r,d,\beta}$ coincide whenever $j_r \notin J$.

With this in mind, we are now ready to bound the probability that GoodBox holds and the tracer does not output a valid accusation against a party in J. Notice that, by Definition 2.10, R is $(\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)$-minimally-useful if and only if $a_b \geq \frac{1}{2} + \epsilon$. We distinguish two cases.

First, assume that $a_\beta < \frac{1}{2} + \epsilon$ for some $\beta \in \{0,1\}$. Then, GoodBox can only hold if $b = 1 - \beta$. By the privacy of the scheme, b is uniformly random and independent of **sh**, so GoodBox holds with probability at most $\frac{1}{2}$.[8]

Now, assume that $a_0, a_1 \geq \frac{1}{2} + \epsilon$. We first show that, except with probability at most $e^{-\lambda}$, all the fractions computed by the tracer are within η of their expected values. Consider the fraction p_r for some $r \in \{0, \ldots, M\}$. In its computation, the h-th query is generated from the copy with label $(0, h)$ and a fresh uniformly random bit β. As shown above, conditioned on **sh** and σ, the queries are independent, and the h-th one is answered correctly with probability $\mu_{D_r}(\sigma((0,h)))$. Since R is stateless, its outputs on these N queries are independent. Hence, by the Chernoff bound,

$$\left| p_r - \frac{1}{N} \sum_{h\in[N]} \mu_{D_r}(\sigma((0,h))) \right| \leq \frac{\eta}{2}$$

except with probability at most $2e^{-N\eta^2/2}$. Moreover, σ is uniformly random and independent of **sh**, so $\{\sigma((0,h))\}_{h\in[N]}$ is a uniformly random subset of N positions in $[L]$. Hence, by the Chernoff bound for sampling without replacement,

$$\left| \frac{1}{N} \sum_{h\in[N]} \mu_{D_r}(\sigma((0,h))) - \overline{\mu}_{D_r} \right| \leq \frac{\eta}{2}$$

except with probability at most $2e^{-N\eta^2/2}$. Therefore, $|p_r - \overline{\mu}_{D_r}| \leq \eta$ except with probability at most $4e^{-N\eta^2/2}$. The same argument, using the copies with labels

[8] Intuitively, this case takes into account any reconstruction box strategy that accounts for working only on one of the two possible secrets.

(j,k,h), shows that if the tracer stops at the party $j = j_r$, then $|p_{j,k} - \overline{\mu}_{D_{r-1}}| \leq \eta$ if $\mathsf{id}_j^k = 1$ and $|p_{j,k} - \overline{\mu}_{D_r}| \leq \eta$ if $\mathsf{id}_j^k = 0$ except with probability at most $4e^{-N\eta^2/2}$. The tracer computes at most $M+1 \leq n$ fractions p_r, and at most $\log m$ fractions $p_{j,k}$ for the party j at which it stops. Since this party can be any party in $[n] \setminus I$, it suffices to consider at most $n(1+\log m)$ fractions. Therefore, by the union bound and the choice of N, all of them are within η of their expected values except with probability at most $4n(1+\log m)e^{-N\eta^2/2} \leq e^{-\lambda}$.

We now show that, in this case, the tracer outputs a valid accusation against a party in J. By the previous claims, $\overline{\mu}_{D_0} = \frac{a_0+a_1}{2} \geq \frac{1}{2} + \epsilon$, $\overline{\mu}_{D_M} = \frac{1}{2}$, and $\overline{\mu}_{D_r} = \overline{\mu}_{D_{r-1}}$ whenever $j_r \notin J$. Since $I \cap J = \emptyset$, the at most n parties of J are among $j_1, \ldots, j_M$, so there exists r with $j_r \in J$ such that $\overline{\mu}_{D_{r-1}} - \overline{\mu}_{D_r} \geq \epsilon/n = 16\eta$. For this r, we have $|p_r - p_{r-1}| \geq 14\eta > \frac{\epsilon}{2n}$, while for every r' with $j_{r'} \notin J$, we have $|p_{r'} - p_{r'-1}| \leq 2\eta < \frac{\epsilon}{2n}$. Hence, the tracer stops at a step r such that $j = j_r \in J$ and $|\overline{\mu}_{D_{r-1}} - \overline{\mu}_{D_r}| > \frac{\epsilon}{2n} - 2\eta = 6\eta$. Then, for each $k \in [\log m]$, if $\mathsf{id}_j^k = 1$, we have $|p_{j,k} - p_{r-1}| \leq 2\eta$ and $|p_{j,k} - p_r| \geq 6\eta - 2\eta = 4\eta$, so the tracer sets the k-th bit to 1. Analogously, if $\mathsf{id}_j^k = 0$, it sets the k-th bit to 0. Therefore, the tracer outputs $(\{j\}, (j, \mathsf{id}_j))$, which is a valid accusation against $j \in J$. Consequently, in this case, the probability that $\mathsf{GoodBox}$ holds but the tracer fails to output a valid accusation against a party in J is at most $e^{-\lambda}$.

Therefore, since $e^{-\lambda} \leq \frac{1}{2}$, combining the two cases shows that the probability that the tracer fails to output a valid accusation against a party in J is at most $\frac{1}{2}$.

Finally, combining the bounds on the two ways in which the adversary can win, we conclude that the experiment in Fig. 2 outputs 1 with probability at most $\frac{1}{2} + \frac{1}{m}$. This proves that the scheme achieves statistical $\emptyset$-strong traceability. □

We conclude the proof of Theorem 3.1 by proving that the scheme satisfies statistical non-imputability.

Proposition 3.5. *Let Γ be an 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 $s \in \mathbb{Z}_m$, the secret sharing scheme of Fig. 9 and Fig. 10 has statistical non-imputability.*

Proof. Let $\mathcal{A}$ be an adversary in the experiment of Fig. 4, which receives the shares of all the parties except i^*, and the tracing key. By construction, the identifier id_{i^*} only affects the components of i^* in the sharings of Step 6 (b) of Share generated for i^*, which are either genuine or uniformly random depending on the bits of id_{i^*}. None of these components is given to $\mathcal{A}$, since they are not included in the tracing key. Moreover, all the other values received by $\mathcal{A}$, including the random values $\omega_{A,u}, f_{A,u}$ of these sharings, are independent of id_{i^*}, since they only depend on the genuine components. Hence, the view of $\mathcal{A}$ is independent of id_{i^*}, which is uniformly random in $\mathbb{Z}_m$, and the probability that $\mathcal{A}$ outputs a valid proof against i^* is at most $1/m$, which is negligible in λ. □

4 An Information-theoretic Hiding Mildly Strong Traceable Scheme

In this section, we construct an information-theoretic traceable secret sharing scheme for general access structures under a relaxed yet meaningful variant of the strong traceability model, which we call *hidden mildly strong traceability*. We first introduce the model and discuss its limitations as well as its relationship to the hidden traceability model. We then outline the main ideas behind our construction and subsequently present the complete scheme. Finally, we present further optimizations for it.

4.1 The Mildly Strong Traceability Model

An impossibility result. As noted in Remark 2.14, Goyal, Jain, and Partap [GJP26] established an impossibility result for the strong traceability model. Specifically, they showed that no traceable secret sharing scheme for general access structures can simultaneously achieve strong traceability together with tracer secrecy or strong non-imputability. Their argument relies on the following key observation: since $I \cap J \neq \emptyset$ and the reconstruction box requires queries containing the shares of all parties in I, any tracing procedure must be able to generate, using only the tracing key, the exact shares of the parties in the intersection $I \cap J$. This requirement inherently violates tracer secrecy or non-imputability, leading to the impossibility result.

Relaxing the Strong Traceability Model. At first glance, this impossibility result may appear discouraging, as it seems to give corrupted parties a perfect opportunity to misbehave without facing consequences. However, a closer examination of the strong traceability model reveals that it is far less attractive for adversaries in practice. In particular, the model has two inherent limitations stemming from the fact that the reconstruction box only functions on input queries that include the shares of all parties in the label I.

First, a minimally-useful reconstruction box is only useful for a single unauthorized subset, namely the one specified by its label I. As a consequence, such a box has at most one potential buyer: the subset I itself. This greatly reduces its appeal, since that subset would be uniquely identifiable as the only group capable of using the box. In other words, minimally-useful reconstruction boxes effectively reveal which specific set of parties should be monitored by the authorities, thereby discouraging their use or purchase.

Second, when combined with the condition $I \cap J \neq \emptyset$, the situation becomes even more restrictive. The unique potential buyer subset I necessarily contains at least one of the parties that helped create the reconstruction box R. As a result, the corrupted parties J cannot sell a minimally-useful reconstruction box to a completely independent group of parties.

To illustrate these limitations, consider a t-threshold access structure and a corrupted set J of size $t-1$. Ideally, these corrupted parties would like to build a

reconstruction box that works for any single party outside J. That is, a box that reconstructs the secret whenever it is given one additional share from outside the corrupted set. However, in the strong traceability model this is not possible. The corrupted parties are instead forced to construct a separate reconstruction box for each party outside J. Moreover, for any such box to function, the input must include not only the share of that external party but also at least one redundant share from within J. Clearly, this requirement is highly artificial and does not reflect realistic adversarial behavior.

Therefore, a more realistic model for a corrupted set J is one in which the reconstruction box carries a label I with $I \cap J \neq \emptyset$, but is able to operate when queried with shares from any subset $I' \subseteq I$ such that $I' \cup J$ is authorized. This behavior better captures how adversaries would actually deploy a reconstruction box.

In particular, this model matches exactly the goal of the corrupted set of size $t-1$ in the t-threshold example above: the corrupted parties could construct a single reconstruction box labeled by a set I that successfully reconstructs the secret when queried with the share of any single party in $I \setminus J$.

This relaxation of the reconstruction box's definition is the key idea underlying the mildly strong traceability model. We now give the formal definition of a mildly-good reconstruction box.

Definition 4.1 (Mildly-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 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 $J \notin \Gamma$, a corresponding set of shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i \in J}$, a label $I \subseteq [n]$, and any subset $I' \subseteq I$ with $I' \cup J \in \Gamma$, a reconstruction box R is $(\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)$-mildly-good if*

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

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

Remark 4.2. Intuitively, the previous definition states that the box is mildly-good for the subset I if for all subsets $I' \subseteq I$ such that $I' \cup J$ is authorized, the box works when queried with the shares of I'.

Once the definition of mildly-good reconstruction box is in place, the notion of mildly strong traceability follows straightforward.

Definition 4.3 (Perfect Strong Traceability). *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 traceable scheme for Γ. For any $\epsilon \in [0, \frac{1}{2}]$, Σ is perfect mildly strong traceable if for every adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathrm{mildlytrac}}_{\mathcal{A},\Sigma,\epsilon} \coloneqq \mathbf{P}[\mathbf{ExpMildlyTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = \frac{1}{2}.$$

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

1. $(\Gamma, J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $b \leftarrow_{\$} \{0,1\}$.
3. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$.
4. $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$.
5. $(R, I) \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
6. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
7. $\mathsf{GoodBox} := (J \notin \Gamma) \wedge (I \notin \Gamma) \wedge (R \text{ is } (\Gamma, I, \mathbf{sh}, s_0, s_1, b, \epsilon)\text{-mildly-good})$.
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. 11: The tracing experiment for a mildly strong traceable secret sharing scheme Σ for an access structure Γ. The steps highlighted in blue indicate the differences compared to the tracing experiment of Fig. 3.

Remark 4.4. Notice that the previous definition sets the adversary's success probability to $\frac{1}{2}$ rather than 0 to rule out the trivial attack in which the reconstruction box always outputs either s_0 or s_1 independently of the input queries received.

Regarding non-imputability, the mildly strong model considers the same definition as the $\emptyset$-strong and strong models, which is presented in Fig. 4 and Definition 2.13.

Limitations of the Mildly Strong Traceability Model. Given that the strong traceability model is subject to an impossibility result, it is natural to ask whether a similar limitation applies to the mildly strong model. In this case, the original impossibility argument no longer applies, since the tracer is not required to query the reconstruction box using shares from the corrupted set J.

However, a natural question arises: how can the tracer know which parties are in $I \cap J$? By definition, the tracer has no extra knowledge of the label I. Without additional capabilities, apparently, the only option is to try all possible subsets of I until finding one that does not intersect with J, which for some choices of Γ, I, and J may require an exponential number of attempts on average. This observation highlights an important limitation: any traceable secret sharing scheme for general access structures, without extra assumptions or properties, cannot support an efficient tracing procedure. An example where this situation occurs follows.

Example 4.5. Consider an access structure Γ over n parties with a minimal authorized subset A of size $\frac{n}{2}$, and let $J \subset A$ be a corrupted subset of size $\frac{n}{4}$

that constructs a mildly-good reconstruction box R labeled by $I = A \setminus \{j\}$ for some $j \in J$. In this scenario, on average, the tracer must try an exponential number of queries to R before finding a meaningful one, i.e. a query that does not intersect J. Indeed, there is only one subset of I that is useful, namely $I \setminus J$.

A similar issue was noted by Boneh, Rotem, and Partap [BPR24] when constructing traceable schemes for threshold access structures. They addressed the problem by introducing anonymity into the traceable scheme. Specifically, shares were made unlinkable to particular parties, preventing corrupted parties from determining which parties contributed to the shares used in the tracer's input queries.

Therefore, one might be tempted to add anonymity to the tracing scheme for the $\emptyset$-strong model from Section 3 to bypass this limitation. However, such a naive approach fails. The tracing method in Section 3, as well as that of Goyal, Jain, and Partap [GJP26], relies on the tracer iteratively disabling all parties except the honest ones within I. While anonymity prevents the reconstruction box from rejecting queries based on party identities, it does not solve the core problem: the tracer still needs to know which parties in I are corrupted in order to perform the disabling step effectively.

Fortunately, this limitation can be overcome by leveraging another extra assumption in the literature of traceable secret sharing: hiding the access structure. By definition, the tracer knows that there exists at least one honest party $i \in I \setminus J$, since $I \cup J$ is authorized while J is not. Therefore, if the parties do not know the exact access structure, the tracer can then craft queries that make the reconstruction box believe it was built for an access structure in which $J \cup \{i\}$ is authorized. This reduces the search space: the tracer now needs to try at most only $|I|$ different input queries, each using a single share, to find a meaningful query (one that simulates the honest party i). Once this honest party is identified, the tracing strategy of Farràs and Guiot [FG26] can be adapted to work in this model, at the cost of making the access structure hidden from the parties.

4.2 The Hidden Mildly Strong Traceability Model.

The previous limitation motivates the definition of the *hidden mildly strong traceability model*, which differs from the previous mildly strong model in one key aspect: the access structure is hidden from the parties. Instead of knowing the exact access structure, the parties are only aware of a family of access structures to which the actual one belongs. As usual, we first present the notion of *hidden-mildly-good reconstruction box*.

Definition 4.6 (Hidden-mildly-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 $J \notin \Gamma$, a corresponding set of shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i \in J}$, and a label $I \subseteq [n]$, a reconstruction box R is $(\mathcal{F}, \mathbf{sh}, s_0, s_1, b, \epsilon)$-hidden-mildly-good if for any access*

structure $\Gamma' \in \mathcal{F}$ *that is consistent with* J*, any unauthorized subset* $I' \notin \Gamma'$ *with* $I' \subseteq I$ *such that* $J \cap I' = \emptyset$ *and* $J \cup I' \in \Gamma'$*, it holds that*

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

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

Once the notion of hidden-mildly-good reconstruction boxes is established, we present the definition of hidden mildly strong traceability. Again, these definitions closely follow those of the mildly strong traceability setting, with the main difference being that the access structure is now hidden from the parties. To highlight the differences introduced in this setting, all modifications are marked in blue. We start with the notion of hidden traceability.

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

1. $(J, s_0, s_1, \mathsf{state}) \leftarrow \mathcal{A}(\mathcal{F})$.
2. $\Gamma \leftarrow_{\$} \{\Gamma' \in \mathcal{F} : J \notin \Gamma'\}$ ▷ The subset J must be unauthorized in Γ.
3. $b \leftarrow_{\$} \{0, 1\}$.
4. $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s_b, \Gamma)$.
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. $\mathsf{GoodBox} \coloneqq (J \notin \Gamma) \wedge (R \text{ is } (\mathcal{F}, \mathbf{sh}, s_0, s_1, b, \epsilon)\text{-hidden-mildly-good})$.
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. 12: The tracing experiment for a hidden mildly strong 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. 11.

Definition 4.7 (Perfect Hidden Mildly Strong 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 hidden mildly strong traceable if for every adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathrm{hiddenmildlytrac}}_{\mathcal{A},\Sigma,\epsilon} \coloneqq \mathbf{P}[\mathbf{ExpHiddenMildlyTrace}_{\mathcal{A},\Sigma,\epsilon} = 1] = \frac{1}{2}.$$

The non-imputability notion of the hidden mildly strong model is the same definition as the hidden model, which is presented in Fig. 6 and Definition 2.17.

Remark 4.8. Note that the hidden traceability model of Farràs and Guiot [FG26] can be seen as a special case of the hidden mildly strong traceability model just described, corresponding to $I = [n]$. In their model, the reconstruction box does not have an explicit label I. Instead, it must handle any input query in which the parties involved, together with the corrupted parties that built the box, form an authorized subset.

4.3 Minimal Covering Families of Access Structures

In the previous subsection, we showed how the tracer can overcome its lack of knowledge of $I \cap J$ by hiding the access structure and revealing only a family of possible structures. However, this strategy requires that, for every unauthorized subset A in any structure in the family and every party $i \notin A$, some structure in the family has $A \cup \{i\}$ as a minimal authorized subset. This condition ensures that the tracer can always simulate a minimal authorized set using a query with only one additional share.

To formalize the class of families where our tracing approach is applicable, we introduce the notion of a *minimal covering family* of access structures, which is slightly more restrictive than the notion of covering families introduced by Farràs and Guiot [FG26]. The definition follows.

Definition 4.9. *A family of access structures over n parties $\mathcal{F}$ is a* minimal covering family *if for every $\Gamma \in \mathcal{F}$, every subset $A \notin \Gamma$, and every party $i \in [n] \backslash A$ there exists $\Gamma' \in \mathcal{F}$ such that $A \cup \{i\}$ is minimal in Γ'.*

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

Lemma 4.10. *For every $n \in \mathbb{N}$, the family of all access structures over n parties is a minimal covering family.*

Proof. Notice that for any $A \subsetneq [n]$ and any $i \in [n] \setminus A$, there exists an access structure over n parties in which $A \cup i$ is a minimal authorized subset. □

Remark 4.11. In addition to the class of all access structures, various well-studied families such as threshold, weighted threshold, and hierarchical access structures also satisfy the minimal covering property. As a result, our tracing technique naturally applies to these families as well, which extends its scope to settings where parties may have partial knowledge of the access structure.

4.4 An Overview of the Algorithms

Our scheme combines the idea from Section 4.1 of using the hiding property to generate meaningful input queries with the tracing technique of [FG26]. For simplicity, we assume the tracer knows the exact number of corrupted parties c; Section 4.7 explains how to remove this assumption. We now describe the main ideas of each algorithm in our construction.

Share. We start by sharing the secret s among the parties according to the access structure Γ, using the hiding scheme of [FG26] instantiated over $\mathbb{Z}_m^2$. Each party i is also assigned 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 included in the tracing key to identify corrupted parties, while the second component id_i^2 is included in the verification key to validate the tracer's proof.

Next, for each unauthorized subset $A \notin \Gamma$ and party $i \in [n] \setminus A$, we uniformly sample $j \in A$. For each bit $b \in \{0,1\}$ of id_j, we independently share the fake secret s_b among $A \cup \{i\}$ using the additive scheme. The share given to i is stored in the tracing key and later reveals the corresponding bit of id_j, identifying the corrupted party. Party i receives a uniformly random real share, ensuring that honest reconstruction does not reveal s_b. Thus, any unauthorized set A can be extended by one party to form a minimal authorized set, enabling the tracer to issue meaningful single-share queries.

Each of these fake sharings is performed N times independently, where N is polynomial in n, and in the inverse of the distinguishing advantage ϵ of the reconstruction box. To account for an imperfect box, the tracer recovers each bit of id_j by querying it multiple times and taking a majority vote. For this amplification to work, each query must use a different copy of the fake sharings. Reusing the same copy would expose identical share components in every query, so the majority vote would merely reproduce the box's behavior on those components, regardless of its success probability on genuine shares. This yields a total of $2N\lceil \log m \rceil$ distinct sharings of the hiding scheme, each of which we call a layer.

Finally, observe that access structures in the covering family $\mathcal{F}$ may have different numbers of unauthorized subsets. Since we assign tracing identifiers for every unauthorized subset of the chosen access structure Γ and for every party outside each subset, the total number of share components could leak information about Γ. To prevent this, we pad each party's share with dummy random values until all shares have the same size, equal to the maximum over all access structures in $\mathcal{F}$. This ensures uniform share sizes and preserves the secrecy of Γ.

Intuitively, this procedure makes the unauthorized set A perceive the shares encoding the identifier id_j as genuine shares of the secret s for the minimal authorized set $A \cup \{i\}$, for any $i \in [n] \setminus A$. Since these shares are indistinguishable from real ones, the reconstruction box cannot tell whether it is processing shares of s or of id_j. Consequently, for each bit b of id_j, when queried with the pseudoshare of party i, the box treats it as a valid share of s for $A \cup \{i\}$ and outputs s_b. Repeating this for all bits allows the tracer to recover id_j and identify a corrupted party.

Reconstruction. This algorithm is identical to that of the hiding general scheme of [FG26], since the secret is shared in the same way.

ShareGeneration. At a high level, this step uses the tracing key to construct a pseudoshare for an honest party i corresponding to the k-th bit of the identifier that is indistinguishable from a genuine share. We then pad it with sufficiently many dummy random values so that it matches the size of a full share. The resulting pseudoshare is later used to fool the reconstruction box during the tracing procedure.

More concretely, let c be the number of corrupted parties, let k index a bit of the identifier, and let $h \in [N]$ index a copy of the fake sharings. During tracing, the tracer runs $\mathsf{ShareGeneration}$ to construct a full pseudoshare for an honest party i. This pseudoshare includes valid components corresponding to all subsets $A \cup \{i\}$, where $A \notin \Gamma$ and $|A| = c$, derived from the h-th copy of the sharings of the k-th bit stored in the tracing key. It is then padded with uniformly random values to match the size of a full share.

Trace. The tracing procedure is carried out by the tracer and proceeds iteratively over each $i \in I$ and each bit of id_j. Since the reconstruction box only accepts full shares, we first generate a complete pseudoshare for the party i containing valid pseudoshare components for unauthorized subsets of size c using the $\mathsf{ShareGeneration}$ algorithm, and then query the reconstruction box with it. As explained earlier, the reconstruction box cannot distinguish these projected pseudoshares from genuine shares. Consequently, when queried in this way, a *good* reconstruction box will run the $\mathsf{Reconstruction}$ algorithm on the pseudoshares and output the fake secret s_b, which reveals the value of the corresponding bit b of the identifier id_j. [9]

After completing the $2 \log m$ queries to recover all bits of id_j, the tracer reconstructs the full identifier and compares its first component id_j^1 with the corresponding entries in the tracing key to ensure validity and identify the party j. Once this check succeeds, the second component id_j^2 is used as the proof π and given to the verifier.

Since the reconstruction box is imperfect, the tracing procedure for each identifier bit is repeated N times, and the tracer outputs the majority response. Crucially, the h-th repetition queries the box with the pseudoshare built from the h-th copy of the fake sharings, so the repetitions are built from independent sharings. By the Chernoff bound, this majority-vote rule ensures that, when pseudoshares are generated for an honest party i, the reconstruction box returns the full identifier id_j correctly with overwhelming probability.

Verify. It follows the same procedure as that of Theorem 3.1: it checks whether the value provided in the proof π matches the second component of the identifier of the accused party, which is obtained from the verification. If so, it outputs 1; otherwise, it returns 0.

Since the tracing key reveals no information about the identifier of any honest party i, a malicious tracer can falsely accuse such a party only by correctly

[9] See Section 4.7 for an optimization that enables tracing multiple corrupted parties.

guessing its identifier. If identifiers are sampled uniformly from a sufficiently large domain, this probability is negligible.

4.5 The Scheme

We provide in Fig. 13 an outline of the scheme's construction, while its full description is given in Fig. 14 and Fig. 15. The formal statement is presented in Theorem 4.12, and its proof is deferred to Section 4.6.

Theorem 4.12. *Let $\mathcal{F}$ be a minimal covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, 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 of Fig. 13 is an information theoretic hidden mildly strong traceable secret sharing scheme with statistical reconstruction, traceability, and non-imputability realizing $\mathcal{F}$ with a share size of $O(2^n \text{poly}(n, \log m)/\epsilon^2)$ and $O(n\lambda \log m/\epsilon^2)$ queries.*

As a simple corollary of Theorem 4.12, we get our hidden mildly strong traceable scheme for general access structures.

Corollary 4.13 (Theorem 1.2 restated). *Let $\mathcal{F}$ be the family of all access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, let $m = 2^{\Theta(\lambda)}$, and let $\epsilon(\lambda) \in (0, \frac{1}{2}]$ be a non-negligible function. For any $s \in \mathbb{Z}_m^2$ there exists a traceable secret sharing scheme with statistical reconstruction, traceability, and non-imputability realizing $\mathcal{F}$ with a share size of $O(2^n \text{poly}(n, \log m)/\epsilon^2)$ and $O(n\lambda \log m/\epsilon^2)$ queries.*

Proof. It suffices to combine Theorem 4.12 with Lemma 4.10. □

4.6 Proof of Theorem 4.12

For clarity, we organize the proof of Theorem 4.12 as a sequence of propositions. We begin by showing that the protocol described in Fig. 14 and Fig. 15 is a secret sharing scheme.

Proposition 4.14. *Let $\mathcal{F}$ be a minimal covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, 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 of Fig. 14 and Fig. 15 is an information theoretic hiding secret sharing scheme with statistical reconstruction realizing $\mathcal{F}$ with a share size of $O(2^n \text{poly}(n, \log m)/\epsilon^2)$.*

Proof. We begin by proving the privacy of the scheme. The Share algorithm distributes the secret using the hiding general scheme of [FG26]. In addition, the extra components of each share are either uniformly random values or shares of fake secrets s_b used to recover the bits of the identifiers. These fake secrets are themselves shared using the same additive scheme and reveal no information about the actual secret. Consequently, the overall privacy of the scheme follows directly from the privacy guarantees of the construction in [FG26].

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

1. Share the indicator and the secret $0, s \in \mathbb{Z}_m^2$ according to Γ with the general hiding scheme of [FG26].
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 $A \notin \Gamma$ and each $i \notin A$
   (a) Select randomly a party $j \in A$.
   (b) For each bit $b \in \{0, 1\}$ of id_j, repeat N times
      i. Share the indicator and the fake secret $0, s_b \in \mathbb{Z}_m^2$ among $A \cup \{i\}$ with the additive scheme.
      ii. Append the share of the party i in the tracing key.
      iii. Replace the share of the party i by uniformly random values.
   (c) Ensure a uniform share size among all parties sending dummy random values to them.
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.

$\mathsf{Rec}(\{\mathsf{sh}_i\}_{i \in A})$:

1. For each set of share components $\{(\mathsf{sh}_{i,A}^0 \mathsf{sh}_{i,A}^s)\}_{i \in A}$ shared for the authorized set A
   (a) If $\sum_{i \in A} \mathsf{sh}_{i,A}^0 = 0$, set $s = \sum_{i \in A} \mathsf{sh}_{i,A}^s$ and **break**.
2. Output s.

$\mathsf{ShGen}(\mathsf{tk}, i, c, k, h)$:

1. Use as the pseudoshare the portions of the shares corresponding to the party i that are obtained in the h-th repetition of the secret sharing of the k-th bit of the identifier to the subsets of size $c + 1$ containing i.
2. Ensure that the pseudoshare matches the size of actual shares by padding dummy random values to it.

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

1. For $i \in I$
   (a) For each bit index $k \in [2 \log m]$
      i. For $h \in [N]$
         A. Compute the pseudoshare of the party i by calling $\mathsf{ShGen}(\mathsf{tk}, i, c, k, h)$.
         B. Query the reconstruction box R with this pseudoshare.
         C. Store the output bit b^*.
      ii. Take as the bit b the majority value among all the N outputs b^*.
   (b) If the first part of the reconstructed identifier matches the first part of the identifier of some party j in the tracing key, set the second part of the reconstructed identifier as the proof π and **break**.
2. If some party j has been identified, output $(\{j\}, \pi)$. Otherwise, output $(\emptyset, \perp)$.

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

1. If the second part of the identifier in the proof π matches that of the party in I contained in the verification key vk, output 1.
2. Otherwise, output 0.

Fig. 13: Outline of a hidden mildly strong traceable secret sharing scheme for any access structure Γ of a minimal covering family of access structures $\mathcal{F}$, where $N = \mathrm{poly}(n, 1/\epsilon)$.

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 $\Phi_t = \Delta_t \cup \{(A, i) : \emptyset \neq A \in \overline{\Delta}_{t-1}, i \in [n] \setminus A\}$.
7. Let $N = \lceil 2(\lambda + log(4n \log m))/\epsilon^2 \rceil$, and let $Q = [2 \log m] \times [N]$ be the set of labels of the $2N \log m$ layers of each share.
8. Let P_t denote the set of permutations of $|\Phi_t|$ elements and P'_m the set of bijections from Q to $[2N \log m]$.

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

1. Share the indicator and the secret $0, s \in \mathbb{Z}_m^2$ with the general hiding anonymous scheme Σ of [FG26] according to Γ, i.e. $\{\mathsf{sh}^s_{i,(2\log m,N)}\}_{i\in[n]} \leftarrow \Sigma(s, \Gamma)$.
2. Parse $\mathsf{sh}^s_{i,(2\log m,N)}$ as $\{\mathsf{sh}^s_{i,(2\log m,N),k}\}_{k\in[n]}$ for $i \in [n]$.
3. Parse $\mathsf{sh}^s_{i,(2\log m,N),k}$ as $\{\mathsf{sh}^*_{i,(2\log m,N),k,A}\}_{A\in\Delta_k}$ for $i \in [n]$, $k \in [n]$.
4. For $(q, h) \in Q \setminus \{(2\log m, N)\}$
   (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 n parties, i.e. $\mathsf{sh}^*_{i,(q,h),k,A} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $i \in [n]$.
5. Set $\mathsf{id}_i = (\mathsf{id}^1_i, \mathsf{id}^2_i) = (\mathsf{id}_{i,1}, ..., \mathsf{id}_{i,2\log m}) \leftarrow_\$ \mathbb{Z}_m^2$ for $i \in [n]$.
6. For $k \in [n]$
   (a) For $A \in \overline{\Delta}_k$
      i. If $A \in \overline{\Gamma}$
         A. For $i \in [n] \setminus A$
            I. Select a party $j \in A$ uniformly at random, i.e. $j \leftarrow_\$ A$.
            II. For $(q, h) \in Q$
               1. Set $b \leftarrow \mathsf{id}_{j,q}$.
               2. Share the indicator and the fake secret $0, s_b \in \mathbb{Z}_m^2$ with the additive scheme Σ' of [FG26] among $A \cup \{i\}$, i.e. $\{\mathsf{sh}^*_{\ell,(q,h),k+1,(A,i)}\}_{\ell\in A\cup\{i\}} \leftarrow \Sigma'(s_b, A \cup \{i\})$.
               3. Set as the traceable key for the h-th copy of the q-th bit of the identifier for the pair (A, i) the shares of the party i, i.e. $\mathsf{tk}^*_{(q,h),(A,i)} \leftarrow \mathsf{sh}^*_{i,(q,h),k+1,(A,i)}$.
               4. Send two random elements $r_0, r_s \in \mathbb{Z}_m^2$ to each party outside A, i.e. $\mathsf{sh}^*_{\ell,(q,h),k+1,(A,i)} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $\ell \in [n] \setminus A$.
      ii. If $A \notin \overline{\Gamma}$, send $4N \log m(n - |A|)$ random elements of $\mathbb{Z}_m^2$ to each of the n parties, i.e. $\mathsf{sh}^*_{\ell,(q,h),k+1,(A,i)} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$ for $\ell \in [n]$, $i \in [n] \setminus A$, and $(q, h) \in Q$.
7. For $t \in [n]$ and $(q, h) \in Q$
   (a) Select a permutation $\sigma_{t,(q,h)} \in P_t$ uniformly at random, i.e. $\sigma_{t,(q,h)} \leftarrow_\$ P_t$.
   (b) For $u \in \Phi_t$
      i. Set $B \leftarrow \sigma_{t,(q,h)}(u)$ and $\mathsf{sh}_{i,(q,h),t,B} \leftarrow \mathsf{sh}^*_{i,(q,h),t,u}$ for $i \in [n]$.
      ii. If $u = (A, i)$ for some $A \in \overline{\Gamma}$, set $\mathsf{tk}_{(q,h),B} \leftarrow \mathsf{tk}^*_{(q,h),u}$.
   (c) Set $\mathsf{sh}_{i,(q,h),t} \leftarrow \{\mathsf{sh}_{i,(q,h),t,B}\}_{B\in\Phi_t}$ for $i \in [n]$.
8. Select a bijection $\sigma'_m \in P'_m$ uniformly at random, i.e. $\sigma'_m \leftarrow_\$ P'_m$.
9. For $(q, h) \in Q$
   (a) Set $p \leftarrow \sigma'_m((q, h))$ and $\mathsf{sh}_{i,p} \leftarrow \{\mathsf{sh}_{i,(q,h),t}\}_{t\in[n]}$ for $i \in [n]$.
10. Set $\mathsf{sh}_i \leftarrow \{\mathsf{sh}_{i,p}\}_{p\in[2N\log m]}$ for $i \in [n]$.
11. Set $\mathsf{tk} \leftarrow (\sigma'_m, \{\sigma_{t,(q,h)}\}_{t\in[n],(q,h)\in Q}, \{\mathsf{tk}_{(q,h),B}\}_{(q,h),B}, \{\mathsf{id}^1_i\}_{i\in[n]})$.
12. Set $\mathsf{vk} \leftarrow \{\mathsf{id}^2_i\}_{i\in[n]}$.
13. Output $(\{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk})$.

Fig. 14: Share algorithm of a hidden mildly strong traceable secret sharing scheme for any access structure Γ of a minimal covering family of access structures $\mathcal{F}$.

$\mathsf{Rec}(\{\mathsf{sh}_i\}_{i\in A})$:

1. The same as in the $\emptyset$-strong traceable scheme of Fig. 9.

$\mathsf{ShGen}(\mathsf{tk}, i, c, k, h)$:

1. Parse tk as $(\sigma'_m, \{\sigma_{t,(q,h')}\}_{t\in[n],(q,h')\in Q}, \{\mathsf{tk}_{(q,h'),B}\}_{(q,h'),B}, \{\mathsf{id}^1_\ell\}_{\ell\in[n]})$.
2. For $(q,h') \in Q$
   (a) Set $p \leftarrow \sigma'_m((q,h'))$.
   (b) For $t \in [n]$
      i. For $B \in \Phi_t$
         A. Set $u \leftarrow \sigma^{-1}_{t,(q,h')}(B)$.
         B. If $(q,h') = (k,h)$, $t = c+1$, and $u = (A,i)$ for some $A \in \overline{\Delta}_c \cap \overline{\Gamma}$
            I. Set $\mathsf{sh}'_{i,p,t,B} \leftarrow \mathsf{tk}_{(k,h),B}$.
         C. Otherwise
            I. Send two random elements $r_0, r_B \in \mathbb{Z}_m^2$ to the party i, i.e. $\mathsf{sh}'_{i,p,t,B} \leftarrow_\$ (\mathbb{Z}_m^2, \mathbb{Z}_m^2)$.
      ii. Set $\mathsf{sh}'_{i,p,t} \leftarrow \{\mathsf{sh}'_{i,p,t,B}\}_{B\in\Phi_t}$.
   (c) Set $\mathsf{sh}'_{i,p} \leftarrow \{\mathsf{sh}'_{i,p,t}\}_{t\in[n]}$.
3. Set $\mathsf{sh}'_i \leftarrow \{\mathsf{sh}'_{i,p}\}_{p\in[2N\log m]}$.
4. Output sh'_i.

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

1. Parse tk as $(\sigma'_m, \{\sigma_{t,(q,h)}\}_{t,(q,h)}, \{\mathsf{tk}_{(q,h),B}\}_{(q,h),B}, \{\mathsf{id}^1_\ell\}_{\ell\in[n]})$.
2. Set three empty lists L, L', L''.
3. For $i \in I$
   (a) For $q \in [2\log m]$
      i. For $h \in [N]$
         A. Set $\mathsf{sh}'_{i,h} \leftarrow \mathsf{ShGen}(\mathsf{tk}, i, c, q, h)$.
         B. Query R on $\mathsf{sh}'_{i,h}$. Let b_h be the response.
         C. Store b_h in L''.
      ii. Set $\mathsf{id}_{i,q}$ the majority value in L'' and erase L''.
   (b) Set $\mathsf{id}'^1_i \leftarrow \{\mathsf{id}_{i,q}\}_{q\in[\log m]}$ and set $\mathsf{id}'^2_i \leftarrow \{\mathsf{id}_{i,q}\}_{q\in[2\log m]\setminus[\log m]}$.
   (c) If $\mathsf{id}'^1_i = \mathsf{id}^1_j$ for some $j \in [n]$
      i. Add j to L.
      ii. Add id'^2_i to L'.
      iii. **Break**.
4. If $|L| \neq 1$, terminate and output $(\emptyset, \perp)$.
5. Set $J^* \leftarrow L$.
6. Set $\pi \leftarrow L'$.
7. Output (J^*, π).

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

1. The same as in the $\emptyset$-strong traceable scheme of Fig. 10.

Fig. 15: Reconstruction, ShareGeneration, Trace, and Verify algorithms of a hidden mildly strong traceable secret sharing scheme for any access structure Γ of a minimal covering family of access structures $\mathcal{F}$.

Similarly, the statistical correctness follows from the hiding general scheme of [FG26] and the fact that, for any authorized set A, all but one of the candidate share components for the subset A are uniformly random due to the analysis of the Share algorithm. As a result, statistical correctness is ensured by choosing a sufficiently large group size m.

Regarding the perfect hideout requirement, it suffices to argue that the secret is shared using the hiding general scheme of [FG26], that the components that any subset of parties holds in the additional sharings are uniformly random, since the original component of the party i in the sharing for each pair (A, i) is only included in the tracing key, 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 of the $2N \log m$ layers, each party receives at most $n + 1$ pairs of share components of $2 \log m$ bits for each subset of $[n]$. Since $\lambda = \Theta(\log m)$ and $\lambda = \Omega(\text{poly}(n))$, we have that $N = O(\log m/\epsilon^2)$, so the total share size per party is $O(2^n \text{poly}(n, \log m)/\epsilon^2)$. □

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

Proposition 4.15. *Let $\mathcal{F}$ be a minimal covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, 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 secret sharing scheme of Fig. 14 and Fig. 15 has tracer privacy.*

Proof. The tracing key consists of the permutations σ'_m and $\sigma_{t,(q,h)}$, the first parts of the identifiers, and the original components of the parties i in the additional sharings for the pairs (A, i), which are all generated independently of the secret. In particular, the tracing key does not contain any component of the sharing of the secret with Σ. Therefore, as in the hidden traceable scheme of [FG26], for any unauthorized subset B, the components of the parties in B in the sharing of the secret are independent of the tracing key and of the remaining components of their shares, and their distribution does not depend on the secret, by the privacy of Σ. Consequently, the joint distribution of the shares of B and the tracing key does not depend on the secret, and the perfect tracer privacy requirement holds. □

Now, we move to prove that the scheme achieves statistical hidden mildly strong traceability.

Proposition 4.16. *Let $\mathcal{F}$ be a minimal covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, 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 secret sharing scheme of Fig. 14 and Fig. 15 has statistical hidden mildly strong traceability and requires $O(n\lambda \log m/\epsilon^2)$ queries.*

Proof. The upper bound on the number of queries follows from the algorithm description, since Trace makes at most $2N|I| \log m \leq 2nN \log m$ queries and $N = O(\lambda/\epsilon^2)$.

Consider an execution of the experiment of Fig. 12, and let I be the label of the reconstruction box R. As in Section 4.4, we assume that the tracer knows the number $c = |J|$ of corrupted parties. The adversary can win in one of two ways: given a $\mathsf{GoodBox}$, the tracer either outputs a valid accusation against a party outside J, or an invalid accusation against a party in J. We bound the probability of each event separately.

We start by bounding the probability that the tracer outputs a valid accusation against a party outside J. Let $j \in [n] \setminus J$. By construction, the second part id_j^2 of the identifier of j only affects the sharings for the pairs (A, x) for which j has been selected, and $j \in A$ in all of them. In each of these sharings, the component of j is included neither in the tracing key nor in the shares of J, so the values of the sharing included in them are uniformly random and independent of id_j^2. Since R only depends on the shares of J and the random coins of the adversary, and the queries of the tracer only depend on the tracing key and on fresh random values, the output of the tracer is independent of id_j^2. Hence, the tracer outputs a valid accusation against a party outside J with probability at most $1/m$.

Next, we bound the probability that $\mathsf{GoodBox}$ holds and the tracer does not output a valid accusation against a party in J. To this end, we fix the shares $\mathbf{sh} = \{\mathsf{sh}_i\}_{i \in J}$ of the corrupted parties and the random coins of the adversary, so that R and its label I are also fixed. We denote by i^* the first party of $I \setminus J$ considered by Trace, and by $j^* \in J$ the party selected for the pair (J, i^*) in Step 6 of Share. Notice that this pair is well defined, since $J \in \overline{\Delta}_c \cap \overline{\Gamma}$ and $i^* \notin J$. Moreover, since $\mathcal{F}$ is a minimal covering family and $J \notin \Gamma$, there exists an access structure $\Gamma^* \in \mathcal{F}$ in which $J \cup \{i^*\}$ is minimal.

First, notice that the corrupted parties cannot learn anything about the permutations nor about the additional sharings that they do not fully hold. More precisely, consider any execution of Share for an access structure $\Gamma' \in \mathcal{F}$ with $J \notin \Gamma'$. Since every authorized subset of Γ' contains a party outside J, by the construction of Σ, the components of J in the sharing of the secret are uniformly random and independent, and the same holds for the components of the rest of the layers that are not part of an additional sharing. Moreover, the sharing for each pair (A, x) is an independent additive sharing among $A \cup \{x\}$ in which the component of x is replaced with uniformly random values, and the parties outside A receive uniformly random values, so the components of J in it are uniformly random and independent, regardless of the bit of the identifier that it encodes. Since the permutations only determine the positions of the components, the shares of J are uniformly random and independent of the secret, the identifiers, and the permutations σ'_m and $\sigma_{t,(q,h)}$. Furthermore, the original component of x in the sharing for a pair (A, x) is equal to the shared value minus the components of the parties in A. Hence, if $A \not\subseteq J$, the component of some party in $A \setminus J$ is uniformly random and independent of the shares of J and of all the other sharings, so, conditioned on the shares of J, the value of the tracing key corresponding to (A, x) is uniformly random and independent of the identifiers and of the rest of the tracing key.

Second, we characterize the distribution of the pseudoshares generated by the tracer. For a layer $p \in [2N \log m]$ and a bit $\beta \in \{0,1\}$, let $\mathcal{Y}_{p,\beta}$ be the distribution of a share of the party i^* whose components are sampled independently as follows:

- At the layer p and the level $c+1$, the component at a uniformly random position $B \in \Phi_{c+1}$ is set so that, together with the components of the parties in J at the same position, it adds up to $(0, s_\beta)$.
- Any other component is uniformly random.

We claim that, for every $q \in [2 \log m]$ and every $h \in [N]$, the pseudoshare generated by $\mathsf{ShGen}(\mathsf{tk}, i^*, c, q, h)$ is distributed according to $\mathcal{Y}_{p,\beta}$ conditioned on $\mathbf{sh}$, σ'_m, and the identifiers, where $p = \sigma'_m((q,h))$ and $\beta = \mathsf{id}_{j^*,q}$. Indeed, by construction, this pseudoshare contains uniformly random values at every position, except at the layer p and the level $c+1$, where it contains the value $\mathsf{tk}_{(q,h),B}$ of the tracing key at each position $B = \sigma_{c+1,(q,h)}((A, i^*))$ with $A \in \overline{\Delta}_c \cap \overline{\Gamma}$. In particular, at the position $B^* = \sigma_{c+1,(q,h)}((J, i^*))$, it contains the original component of i^* in the sharing for the pair (J, i^*). Since this is a sharing of $(0, s_\beta)$ among $J \cup \{i^*\}$, this component adds up, together with the components of the parties in J at the same position, to $(0, s_\beta)$. Moreover, $\sigma_{c+1,(q,h)}$ is uniformly random and independent of $\mathbf{sh}$, so B^* is a uniformly random position of Φ_{c+1}. For every other $A \in \overline{\Delta}_c \cap \overline{\Gamma}$ with $i^* \notin A$, we have $A \not\subseteq J$, since $|A| = |J|$ and $A \neq J$, so the corresponding value is uniformly random and independent of all the other values, as shown above. In contrast, for every corrupted party $i \in I \cap J$, every pair (A, i) with $A \in \overline{\Delta}_c$ satisfies $A \neq J$, since $i \notin A$, and thus $A \not\subseteq J$. Therefore, the output of $\mathsf{ShGen}(\mathsf{tk}, i, c, q, h)$ consists of uniformly random values, which are independent of the identifiers. Finally, since different copies correspond to independent sharings and independent permutations $\sigma_{c+1,(q,h)}$, pseudoshares generated from different copies are independent and the uniformly random values used by $\mathsf{ShareGeneration}$ are sampled independently.

Third, we relate the behavior of R on these pseudoshares with its behavior on genuine shares. We denote by $\mu_\beta(p)$ the probability that $R(Y) = \beta$, where $Y \leftarrow \mathcal{Y}_{p,\beta}$, and the probability is also taken over the random coins of R, and we let $\overline{\mu}_\beta = \frac{1}{2N \log m} \sum_{p \in [2N \log m]} \mu_\beta(p)$. Moreover, for $\beta \in \{0,1\}$, let a_β be the probability that $R(\{\mathsf{sh}'_{i^*}\}) = \beta$, where the probability is taken over $(\{\mathsf{sh}'_i\}_{i \in [n]}, \mathsf{tk}', \mathsf{vk}') \leftarrow \mathsf{Share}(s_\beta, \Gamma^*)$ conditioned on $\{\mathsf{sh}'_i\}_{i \in J} = \mathbf{sh}$, and the random coins of R. We claim that $a_\beta = \overline{\mu}_\beta$. To see this, consider such a sharing, and let p_0 be the layer that contains the sharing of the secret with Σ. Since $J \notin \Gamma^*$, the permutations of this sharing are uniformly random and independent of $\mathbf{sh}$, as shown above. Hence, p_0 is a uniformly random layer, and the position B_0 of the subset $J \cup \{i^*\}$ in the level $c+1$ of the layer p_0 is a uniformly random position of Φ_{c+1}. Moreover, conditioned on $p_0 = p$, the share of i^* is distributed according to $\mathcal{Y}_{p,\beta}$. Indeed, at the position B_0, the components of the parties in $J \cup \{i^*\}$ add up to $(0, s_\beta)$, by the construction of Σ. In addition, since $J \cup \{i^*\}$ is minimal, every other authorized subset $A \in \Gamma^*$ with $i^* \in A$ contains a party outside $J \cup \{i^*\}$, so the component of i^* at the position of A is uniformly random

and independent of **sh**. Similarly, in the sharing for every pair (A, x) with $i^* \in A$, the original component of x is not included in the shares, so the component of i^* is uniformly random and independent of **sh**. Since all the remaining components of i^* are uniformly random, the claim follows by averaging over $p \in [2N \log m]$.

We are now ready to bound the probability that $\mathsf{GoodBox}$ holds and the tracer does not output a valid accusation against a party in J. Since Γ^* and $I' = \{i^*\}$ satisfy the conditions of Definition 4.6, $\mathsf{GoodBox}$ can only hold if $a_b \geq \frac{1}{2} + \epsilon$. We distinguish two cases.

First, assume that $a_\beta < \frac{1}{2} + \epsilon$ for some $\beta \in \{0, 1\}$. Then, $\mathsf{GoodBox}$ can only hold if $b = 1 - \beta$. By the privacy of the scheme, b is uniformly random and independent of **sh**, so $\mathsf{GoodBox}$ holds with probability at most $\frac{1}{2}$.

Now, assume that $a_0, a_1 \geq \frac{1}{2} + \epsilon$, so that $\overline{\mu}_0, \overline{\mu}_1 \geq \frac{1}{2} + \epsilon$. We first consider the parties of $I \cap J$ that Trace considers before i^*. As shown above, the queries generated for them are independent of the identifiers, and so are the outputs of R on them. Hence, for each of these parties, the first part of the reconstructed identifier coincides with the first part of the identifier of some party with probability at most n/m, so the tracer reaches i^* except with probability at most n^2/m. Next, consider the q-th bit computed for i^*, and let $\beta = \mathsf{id}_{j^*,q}$. In its computation, the h-th query is generated from the copy with label (q, h). As shown above, conditioned on **sh**, σ'_m, and the identifiers, these queries are independent, and the h-th one is answered with β with probability $\mu_\beta(\sigma'_m((q, h)))$. Since R is stateless, its outputs on these N queries are independent. Hence, by the Chernoff bound, the fraction of outputs equal to β is greater than $\frac{1}{N} \sum_{h \in [N]} \mu_\beta(\sigma'_m((q, h))) - \frac{\epsilon}{2}$ except with probability at most $e^{-N\epsilon^2/2}$. Moreover, σ'_m is uniformly random and independent of **sh** and of the identifiers, so $\{\sigma'_m((q, h))\}_{h \in [N]}$ is a uniformly random subset of N layers. Hence, by the Chernoff bound for sampling without replacement,

$$\frac{1}{N} \sum_{h \in [N]} \mu_\beta(\sigma'_m((q, h))) > \overline{\mu}_\beta - \frac{\epsilon}{2} \geq \frac{1}{2} + \frac{\epsilon}{2}$$

except with probability at most $e^{-N\epsilon^2/2}$. Therefore, the majority value computed for the q-th bit is equal to β except with probability at most $2e^{-N\epsilon^2/2}$, and, by the union bound and the choice of N, the tracer reconstructs all the bits of id_{j^*} correctly except with probability at most $4 \log m \cdot e^{-N\epsilon^2/2} \leq e^{-\lambda}$. In this case, the first part of the reconstructed identifier coincides with $\mathsf{id}^1_{j^*}$, and it does not coincide with the first part of the identifier of any other party except with probability at most $(n-1)/m$. Then, the tracer outputs $(\{j^*\}, \mathsf{id}^2_{j^*})$, which is a valid accusation against $j^* \in J$. Consequently, in this case, the probability that $\mathsf{GoodBox}$ holds and the tracer does not output a valid accusation against a party in J is at most $e^{-\lambda} + (n^2 + n)/m$.

Therefore, since $e^{-\lambda} + (n^2 + n)/m \leq \frac{1}{2}$ fir sufficiently large λ, combining the two cases shows that the probability that the tracer fails to output a valid accusation against a party in J is at most $\frac{1}{2}$.

Finally, combining the bounds on the two ways in which the adversary can win, we conclude that the experiment in Fig. 12 outputs 1 with probability at

most $\frac{1}{2} + \frac{1}{m}$. This proves that the scheme has statistical hidden mildly strong traceability. □

We conclude the proof of Theorem 4.12 by proving that the scheme satisfies statistical hidden non-imputability.

Proposition 4.17. *Let $\mathcal{F}$ be a minimal covering family of access structures over n parties, let $\lambda \in \mathbb{N}$ be the security parameter with $\lambda = \Omega(\text{poly}(n))$, 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 secret sharing scheme of Fig. 14 and Fig. 15 has statistical hidden non-imputability.*

Proof. Let $\mathcal{A}$ be an adversary in the experiment of Fig. 6, which receives the shares of all the parties except i^*, and the tracing key. As in the hidden traceable scheme of [FG26], the second part $\mathsf{id}_{i^*}^2$ of the identifier of i^* only affects the sharings for the pairs (A, x) for which i^* has been selected, and $i^* \in A$ in all of them. In each of these sharings, the component of i^* is not given to $\mathcal{A}$, so all the other components of the sharing, including the original component of x included in the tracing key, are uniformly random and independent of $\mathsf{id}_{i^*}^2$. Since all the other values received by $\mathcal{A}$ are independent of $\mathsf{id}_{i^*}^2$, the view of $\mathcal{A}$ is independent of $\mathsf{id}_{i^*}^2$, which is uniformly random in $\mathbb{Z}_m$. Hence, the probability that $\mathcal{A}$ outputs a valid proof against i^* is at most $1/m$, which is negligible in λ. □

4.7 Further Optimizations

Next, we present additional optimizations for our hidden mildly strong traceable scheme. First, we extend the construction to the setting where the tracer does not know in advance the number of corrupted parties that contributed to the reconstruction box. Then, we describe a modification that enables tracing multiple corrupted parties simultaneously.

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

Remark 4.18. To handle the case where the number of corrupted parties c is unknown, we adopt the approach of [BPR24]. Specifically, we run the Trace algorithm iteratively with $c = n-1, n-2, \ldots$ until the tracing succeeds.

We now justify the soundness of the approach described in Remark 4.18. Let c^* denote the actual number of corrupted parties. By construction, for any $c > c^*$, the queries generated by the Trace algorithm consist of pseudoshares that can only reconstruct an identifier when combined with shares from a set of c parties. By the privacy of the scheme, when $c > c^*$, these queries reveal no information about any identifier. Consequently, the reconstruction box outputs

a valid identifier only with negligible probability, and the tracing procedure succeeds only negligibly.

In contrast, when $c = c^*$, the tracing procedure is identical to the one analyzed in Theorem 4.12, which guarantees that the tracer fails to identify a corrupted party with negligible probability. Therefore, the procedure can be adapted to settings where the number of corrupted parties is unknown by simply testing all possible values of c in descending order.

Tracing Multiple Corrupted Parties Simultaneously. In the traitor tracing literature [CFNP00,GSS21,BPR24], identifying a single corrupted party is usually considered sufficient, as this serves as a deterrent for others. Nonetheless, our scheme can be adapted to trace multiple corrupted parties, even all of them, at the cost of increasing the share size 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 encoding a single random identifier per unauthorized subset, we independently encode the identifiers for all parties in the subset. Concretely, for every unauthorized subset of size c, we run c independent instances of the base scheme in parallel, one for each party, where the j-th instance embeds the identifier of the j-th party. This effectively 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, once for each party.

5 Impossibility of Stateful Tracing

In this section, we show that even the weakest notion of traceability for general access structures is impossible to achieve together with non-imputability when reconstruction boxes are *stateful*, i.e., they maintain an internal state and adapt their behavior based on past queries. We first introduce the notions of a *consistent sequence of input queries* and a *stateful-good reconstruction box*. We then reduce the problem to the simpler case of a box operating on a single input query, and use this reduction to prove the impossibility result.

5.1 Definitions

Before defining a stateful-good reconstruction box, we introduce the notion of a consistent sequence of shares. Intuitively, for an access structure Γ and a secret sharing scheme Σ, a sequence of shares is (Γ, Σ)-consistent if there exists a secret s such that all shares could have been generated in a single execution of Share for s and Γ.

Definition 5.1. *Let $\mathcal{S}$ be a finite set of secrets, let Γ be an access structure over n parties, and let Σ be a secret sharing scheme for Γ with sharing algorithm* Share*. Let $k \in \mathbb{N}$, and let $\{I_j\}_{j\in[k]}$ be a sequence of subsets $I_j \subseteq [n]$. A sequence*

of shares $\{\mathbf{sh}_{I_j}\}_{j\in[k]}$ *is* (Γ,Σ)*-consistent if there exist a secret* $s \in \mathcal{S}$ *and shares* $\{\mathsf{sh}_i\}_{i\in[n]}$ *such that for every* $j \in [k]$*, it holds that*

$$\mathbf{sh}_{I_j} = \{\mathsf{sh}_i : i \in I_j\},$$

where $\{\mathsf{sh}_i\}_{i\in[n]} \leftarrow \mathsf{Share}(s,\Gamma)$.

Once the notion of a consistent sequence of shares is defined, we introduce stateful-good reconstruction boxes. To prove our impossibility result, we strengthen the definition to make the reconstruction box as easy to trace as possible.

Concretely, given a corrupted set J constructing a reconstruction box with label I, we require that $I \cap J = \emptyset$. The box must succeed with non-negligible probability on queries corresponding to any subset $I' \subseteq I$, outputting the full secret rather than a single bit. To model statefulness, we additionally require that each query forms a consistent sequence of shares with both the previous queries and the shares of the corrupted parties in J.

Definition 5.2 (Stateful-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 traceable secret sharing for* Γ*. For* $\epsilon \in [0,1]$*, secrets* $s \in \mathcal{S}$*, an unauthorized subset* $J \notin \Gamma$*, a corresponding set of shares* $\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$*, and a label* $I \subseteq [n]$ *with* $I \cap J = \emptyset$*, a reconstruction box* R *is* $(\Gamma, I, \mathbf{sh}, s, \epsilon)$*-stateful-good if for any* $I' \subseteq I$ *with* $I' \cup J \in \Gamma$*, it holds that*

$$\mathbf{P}[R(\{\mathsf{sh}'_i\}_{i\in I'}) = s] \geq \epsilon,$$

where the probability is taken over $(\{\mathsf{sh}'_i\}_{i\in[n]}, \mathsf{tk}, \mathsf{vk}) \leftarrow \mathsf{Share}(s,\Gamma)$ *conditioned on* $\{\mathsf{sh}'_i\}_{i\in I'}$*,* $\mathbf{sh}$*, and the previous queries to* R *forming a* (Γ,Σ)*-consistent sequence of shares, and the random coins of* R*.*

Given the notion of a stateful-good reconstruction box, we define *stateful traceability* accordingly. It mirrors $\emptyset$-strong traceability, except that the secret is sampled uniformly from its domain and the reconstruction box must be stateful-good. As usual, modifications to the experiment are highlighted in blue.

Definition 5.3 (Perfect Stateful Traceability). *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 traceable scheme for* Γ*. For any* $\epsilon \in [0,1]$*,* Σ *is perfect stateful traceable if for every adversary* $\mathcal{A}$*, it holds that*

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

The notion of *stateful non-imputability* is identical to that of $\emptyset$-strong and strong non-imputability, as defined in Fig. 4 and Definition 2.13.

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

1. $(\Gamma, J, \mathsf{state}) \leftarrow \mathcal{A}(\cdot)$.
2. $s \leftarrow_{\$} \mathcal{S}$.
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. $(R, I) \leftarrow \mathcal{A}(\mathsf{state}, \mathbf{sh})$.
6. $(J^*, \pi) \leftarrow \mathsf{Trace}^R(\mathsf{tk})$.
7. $\mathsf{GoodBox} := (I \cap J = \emptyset) \wedge (J \notin \Gamma) \wedge (I \notin \Gamma) \wedge (R \text{ is } (\Gamma, I, \mathbf{sh}, s, \epsilon)\text{-stateful-good})$.
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. 16: The tracing experiment for a stateful traceable secret sharing scheme Σ for an access structure Γ. The steps highlighted in blue indicate the differences compared to the tracing experiment of Fig. 2.

5.2 Reduction to a Single Query Reconstruction Box

Next, we show that any stateful-good reconstruction box can adopt a strategy that makes its behavior depend solely on the first valid input query, that is, a query corresponding to a subset $I' \subseteq I$ such that $I' \cup J \in \Gamma$. At a high level, the strategy works as follows.

Until receiving its first valid query, the stateful-good reconstruction box ignores all inputs, since its behavior on invalid input queries is unconstrained. Upon receiving the first valid query, it checks whether the query is consistent with the corrupted parties' shares. If so, it runs the reconstruction algorithm and outputs the recovered value; otherwise, it outputs $\perp$. From then on, the box ignores all future queries and always returns the same output produced for that first valid query, namely either the reconstructed value or $\perp$.

Note that this is a valid strategy for any stateful-good reconstruction box, since such a box is only required to output the result of the reconstruction procedure on an input query when that query forms a consistent sequence of shares with both the shares of the corrupted parties and all previous queries.

Intuitively, this strategy exploits the fact that the corrupted parties' shares are generated from a single, fixed secret. As a result, any query consisting of genuine shares produced by a correct sharing of that secret must reconstruct exactly the same value (the real secret). This intuition is formalized in the following lemma.

Lemma 5.4. *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 traceable secret sharing for Γ. For $\epsilon \in [0, 1]$, a secret $s \in \mathcal{S}$, an unauthorized subset $J \notin \Gamma$, a corresponding set of shares*

$\mathbf{sh} = \{\mathsf{sh}_i\}_{i\in J}$, *and a label* $I \subseteq [n]$ *with* $I \cap J = \emptyset$, *there exists a* $(\Gamma, I, \mathbf{sh}, s, \epsilon)$-*stateful-good reconstruction box* R *whose output only depends on the first valid input query.*

Proof. Let R be the reconstruction box that follows the strategy described in Section 5.2. It suffices to verify that R is a $(\Gamma, I, \mathbf{sh}, s, \epsilon)$-stateful-good reconstruction box.

To this end, it suffices to consider the case where, from the first valid input query onward, all subsequent queries, together with the shares of the corrupted parties in J, form a consistent sequence of shares, since this is the only situation in which the definition constrains the behavior of a stateful-good reconstruction box. In this case, by the correctness property of secret sharing schemes and the definition of a consistent sequence of shares, the behavior of R clearly matches that required of a stateful-good reconstruction box, as it always outputs the correct secret from which those shares were generated. □

This reduction allows us, without loss of generality, to focus the impossibility result on tracing stateful-good reconstruction boxes whose behavior depends only on the first valid input query.

5.3 An Impossibility Result

Finally, we prove the impossibility result. Specifically, we show that for a particular family of access structures, no traceable secret sharing scheme can simultaneously achieve both stateful traceability and stateful non-imputability.

To demonstrate this, we consider a family of access structures over $n \geq 7$ parties defined by monotone Boolean functions of the form $f_\Gamma = f_{\Gamma'} \wedge (\bigvee_{i\in[6]} x_i)$, where Γ' is a non-empty access structure on the remaining parties $[7, n]$. In these access structures, the first six parties are equivalent, i.e. they play the same role within Γ. We will show that whenever any pair of these six parties forms a corrupted subset, they can evade tracing.

The impossibility statement follows.

Theorem 5.5 (Theorem 1.3 restated). *Let* $n \geq 7$ *and let* Γ *be an access structure over* n *parties whose monotone Boolean function representation is* $f_\Gamma = f_{\Gamma'} \wedge (\bigvee_{i\in[6]} x_i)$, *where* Γ' *is a non-empty access structure on the rest of participants* $[7, n]$. *Then, there exists no traceable secret sharing scheme for* Γ *that simultaneously satisfies both stateful traceability and stateful non-imputability.*

Proof. We proceed by contradiction. Assume there exists a traceable secret sharing scheme Σ for an access structure Γ of the form $f_\Gamma = f_{\Gamma'} \wedge (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6)$ that simultaneously satisfies both stateful traceability and stateful non-imputability.

First, observe that by Lemma 5.4, we can assume without loss of generality that the tracing procedure involves only a single valid input query because any corrupted subset of parties can construct a reconstruction box whose behavior depends solely on its first valid input query. If this were not the case, the tracer

would be unable to perform the tracing procedure on such a box, leading to a contradiction.

Next, observe that for any two distinct parties i and j, and for any input query q performed by the tracer that is valid for both parties, it must hold that $R_i(q) \neq R_j(q)$, where R_i and R_j are reconstruction boxes built by i and j, respectively. Indeed, by stateful traceability, the tracer run on R_i must output a valid accusation against i. If $R_i(q) = R_j(q)$, the view of the tracer when run on R_j is identical to its view when run on R_i, since both boxes ignore all the queries before the first valid one and keep repeating the same output after it. Hence, an adversary that receives the shares of all the parties except i and the tracing key could build R_j and run the tracer on it, obtaining a valid accusation against i. This breaks non-imputability, which would again lead to a contradiction.

Once this observations are in place, we consider three particular stateful-good reconstruction boxes, $R_{1,2}$, $R_{3,4}$, and $R_{5,6}$ and show that they can adopt a strategy to avoid being traced. The strategy works as follows: each box internally simulates the two reconstruction boxes corresponding to the individual parties whose shares it contains. That is, $R_{1,2}$ simulates R_1 and R_2, while $R_{3,4}$ does the same with R_3 and R_4, and analogously for $R_{5,6}$. When the first valid input query q is received, each box passes q to its two simulated single-party boxes and compares their outputs. If the outputs coincide, the box returns the corresponding secret; otherwise, it outputs $\perp$. It is straightforward to verify that this strategy satisfies the definition of a stateful-good reconstruction box.

Now, notice that by the definition of Γ, any input query q is valid for $R_{1,2}$ if and only if it is valid for $R_{3,4}$ and if and only if it is valid for $R_{5,6}$. Moreover, recall the previous observation stating that for any two distinct parties i, j and any input query q valid for both parties, it must hold that $R_i(q) \neq R_j(q)$. Combining these facts, we conclude that the three reconstruction boxes $R_{1,2}$, $R_{3,4}$ and $R_{5,6}$ will always output $\perp$. Consequently, the tracer cannot distinguish whether the corrupted subset is $\{1, 2\}$, $\{3, 4\}$, or $\{5, 6\}$ and thus break traceability, which leads to a contradiction.[10]

Acknowledgments

This work was partly funded by MCIN/AEI/ 10.13039/501100011033 (projects MATSE PID2024-156636NB-C22 and ACITHEC PID2021-124928NB-I00).

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.

[10] Note that traceability is broken because the tracer only has a $\frac{1}{3}$ probability of accusing a corrupted party.

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.

BDF+25. Jan Bormet, Stefan Dziembowski, Sebastian Faust, Tomasz Lizurej, and Marcin Mielniczuk. Strong secret sharing with snitching. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025*, pages 295–327, Cham, 2025. Springer Nature Switzerland.

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

BF20. Amos Beimel and Oriol Farràs. The share size of secret-sharing schemes for almost all access structures and graphs. In *TCC 2020*, volume 12552 of *LNCS*, pages 499–529, 2020.

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

BGW99. László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

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 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.

BPR24. Dan Boneh, Aditi Partap, and Lior Rotem. Traceable secret sharing: Strong security and efficient constructions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – 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.

CDK+26. Ran Canetti, Ivan Damgård, Sebastian Kolby, Divya Ravi, and Sophia Yakoubov. Deniable secret sharing. In Benny Applebaum and Huijia (Rachel) Lin, editors, *Theory of Cryptography*, pages 399–427, Cham, 2026. Springer Nature Switzerland.

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

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.

DHKP26. Pousali Dey, Rittwik Hajra, Subha Kar, and Soumit Pal. Towards public tracing: Collaborative traceable secret sharing. *Cryptology ePrint Archive*, 2026.

FG26. Oriol Farràs and Miquel Guiot. Traceable secret sharing schemes for general access structures. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 3–32, Cham, 2026. Springer Nature Switzerland.

GJP26. Vipul Goyal, Abhishek Jain, and Aditi Partap. Traceable secret sharing revisited. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 33–63, Cham, 2026. Springer Nature Switzerland.

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

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

HKMP25. Rittwik Hajra, Subha Kar, Pratyay Mukherjee, and Soumit Pal. Traceable bottom-up secret sharing and law and order on community social key recovery. In *International Conference on Cryptology in India*, pages 417–441. Springer, 2025.

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.

LD25. M. Leslie and R. Dutta. A traceable threshold asmuth–bloom secret sharing scheme. Cryptology ePrint Archive, Paper 2025/1561, 2025.

LPLK25. Duc Anh Luong, Jong Hwan Park, Changmin Lee, and Hyoseung Kim. Tss-pv: Traceable secret sharing with public verifiability. *Cryptology ePrint Archive*, 2025.

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

Nir26. Oded Nir. Resolving the complexity of linear secret sharing. Cryptology ePrint Archive, Paper 2026/1667, 2026.

PR18. Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In *50th STOC*, pages 1207–1219, 2018.

RBG01. Lajos Rónyai, László Babai, and Murali K. Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the AMS*, 14(3):717–735, 2001.

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

Wil26. Christopher Williamson. Secret sharing at the shannon ceiling, 2026.