

Deniable Secret Sharing: Towards Practical Faking

Ignacio Amores¹, Tamar Ben David³ , Matteo Campanelli²,
Sebastian Kolby¹, Eran Omri³, Anat Paskin-Cherniavsky³, and
Sophia Yakubov¹

¹ Aarhus University; {amores-sesar,sk,sophia.yakubov}@cs.au.dk

² Offchain Labs and University of Tartu; binarywhalesinternaryseas@gmail.com

³ Ariel University; {omrier,anatpc}@ariel.ac.il,tamaryahav123@gmail.com

Abstract. *Deniable secret sharing* (DSS; Canetti *et al.*, TCC 2025) is designed for a scenario where an adversary is able to bribe or coerce all shareholders to divulge their shares of a secret. A deniable secret sharing scheme is equipped with a *faking function* that enables shareholders to produce fake shares that *look* like real shares, but do not actually reveal any additional information about the secret.

Existing DSS constructions require the fakers to jointly compute on their shares, without considering how this computation might be realized. In this paper, we investigate two crucial properties of such computations: the extent to which the fakers themselves learn information about the secret (*privacy against fakers*), and the number of messages they need to exchange (*message complexity*). Message complexity can be seen as a measure of faking conspicuousness, if the adversary is able to observe network traffic.

We prove that to obtain the strongest notion of DSS, any information-theoretic construction requires the fakers to exchange a non-zero number of messages, which could be large (depending on the access structure). We describe a general construction that matches this lower bound in many cases, e.g. for threshold access structures. Furthermore, for linear secret sharing schemes, we prove that certain unqualified sets must learn the secret as a result of faking.

Moving to the computational setting, we describe two constructions which require *no interaction beyond the knowledge of who the fakers are*. This implies that both inconspicuousness and privacy against fakers are achieved. The first construction requires the share size to be exponential in the number of shareholders, but assumes only a PRG. The second construction has small shares, but assumes virtual black-box obfuscation. We see the latter construction as a stepping stone for a future *iO*-based instantiation.

All of our lower bounds and constructions consider *general* access structures.

Table of Contents

Deniable Secret Sharing: Towards Practical Faking	1
<i>Ignacio Amores[Ⓜ], Tamar Ben David[Ⓜ], Matteo Campanelli[Ⓜ], Sebastian Kolby[Ⓜ], Eran Omri[Ⓜ], Anat Paskin-Cherniavsky[Ⓜ], and Sophia Yakoubov[Ⓜ]</i>	
1 Introduction	3
1.1 Pragmatic DSS and Its Challenges	3
1.2 Our Results	4
1.3 Technical Overview	5
1.3.1 DSS with Fake Hiding	5
1.3.2 DSS with Fake-Real Indistinguishability	6
1.4 Notation	8
2 Definitions of Deniable Secret Sharing	8
2.1 Properties of the Faking Function	9
2.2 Properties of the Protocol	11
2.2.1 Message Complexity	12
3 Lower Bounds for Fake Hiding Message Complexity	13
4 Constructions with Minimal Message Complexity	16
4.1 Notation	16
4.2 A General Construction	16
5 Achieving FRI with Non-Interactive Faking	18
5.1 A First Construction from PRGs	19
5.2 Replacing Exponential Randomness with Obfuscated Programs	20
5.3 Challenges when moving to indistinguishability obfuscation	23
A A Stronger Lower Bound for Linear Secret Sharing	25

Acknowledgments. The work of Tamar Ben-David and Eran Omri was supported by the Israel Science Foundation (ISF), Grant 1614/25. The work of Ignacio Amores-Sesar has been funded by the Cryptographic Foundations for Digital Society, CryptoDigi, DFF Research Project 2, Grant ID 10.46540/3103-00077B. The work of Sophia Yakoubov was supported in part by the Danish Independent Research Council under Grant-ID DFF-2064-00016B (YOSO) and Grant-ID DFF-5329-00028B (Deniable Secret Sharing). Matteo Campanelli was supported in part by the Estonian Research Council grant PRG2531. Sebastian Kolby was supported by the Next-Generation Cybersecurity programme (Digital Research Centre Denmark (DIREC), National Defence Technology Centre (NFC) and Danish Industry Foundation under project 2025-0766).

1 Introduction

This work studies Deniable Secret Sharing (DSS), a notion recently introduced by Canetti *et al.* [CDK⁺25].

Why Deniability in Secret Sharing? In standard secret sharing we guarantee that an adversary corrupting an *unqualified* subset of shareholders (e.g., one too small to reconstruct the secret) learns nothing about a secret m . A recent line of work [BGG⁺20,GHK⁺21,BHK⁺22] has studied how to combine communication efficiency with security against adaptive corruption in large-scale networks by giving shares of m to only some of the parties in the network, while hiding their identities. Such approaches, however, are still susceptible to attacks where adversaries do not need to know the identity of the shareholders to be able to learn their shares; for example, they may still be vulnerable to *bribing* attempts where an adversary may post a bounty and simply wait for shareholders to come forward and sell their shares.

Deniability closes this gap and strengthens the security properties of standard secret sharing by making it resilient against adversaries bribing shareholders (or coercing them; see below). In order to do that, in Deniable Secret Sharing (DSS), shareholders can turn their real secret share into a *fake* one, i.e., one that looks like a real share, but that does not contribute to the recovery of the secret. As a consequence, an adversary would have little to gain by offering a bounty for shares or, e.g., coercing users at gunpoint (what they received after that corruption attempt might as well be a fake share).

Properties of DSS. In a DSS scheme, shareholders may be divided in their response to the adversary’s coercion. *Fakers* follow the faking mechanism prescribed by the DSS scheme. *Snitches*, on the other hand, hand over their true shares anyways. Privacy should hold as long as the snitches are unqualified; additionally, if all shareholders are fakers, the adversary should be unable to tell that faking took place.

In stronger flavors of DSS, the fakers’ identities are protected even if some shareholders fake while others snitch. In particular, in *fake real indistinguishability*, the adversary should be unable to tell which is the set of fakers and which is the set of snitches; in *fake hiding*, the adversary should be unable to tell that faking took place even if some shareholders snitch.

1.1 Pragmatic DSS and Its Challenges

Canetti *et al.* [CDK⁺25] defined *uncoordinated faking*, where fakers compute their fake shares using only their real shares, and a target fake message to which the fake shares should reconstruct. Uncoordinated faking can be done by the fakers locally (without communication), which is, in many ways, ideal. Unfortunately, Canetti *et al.* prove that uncoordinated faking is impossible in most settings. They instead turn to *coordinated faking*, where fake shares must be

computed based on all of the fakers’ real shares taken together. The fakers accomplish this by pooling their information.

With the aim of obtaining schemes that can be effective in real-world applications, we identify two shortcomings in the coordinated DSS constructions in [CDK⁺25]: (1) the faking process — wherein the fakers pool their information — might be *conspicuous* (even if the fake shares themselves are convincing), and (2) even though the fakers might be unauthorized, they learn the message during faking. To address these shortcomings, we identify two goals/desiderata for DSS designs.

Goal 1: Low Message Complexity. Though it follows from the lower bound of Canetti *et al.* that avoiding communication entirely is impossible, it is desirable to minimize the message complexity of the faking protocol, since the adversary might be able to see network traffic.

Goal 2: Privacy Against Fakers. It is desirable that any set of fakers that is unqualified according to the access structure remain unable to reconstruct the secret after engaging in faking. The fakers wish to respect the dealer’s wishes by helping preserve the privacy of the secret from the adversary; it is natural to respect the dealer’s wishes also when it comes to unqualified sets of share holders. This notion of “privacy against fakers” implies, in particular, that no single faker should learn the secret as a result of the faking protocol.

This work. Motivated by the desiderata above, we study the communication complexity of deniable secret sharing — taking message complexity as a proxy for how conspicuous a faking protocol is — and prove both lower and upper bounds. We additionally ask whether non-interactive faking is achievable at all (sidestepping the impossibility for uncoordinated faking in [CDK⁺25]; see the technical overview): non-interactive faking is immediately “maximally inconspicuous” and gives privacy against fakers for free, since no online messages are sent. All of our results hold for general access structures, whereas prior work was limited to the threshold case.

1.2 Our Results

Our main contributions are as follows.

- We show that in any information-theoretic DSS construction that achieves the strongest DSS property — fake hiding — the fakers need to send a non-zero number of messages⁴, which can be large depending on the access structure. As an example, for two-of- n threshold sharing, faking requires at least $2(n - 1)$ messages. Furthermore, for linear secret sharing schemes, we prove that certain unqualified sets must learn the secret as a result of faking.

⁴ Our result is stronger than previous results on impossibility of non-coordination from Canetti *et al.* since it holds even if the fakers know each others’ identities or hold correlated randomness.

- We show a general construction of information-theoretic DSS with fake hiding that meets our lower bound on message complexity.
- In the computational setting, we give two constructions that achieve both of our desiderata while *requiring no interaction* beyond the knowledge of who the fakers are: our first scheme is based only on PRGs but has share size exponential in the number of shareholders; our second scheme has small shares but uses virtual black-box obfuscation.

1.3 Technical Overview

Toward the goal of more pragmatic Deniable Secret Sharing, we begin the explicit study of the two properties we described above: namely, *message complexity* and *privacy against fakers*.

One might hope to achieve privacy against fakers by executing the fake function within secure multiparty computation (MPC) [GMW87], which avoids the explicit reconstruction of the real secret. However, there are two issues with this. First, general-purpose MPC has high communication overhead, so its use incurs high message complexity. Second, for information-theoretic constructions, it turns out that knowledge of the fake shares themselves reveals information; so, using an ideally secure faking process doesn't help.

We explore the extent to which low message complexity can be achieved in DSS with fake hiding, and in DSS with fake-real indistinguishability.

1.3.1 DSS with Fake Hiding Here, we focus on information-theoretic constructions. Since Canetti *et al.* [CDK⁺25] ruled out information-theoretic DSS with fake hiding when the fakers are unqualified or the snitches are qualified, we focus on the setting where the fakers are qualified and the snitches are unqualified. We explore the number of messages that need to be exchanged among fakers.

Lower Bounds on Message Complexity. As a first illustrative example, consider Shamir secret sharing for a $t+1$ -out-of- n threshold access structure. Fake hiding demands that the fakers' fake shares together with the snitches' real shares look like a sharing of m^{fake} . Thus, following the faking protocol, the fakers end up with points on a polynomial which has m^{fake} as the y-intercept. Additionally, the shares of the snitches in both this and original sharing of original message m agree. By linearity of Shamir, letting each party subtract the two shares, results in a valid sharing of $m - m^{\text{fake}}$ – in particular, the resulting shares lie on a polynomial p' with 0's on all snitches' points. Now, consider a set T of t fakers (which is maximally unqualified). This set holds t points on p' . They also know the value at at least one additional point held by a snitch (which evaluates to 0). They can this extrapolate the y-intercept of p' and recover $m - m^{\text{fake}}$, from which they recover m , as they know m^{fake} .

Our faker set T have now recovered the secret, violating privacy of Shamir, unless *some* message was sent to (some party) in T . In Appendix A we generalize

this intuition to perfect faking for any linear secret sharing scheme for general access structures where the fakers are qualified, and the snitches are not qualified and not redundant (at least one snitch belongs to a minterm).

In [Section 3](#), we prove a somewhat weaker claim for access structures as above for the more challenging setting of general (not necessarily linear) secret sharing schemes. More precisely, we show that in any perfect faking protocol for a set $G \setminus S$, where G is a minterm for which G contains one or more snitches, $T = G \setminus \text{snitches}$ must receive a message during the faking protocol. Here we can not rely on linearity, which makes the proof more involved. Note that in this theorem we do not show that information about the secret itself leaks to T , and leave it as an open problem. Still, even the somewhat weaker general theorem implies an interesting message complexity lower bound on the faking protocol. Construct a hypergraph on $V = \text{fakers}$ with hyperedges of the form $G \setminus \text{snitches}$ which G is a minterm as above. Then the number of messages sent by any faking protocol must be at least as large as the smallest vertex cover of this hypergraph ([Theorem 2](#)). This is the case, since a smaller number of messages will fail to send a message to at least one of the sets $G \setminus \text{snitches}$ as above. We show that this bound is tight for certain access structures by devising a matching upper bound [4](#). One example is that of the $t + 1$ -out-of- n access structure considered above, and let the snitches be a set of size t . Our lower bound implies that the hyperedge set in the above graph consists of all the singletons among the fakers - implying a lower bound of $|\text{fakers}|$. This is therefore tight up to a factor of two, since a simple protocol based on our approach in [Section 4](#) yields a share complexity of $|\text{fakers}| + t - 1$ in this case (and $|\text{fakers}| > t$, since they are qualified). See further discussion on the meaning of our results toward the end of [Section 3](#).

Constructions with Low Message Complexity Next, using any linear secret sharing scheme (for any access structure) as a base, we instantiate faking protocols with just $\Theta(n)$ message complexity (where n is the number of shareholders), which is optimal up to a constant factor. In the most generic version of our construction, the fakers choose a leader (non-interactively — e.g. as the lexicographically first among them), and reconstruct the real message m to the leader. The leader then computes a sharing of $m^{\text{fake}} - m$ with the constraint that all of the snitch shares are zero, and distributes this sharing to the other fakers. Each other faker then computes their fake share as a sum of their real share and this new share.

The message complexity of this construction is $\Theta(n)$. When Shamir secret sharing (which has shares of optimal size) is used as the base secret sharing scheme, the *communication* complexity — the number of bits sent — is $\Theta(n|m|)$.

1.3.2 DSS with Fake-Real Indistinguishability We complement our fake-hiding results with constructions achieving the weaker notion of fake-real indistinguishability (or FRI, [Definition 5](#)), in which real and fake share-sets are indistinguishable even when it is clear which message each set reconstructs to.

When our goal is FRI, it suffices to generate a fresh-looking sharing of the target fake message m^{fake} . Privacy against fakers comes for free, as the generation

of the fake shares is totally independent of the original shares and message. This approach has one issue: it requires conspicuous communication among fakers.

Instead, we turn our attention to the more challenging problem of achieving FRI *non-interactively*. Specifically, we will show how fakers can produce their own fake shares without any online communication, given just their individual real share and knowledge of the identities of their fellow fakers F ⁵. Being non-interactive, such an approach achieves ideal inconspicuousness. For the same reason, it also gets privacy against fakers: for non-interactive faking, breaking privacy against fakers trivially implies breaking privacy.

With Exponential Share Size, in Minicrypt. Achieving FRI in Minicrypt is quite easy and clean. The dealer generates independent randomness for each of the possible set of fakers, and includes in party i 's share all the randomness for sets which i is a member of.

When faking, the fakers use the randomness corresponding to their faking set to locally generate a fresh sharing of m^{fake} , and use their part of the output as their fake share. This may be done for any underlying secret sharing scheme.

Now, though, since randomness for every possible faking set is part of the shares, a coercing adversary will expect to be given this randomness. If our fakers hand over their original randomness, the adversary will be able to tell that their share is fake just by re-running the secret sharing algorithm on this randomness and seeing if the output matches. If our fakers make up new randomness, they run the risk of contradicting their fellow fakers, which is also a giveaway.

Instead, the fakers input their randomness into a PRG: they include the first half of the output directly as the randomness in their fake share, and run the secret sharing algorithm with the second half of the randomness to generate their fake shares.

The issue with this construction is that the shares include randomness for each possible set of fakers, and the number of possible sets of fakers is exponential in the number of shareholders.

With Polynomial Share Size, From Ideal Obfuscation. We ask whether an exponential share size is inherent. We show that it is not: assuming *ideal* obfuscation [BGI⁺01], the dealer can instead distribute an obfuscated program that generates, on demand, the shared random value for any faker set the caller belongs to (Section 5.2, Theorem 8). The program controls which randomnesses each party is allowed to obtain by giving each party an authenticated encryption of their index and a common PRF key. To obtain the randomness for a set of fakers a party must input a ciphertext containing the an index in that set allowing the PRF to be evaluated on the key. We prevent the adversary from obtaining the same randomness we extend the program to output new ciphertexts with updated PRF keys. This can again extend any underlying secret sharing scheme to achieve FRI.

⁵ Providing the set F as an argument to each faking party is also the key idea that allows us to circumvent the impossibility results on fully uncoordinated faking in [CDK⁺25, Section 5.5].

Our hope is that this construction, currently based on VBB obfuscation [BGI⁺01], may be a stepping stone toward an instantiation from indistinguishability obfuscation ($i\mathcal{O}$) [SW14]⁶. We outline the obstacles to replacing the idealized obfuscation with $i\mathcal{O}$ in Section 5.3 and leave a full $i\mathcal{O}$ -based construction as an interesting open problem stemming from our work.

1.4 Notation

For $n \in \mathbb{N}^+$, we let $[n]$ denote $\{i \in \mathbb{N}^+ | 1 \leq i \leq n\}$. Throughout the paper, $\lambda \in \mathbb{N}$ denotes the security parameter. A function $\text{negl} : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is *negligible* if for every positive polynomial p there exists $\lambda_0 \in \mathbb{N}$ such that $\text{negl}(\lambda) < 1/p(\lambda)$ for all $\lambda \geq \lambda_0$. Unless stated otherwise, all algorithms (including the adversary) implicitly take 1^λ as an additional input, and whenever we write negl without specifying its argument we mean a function that is negligible in λ .

2 Definitions of Deniable Secret Sharing

The purpose of secret sharing is to hide a secret from sets of parties which the secret owner — referred to as the dealer — deems “unqualified”, while enabling sets of parties which the dealer deems “qualified” to recover the secret. This description of qualification is known as an *access structure*.

Definition 1 (Access Structure). Let $\{p_1, \dots, p_n\}$ be a set of parties. A collection $\Gamma \subseteq 2^{\{p_1, \dots, p_n\}}$ is *monotone* if $B \in \Gamma$ and $B \subseteq C$ imply that $C \in \Gamma$. An *access structure* is a monotone collection $\Gamma \subseteq 2^{\{p_1, \dots, p_n\}}$ of non-empty subsets of $\{p_1, \dots, p_n\}$. Sets in Γ are called *authorized*, and sets not in Γ are called *unauthorized*.

A set B is called a *minterm* — or minimally qualified set — of Γ if $B \in \Gamma$ and for every $C \subsetneq B$, the set C is unauthorized. Dually, a set B is called a *maxterm* — or maximally unqualified set — of Γ if $B \notin \Gamma$ and for every $C \supsetneq B$, the set C is authorized. A party $p \in \{p_1, \dots, p_n\}$ is called *redundant* if there is no minterm of Γ that contains it.

The most commonly studied access structure is the *threshold* access structure, where any $t+1$ of the n shareholders are qualified, and any t or fewer shareholders are unqualified. In this paper, we look beyond the threshold access structure to more general access structures.

Definition 2 (Secret Sharing Scheme - Informal). A secret sharing scheme $\mathcal{S}$ for an access structure Γ is a tuple of algorithms $(\text{Share}, \text{Rec})$:

$\text{Share}(x) \rightarrow (sh_1, \dots, sh_n)$ shares a secret x .
 $\text{Rec}(\{sh_i\}_{i \in Q}) \rightarrow x'$ reconstructs a secret x' .

⁶ This hope is warranted by precedents, notably in the context of *deniability*, where it was possible to tweak primitives based on ideal obfuscations into designs based on $i\mathcal{O}$. See, e.g., [SW14].

Any secret sharing scheme should have correctness, meaning that a qualified set of shares reconstructs to the shared secret, and privacy⁷, meaning that an unqualified set of shares should reveal nothing about the shared secret.

A *deniable* secret sharing scheme is a secret sharing scheme additionally equipped with a function `fake`:

$\text{fake}(\{sh_i\}_{i \in F}, m^{\text{fake}}) \rightarrow \overline{sh}_i$ takes in a set of real shares and a target fake message, and produces fake shares which look like the real thing, but aren't.

Protocols realizing fake. The function `fake` is evaluated by the set of fakers via a protocol Π_{fake} . We say that Π_{fake} realizes `fake` if, when run by shareholders in F where each party $i \in F$ inputs (sh_i, m^{fake}) , the output of Π_{fake} has the same distribution as the output of $\text{fake}(\{sh_i\}_{i \in F}, m^{\text{fake}})$. We say “ Π_{fake} fakes for F ” when it is executed by the shareholders in F .

We separate our discussion of the properties of deniable secret sharing into properties of the function `fake` itself (Section 2.1), and properties of the protocol Π_{fake} (Section 2.2). The properties of the function have already been defined by Canetti *et al.* [CDK⁺25] for the threshold access structure; we extend them to general access structures. The properties of the protocol are new in this paper.

All of the properties we define (Definition 4, Definition 5, Definition 6 and Definition 7) can be *perfect* or *computational*. Below, we define the properties as computational; to get the perfect definition, simply (1) remove the restriction that the adversary be PPT, and (2) remove the “negl” from the inequality, instead requiring that the adversary have no advantage in the guessing game it is playing.

2.1 Properties of the Faking Function

In a deniable secret sharing scheme, privacy should hold even if the adversary has some fake shares in addition to the unqualified set of real shares.

Deniable secret sharing schemes might additionally have the following properties:

Fake-Real Indistinguishability: given two sets of shares belonging to disjoint sets of shareholders, an adversary should be unable to tell which set is real and which set is fake, even if it is clear (e.g. because one of the sets might be qualified) which message each set of shares is consistent with.

Faker Anonymity: given shares from all shareholders, the adversary should be unable to tell which shares are real, and which shares are fake.

Fake Hiding: given shares from all shareholders, the adversary should be unable to tell whether faking took place.

⁷ We give a rigorous definition of privacy — modified to account for deniability — in Definition 4.

At a minimum, given a set of shares which is either entirely real or entirely fake, an adversary should be unable to guess which it is. This property is captured by fake-real indistinguishability for the trivial partition of shareholders into $[n]$ and $\emptyset$ (or, equivalently, by fake hiding for the same partition).

Definition 3 (Deniable Secret Sharing). A secret sharing scheme $(\text{Share}, \text{Rec})$ for access structure Γ (Definition 2) is deniable with faking function fake if $(\text{Share}, \text{Rec}, \text{fake})$ satisfies Γ -privacy (Definition 4) and $([n], \emptyset)$ -fake real indistinguishability (Definition 5).

Below, we replicate the definitions of privacy (Definition 4), fake-real indistinguishability (Definition 5) and fake hiding (Definition 6) introduced by Canetti *et al.* [CDK⁺25], with several minor changes:

- We adapt the definitions to accommodate general access structures.
- We run fake on all faked shares jointly, which is consistent with the coordinated faking notion of Canetti *et al.*, but notably does not allow faking to *different* messages by different faking parties, as captured in the original definition of privacy.

We omit faked anonymity because it is not directly relevant to our results. We begin with privacy, which strengthens the standard privacy of secret sharing to require that the secret remain hidden even when the adversary is given fake shares from the fakers in addition to real shares from any unqualified set of snitches.

Definition 4 (Γ -Privacy). A scheme is Γ -private for access structure Γ if for any ...

- partition of the parties into F and S s.t. $S \notin \Gamma$;
- messages $m_0, m_1, m^{\text{fake}}$ s.t. $|m_0| = |m_1| = |m^{\text{fake}}|$;
- PPT adversary $\mathcal{A}$;

the following holds:

$$\Pr \left[\mathcal{A} \left(\begin{array}{c} m_0, m_1, m^{\text{fake}}, \\ F, S, \\ \{sh_i\}_{i \in [n]} \end{array} \right) = b \mid \begin{array}{l} b \leftarrow \{0, 1\}; \\ (sh_1^{\text{real}}, \dots, sh_n^{\text{real}}) \leftarrow \text{Share}(m_b); \\ \{sh_i\}_{i \in F} \leftarrow \text{fake}(\{sh_i^{\text{real}}\}_{i \in F}, m^{\text{fake}}); \\ \text{For } i \in S : sh_i := sh_i^{\text{real}} \end{array} \right] = \frac{1}{2} + \text{negl.}$$

We now define fake-real indistinguishability (FRI), which requires that given a set of shares of m_0 from parties in P_0 and a set of shares of m_1 from parties in P_1 , it should be impossible to tell which are real, and which are fake. When $P_0 = [n]$ and $P_1 = \emptyset$, FRI implies that when all parties fake, the adversary should not be able to tell the fake shares from a fresh sharing of the fake message. This is a minimal requirement for deniable secret sharing.

FRI is also a property which is interesting to consider even when both sets of parties are qualified, and thus privacy cannot hold. It can be seen as a “fallback” security notion, which gives the fakers some deniability even when the snitches are qualified.

Definition 5 ((P_0, P_1)-Fake-Real Indistinguishability). A scheme is (P_0, P_1)-fake-real indistinguishable for a partition P_0, P_1 of $[n]$ if for any ...

- messages m_0, m_1 s.t. $|m_0| = |m_1|$;
- PPT adversary $\mathcal{A}$;

the following holds:

$$\Pr \left[\mathcal{A} \left(\begin{array}{c} m_0, m_1, \\ P_0, \\ P_1, \\ \{sh_i\}_{i \in [n]} \end{array} \right) = b \mid \begin{array}{l} b \leftarrow \{0, 1\}; \\ (sh_1^{real}, \dots, sh_n^{real}) \leftarrow \text{Share}(m_b); \\ \{sh_i\}_{i \in P_{1-b}} \leftarrow \text{fake}(\{sh_i^{real}\}_{i \in P_{1-b}}, m_{1-b}); \\ \text{For } i \in P_b : sh_i := sh_i^{real} \end{array} \right] = \frac{1}{2} + \text{negl}.$$

We next define fake hiding, which is the strongest property of a deniable secret sharing scheme: it requires that a set of mixed fake and real shares look just like a fresh — consistent — set of shares. Note that FRI does not require that fake and real shares look consistent.

Definition 6 (F -Fake Hiding). A scheme is F -fake hiding for $F \subseteq [n]$ if for any ...

- messages m_0, m_1 s.t. $|m_0| = |m_1|$;
- PPT adversary $\mathcal{A}$;

the following holds:

$$\Pr \left[\mathcal{A} \left(\begin{array}{c} m_0, m_1, \\ F, S, \\ \{sh_i\}_{i \in [n]} \end{array} \right) = b \mid \begin{array}{l} b \leftarrow \{\text{real}, \text{fake}\}; \\ \text{if } b = \text{real} : (sh_1, \dots, sh_n) \leftarrow \text{Share}(m_1); \\ \text{if } b = \text{fake} : \\ \quad (sh_1, \dots, sh_n) \leftarrow \text{Share}(m_0); \\ \quad \{sh_i\}_{i \in F} \leftarrow \text{fake}(\{sh_i\}_{i \in F}, m_1) \end{array} \right] = \frac{1}{2} + \text{negl}.$$

2.2 Properties of the Protocol

In this section, we discuss properties of the protocol Π_{fake} which the fakers use to evaluate the function `fake`. In the work of Canetti *et al.* [CDK+25], the protocol was idealized: fakers were considered to simply tell one another their shares, and locally evaluate the same algorithm on the now-same input. We consider the protocol Π_{fake} explicitly. We analyze two aspects of Π_{fake} :

1. How much fakers learn about the secret (Definition 7); and
2. The message complexity of the protocol (Definition 8).

These are very tightly linked. Observe, for instance, that if the message complexity is zero, then the fakers learn nothing about the secret, by the privacy of the scheme.

Definition 7 (*F*-Privacy Against Fakers). We say that a DSS scheme satisfies privacy against fakers for the faker set F and for access structure Γ if for any ...

- unqualified subset T of F (i.e., with $T \subset F, T \notin \Gamma$);
- messages m_0, m_1, m^{fake} s.t. $|m_0| = |m_1| = |m^{fake}|$;
- PPT adversary $\mathcal{A}$;

the following holds:

$$\Pr \left[\mathcal{A} \left(\begin{matrix} m_0, m_1, m^{fake} \\ F, \text{view}^T \end{matrix} \right) = b \mid \begin{matrix} b \leftarrow \{0, 1\}; \\ (sh_1^{real}, \dots, sh_n^{real}) \leftarrow \text{Share}(m_b) \end{matrix} \right] = \frac{1}{2} + \text{negl},$$

where view^T is the local joint view of parties in T during an execution of the faking protocol on input the real shares and the target fake message m^{fake} , i.e. it is a function of $(T, F, m^{fake}, \{sh_i^{real}\}_{i \in F})$.

Privacy against fakers (Definition 7) requires that *nothing* leak to unqualified sets as a result of the faking protocol. In cases where this is unachievable, an alternative goal might be to minimize such leakage, e.g. by minimizing the number of maximally unqualified sets — maxterms — that learn new information about the secret.

2.2.1 Message Complexity We provide definitions modeling the message complexity in the context of deniable secret sharing.

On the choice of metric. Several natural metrics measure how conspicuous a faking protocol is. We could make sure that each faker communicates with as few other fakers as possible before deciding upon its share. Other natural metrics are total number of messages, and incoming/outgoing message locality: we could make sure that no faker needs to receive or send messages from or to too many other fakers, since a faker corresponding with many others may itself raise suspicion. In this work, we focus on the total number of messages as the metric to minimize.

Definition 8 (Message Complexity of a Protocol). Let Π be a protocol. The message complexity $\text{MC}(\Pi)$ of Π is the number of messages exchanged by the participants during an execution of Π .

Definition 9 (Message Complexity of a Function). Let f be a k -input function. Let PROTS be the set of all k -party protocols Π_f which realize f , assuming each of the k inputs is held by a different party. The message complexity $\text{MC}(f)$ of f is the minimum message complexity of any protocol Π_f realizing f :

$$\text{MC}(f) = \min_{\Pi_f \in \text{PROTS}} \text{MC}(\Pi_f).$$

Definition 10 (FH Message Complexity of a Secret Sharing Scheme). Let $\mathcal{S} = (\text{Share}, \text{Rec})$ be a secret sharing scheme for access structure Γ . Let F be a set of fakers, and let FHFUNCS_F be the set of all faking functions fake which extend $\mathcal{S}$ such that $(\text{Share}, \text{Rec}, \text{fake})$ is a deniable secret sharing scheme (Definition 3) with F -fake-hiding. The FH message complexity $\text{MC}_{FH}(\mathcal{S}, F)$ of $\mathcal{S}$ for F is the smallest message complexity $\text{MC}(\text{fake}, F)$:

$$\text{MC}_{FH}(\mathcal{S}, F) = \min_{\text{fake} \in \text{FHFUNCS}_F} \text{MC}(\text{fake}, F).$$

Definition 11 (FH Message Complexity of an Access Structure). Let Γ be an access structure. Let SCHEMES be the set of all secret sharing schemes $\mathcal{S} = (\text{Share}, \text{Rec})$ for Γ . Let F be a set of fakers. The FH message complexity $\text{MC}_{FH}(\Gamma, F)$ of Γ for F is the smallest FH message complexity $\text{MC}(\mathcal{S}, F)$: let

$$\text{MC}_{FH}(\Gamma, F) = \min_{\mathcal{S} \in \text{SCHEMES}} \text{MC}(\mathcal{S}, F).$$

Let FAKERS be the set of all faker sets F s.t. $F \in \Gamma$ and $S = [n] \setminus F \notin \Gamma$. The FH message complexity $\text{MC}_{FH}(\Gamma)$ is defined as

$$\text{MC}_{FH}(\Gamma) = \min_{\mathcal{S} \in \text{SCHEMES}} \left(\max_{F \in \text{FAKERS}} \text{MC}(\mathcal{S}, F) \right).$$

3 Lower Bounds for Fake Hiding Message Complexity

In this section, we prove a lower bound on message complexity of faking protocols for DSS with fake hiding.⁸

Theorem 1. Let Γ be an access structure, and let $\mathcal{S}_\Gamma = (\text{Share}, \text{Rec})$ be a (perfect) secret sharing scheme realizing Γ . Let fake be a faking function such that $(\text{Share}, \text{Rec}, \text{fake})$ is a perfect DSS satisfying fake hiding (Definition 6).

Let F be a qualified set of fakers such that $S = \overline{F}$ is an unqualified set (of snitches).

Let Π be a protocol realizing fake for our set of fakers F .

Let $F' \subseteq F$ and let $S' \subseteq S$ be non-empty sets such that $F' \cup S'$ is a minterm. Then, (at least one party in) F' receives some message from a party $p \in F \setminus F'$ during the execution of Π .

Remark 1. Note that in the above setting, unless S is redundant (in the sense that no snitch is part of a minterm), there must exist $F' \subseteq F$ and a non-empty $S' \subseteq S$ such that $F' \cup S'$ is a minterm.

Proof. Let Γ , F , and Π be as in the theorem. Denote by $M = \{m_1, \dots, m_\ell\}$ the domain of secrets. Assume towards a contradiction that Π always outputs

⁸ We note our lower bound also generalizes to the setting where parties share some correlated randomness generated at sharing time.

a valid sharing (f) for some fixed $f \in M$, where no party in F' receives any message from a party not in F' .⁹

Consider an execution of Π on a sharing of a secret sampled from the uniform distribution over the secret domain M . Let τ be the random variable describing the sampled message. Fix some value $\bar{u}^{(0)}$ in $\text{support}((M)_{F'})$. That is, $\bar{u}^{(0)}$ is a tuple of possible shares received by the parties in F' in some execution of $\mathcal{S}_F$. There exist disjoint non-empty subsets of possible values $V_{m_1}^{(0)}, \dots, V_{m_g}^{(0)}$ of $(\tau)_{S'}$ consistent with $\bar{u}^{(0)}$ (recall that $(m)_T$ is tuple of shares for a secret m , given to the parties in a set T). Specifically, for every $1 \leq i \leq g$ and every element (vector of shares) $\bar{v}_{m_i}^{(0)} \in V_{m_i}^{(0)}$, it holds that $\bar{u}^{(0)}$ together with $\bar{v}_{m_i}^{(0)}$ form a valid (partial) sharing (for $F' \cup S'$) of the secret m_i . Indeed, these sets are non-empty by the privacy of $\mathcal{S}_F$, and they are disjoint by correctness of $\mathcal{S}_F$.

Assume, without loss of generality, that the protocol Π is deterministic (by perfect correctness, we can fix the randomness of Π), and that given shares $(\tau)_F = \bar{u}^{(0)}$, the protocol Π transforms it into $\bar{u}^{(1)}$ (deterministically and without interaction with parties outside F'). Consider the sampled value of $(\tau)_{S'}$, sampled together with $\bar{u}^{(0)}$ in the current execution of $\mathcal{S}_F$. By the above, even given that $(\tau)_{F'} = \bar{u}^{(0)}$, the value of $\bar{v}^{(0)}$ can be any of the elements in $\bigcup_i V_i^{(0)}$ with non-zero probability (as τ is not determined by $\bar{u}^{(0)}$).

Assume that there exists some $\bar{v}' \in \bigcup_i V_{m_i}^{(0)}$ such that $\bar{v}', \bar{u}^{(1)}$ does not reconstruct into f (in particular, it may form an invalid partial sharing altogether). Then, with non-zero probability the resulting sharing is not consistent with f , contradicting the perfect correctness of the faking scheme. Thus, it must hold that $\bigcup_i V_{m_i}^{(0)} \subseteq V_f^{(1)}$, where $V_f^{(1)}$ is the set of share tuples for the parties in S' complementing $\bar{u}^{(1)}$ to a sharing of f .

Now, let us think of $\bar{u}^{(1)}$ as an initial value of $(\tau)_{F'}$ of some sharing. Again, consider $V_{m_1}^{(1)}, \dots, V_{m_g}^{(1)}$ induced by $\bar{u}^{(1)}$ and let $\bar{u}^{(2)}$ denote the faked value on input $\bar{u}^{(1)}$. To avoid the first situation (leading to incorrect recovery with non-zero probability), we must have $\bar{v}^{(1)}, \bar{u}^{(2)}$ decode to f for every $\bar{v}^{(1)} \in \bigcup_i V_i^{(1)}$. Continuing in this manner, so that the first option does not hold at any step, we obtain a sequence of values $\bar{u}^{(0)}, \bar{u}^{(1)}, \bar{u}^{(2)}, \dots$ of $(M)_{F'}$ lying in its support. Since the share domain of F' is finite, it must be the case that $\bar{u}^{(j+1)} = \bar{u}^{(j')}$ for some $1 \leq j' \leq j$. We note that

Claim. Option 1 holds when moving from step j to step $j+1$ in the sequence. That is, $\bigcup_{i \in [g]} V_i^{(j)} \not\subseteq V_f^{(j')} = V_f^{j+1}$

Assume the contrary. By construction, for every $k \leq j$, $(*) \bigcup_{i \in [g]} V_i^{(k)} \subset \bigcup_{i \in (g)} V_i^{(k+1)}$, as $\bigcup_{i \in [g]} V_i^{(k)} \subseteq V_f^{(k+1)}$, and the rest of the $V_i^{(k+1)}$'s for $i \neq f$ are non-empty. Thus, by induction, $\bigcup_{i \in [g]} V_i^{(j')} \subset \bigcup_{i \in (g)} V_i^{(j)}$. If $j' < j$, we have $\bigcup_{i \in (g)} V_i^{(j)} \subset$

⁹ Note that we put no constraints on the output distribution of Π . Specifically, we only require that the shares of given to $F' \cup S'$ are some valid sharing of f . Hence, we prove a somewhat stronger statement.

$\bigcup_{i \in [g]} V_i^{(j')}$ and $\bigcup_{i \in [g]} V_i^{(j')} \subset \bigcup_{i \in [g]} V_i^{(j)}$, which is a contradiction. If $j = j'$, it does not hold, in this case $\bigcup_{i \in [g]} V_i^{(j)} = \bigcup_{i \in [g]} V_i^{(j+1)}$, as $V_i^{(j)} = V_i^{(j+1)}$ for each $i \in [g]$. However, this contradicts (*). Thereby, contradicting the fact that a sequence $\bar{u}^{(0)}, \dots, \bar{u}^{(j)}$ as above exists. This ends the proof of the claim, and implies that $\Pi_{\mathcal{S}_F, F}$ does not exist. $\square$

Note that the above theorem strongly relies on perfect correctness of $\Pi_{\mathcal{S}_F, F}$. However, the proof could tolerate a weaker notion of privacy of the underlying secret sharing scheme $\mathcal{S}_F$ (all we need is that the $V_i^{(k)}$'s above are non-empty. That is, we need that an unqualified set can not rule out any of the secrets.). A useful corollary from Theorem 1 is the following combinatorial lower bound.

Theorem 2. *Let $n \geq 2$, and let $(\Gamma, \mathcal{S}_F, F, S)$ be as in Theorem 1. For information-theoretic DSS Π for $(\mathcal{S}_F, F, S)$ scheme in the CR model that achieves FH, it must be the case that for every subset $F' \subseteq F$ for which there exists a non empty $S' \subseteq S$ where $F' \cup S'$ is minterm, (at least one party in) F' receives some message from a party $p \in F \setminus F'$. Then the FH message complexity of (Γ, F, S) is at least as large as the minimum vertex cover in the hypergraph $G(F, E)$ where $E = \{D \setminus S \mid D \text{ is a minterm of } \Gamma, \text{ and } D \cap S \neq \emptyset\}$.*

Proof (Proof sketch). By Theorem 1, for every $F' \subseteq F$ for which there exists $\emptyset \neq S' \subseteq S$ such that $F' \cup S'$ is a minterm of Γ , at least one party $p' \in F'$ must receive a message from some $p \in F \setminus F'$ during the faking procedure. Equivalently, the set $R \subseteq F$ of message-receivers must intersect $D \cap F = D \setminus S$ for every minterm D with $D \cap S \neq \emptyset$ — i.e., R is a vertex cover of the hypergraph $G(F, E)$. The total number of messages is therefore at least $|R| \geq$ the size of a minimum vertex cover of G . $\square$

Below we state another impossibility result that follows by a straightforward modification of the proof of Theorem 1 (one that is not merely a lower bound on message complexity).

Corollary 3. *Let Γ be an access structure, and let $\mathcal{S}_F = (\text{Share}, \text{Rec})$ be a (perfect) secret sharing scheme realizing Γ . Let F be an unqualified set of fakers such that $S = \bar{F}$ is a non-redundant set (of snitches). Then there exists no faking function fake for which $(\text{Share}, \text{Rec}, \text{fake})$ which has a DSS protocol implementing it with perfect fake hiding (Definition 6).*

Proof (Proof sketch). There exists $\emptyset \neq S' \subseteq S$ such that $F \cup S'$ is a minterm: it exists as S is non redundant, so at least $F \cup S$ is qualified. Suppose for contradiction that such a scheme Π exists. Applying reasoning similar to Theorem 1 to $F' = F$, demonstrates that F' must receive a message during faking. But this means someone from S would need to send it, which is impossible by the communication pattern of a faking protocol. $\square$

Discussion. The proof of [Theorem 1](#) has two implications. First, *some* online communication is necessary for faking; [Theorem 2](#) quantifies this, and [Construction 2](#) shows the bound is tight in many cases. Second, certain unqualified subsets of fakers necessarily learn information about the secret during faking — for linear schemes the leakage is total ([Theorem 9](#), [Appendix A](#)). Studying the tradeoff between message complexity and faker leakage in general is an interesting direction we leave to future work.

In the next section, we prove that there exist access structures with qualified faker sets, and unqualified non-redundant snitches sets which admit FH schemes, with a minimal possible message complexity. We do not attempt to optimize leakage, and in fact purposefully allow potentially unnecessary leakage for the sake of optimizing message complexity — see [Section 4](#) for details. We leave the study of the tradeoffs between message complexity (a measure of conspicuousness) and leakage complexity to future work.

4 Constructions with Minimal Message Complexity

In this section, we consider the question of devising perfectly fake hiding faking protocols. While we allow coordination (in light of the lower bounds from [Section 3](#)), our goal is to minimize the number of messages sent.

In the most naïve constructions, all fakers would send messages to all other fakers, resulting in $O(|F|^2)$ message and communication complexity. In [Construction 2](#) we instead leverage a designated leader to lower the message complexity to $\Theta(|F|)$ for any linear secret sharing scheme.

4.1 Notation

Throughout this section, when describing faking protocols for a *linear* secret sharing scheme $\mathcal{S}$ implementing some access structure Γ , we denote a sharing of m according to $\mathcal{S}$ via (m) . We also denote via $(m)^T$ a sharing according to $\mathcal{S}$ with the special property that all shares given to parties in T are 0 (vectors). Such a sharing can always be computed as long as T is not qualified by sampling a sharing of m conditioned on the shares of T being zero; perfect privacy of $\mathcal{S}$ for the unqualified set T guarantees the conditional support is non-empty.

4.2 A General Construction

We start by extending any perfect linear secret sharing scheme into a DSS by specifying a faking function ([Construction 1](#)), and then describe a protocol for evaluating this faking function ([Construction 2](#)).

Construction 1 (DSS Scheme) *Let $n \geq 2$, and let $\Gamma \subseteq 2^{[n]}$ denote an access structure. Let $\mathcal{S}$ be a linear secret sharing scheme realizing Γ .*

*We extend $\mathcal{S}$ with a faking function **fake** — which requires that the fakers F be qualified, that the snitches $S = \overline{F}$ be unqualified, and that T be an unqualified set with $S \subseteq T$ — as follows:*

$\text{fake}(\{sh_i\}_{i \in F}, m^{\text{fake}})$:

- Reconstruct the real message as $m \leftarrow \text{Rec}(\{sh_i\}_{i \in F})$.
- Compute a random sharing $(m^{\text{fake}} - m)^T$. Let sh'_i denote party i 's share in this sharing.
- For each party $i \in F$, set $\overline{sh}_i = sh_i + sh'_i$.
- Return $\{\overline{sh}_i\}_{i \in F}$.

We observe that [Construction 1](#) is *correct*, by the linearity of $\mathcal{S}$. [Construction 1](#) is also *private* ([Definition 4](#)). Consider the given set of shares: the set of fakers' fake shares and snitches' real shares (which jointly reconstruct to m^{fake}). For any sharing (m) of m (under the constraint that the sharing matches the snitches' shares), there exists a unique sharing $(m^{\text{fake}} - m)^T$ which explains the given shares as a faking from (m) . By the privacy of $\mathcal{S}$, there is an equal number of such sharings (m) for each m . It follows that there is an equal number of plausible $(m^{\text{fake}} - m)^T$ for each m . Finally, [Construction 1](#) is *perfect fake-hiding*, since the resulting sharing is identically distributed to a fresh sharing of m^{fake} .

Construction 2 (DSS Faking Protocol) Let n , Γ , and $\mathcal{S}$ be as in [Construction 1](#). Let F be qualified ($F \in \Gamma$) and $S = \overline{F}$ be unqualified ($S \notin \Gamma$). Let ...

- $Q \subseteq F, Q \in \Gamma$ be the smallest qualified set of fakers, and T be the largest unqualified set that contains the snitches (i.e. $S \subseteq T$). If Q and T are not unique (i.e. there are multiple smallest qualified sets of fakers, and/or multiple largest unqualified sets that contain the snitches), choose Q and T in some canonical way — e.g. by lexicographic ordering — prioritizing Q and T that intersect.
- The leader p be the lexicographically first member of $Q \cap T$ if $Q \cap T \neq \emptyset$, and the lexicographically first member of Q otherwise.

We define the faking protocol Π_{fake} as follows:

1. All parties in Q send their shares to p .
2. Party p ...
 - (a) Reconstructs the real message m .
 - (b) Samples a random sharing $(m^{\text{fake}} - m)^T$.
 - (c) Sends each party not in T their share.
3. All parties not in T compute their fake share as the sum of their real share and the “difference” share they received from p .

Observe that Π_{fake} perfectly implements the faking function fake from [Construction 1](#).

Remark 2 (Beyond Linear Secret Sharing). A variant of [Construction 2](#) can be made to work even for secret sharing schemes that aren't linear, as long as they have the property that snitch shares can be uniquely recovered from some subset of the fakers' real shares. Party p would then recover the snitch shares, and compute a fresh sharing of m^{fake} under the constraint that the shares belonging to parties in T remain fixed.

Theorem 4 (Complexity of Construction 2). *Construction 2 has round complexity 2 and message complexity at most $2(n - 1)$.*

Proof. The message complexity of Construction 2 is $(|Q| - 1) + \min(n - |T|, n - 1) \leq 2(n - 1)$.

The term $(|Q| - 1)$ represents the messages that members of the qualified set Q send to p ; we subtract one because p is a member of Q and does not need to send a message to herself.

The term $\min(n - |T|, n - 1)$ represents the messages that p sends to parties outside of T . This is maximized when T is empty; in that case, all parties are outside of T , but again, p need not send a message to herself. $\square$

Next, we prove that there exist access structures Γ and sets of fakers and snitches (F, S) — qualified and unqualified, respectively — for which the bound in Theorem 1 is tight.

Example 5 (Threshold). Let $n \geq 2$, and let $\Gamma \subseteq 2^{[n]}$ denote the $(2, n)$ -threshold access structure. Let $S = \{n\}$, and let $\mathcal{S}$ be some secret sharing scheme realizing Γ . Construction 2 has message complexity $(1) + (n - 2) = n - 1$.

Now, recall that the lower bound in Theorem 1 requires that for every minimally qualified set Q that includes at least one snitch, at least one faker in Q receives at least one message. Observe that in our threshold access structure, the single snitch together with each individual faker defines a minimally qualified set; it follows that each faker must receive at least one message. Since there are $n - 1$ fakers, the message complexity of Construction 2 is optimal.

Generally, Construction 2 does best when the largest unqualified set containing the snitches is big, and the smallest qualified set contained within the fakers is small.

Example 6 (One-Message Faking). Let $n > 2$, and let $\Gamma = \{\{1, 2\}, \{2, \dots, n\}\} \subseteq 2^{[n]}$ denote the access structure. For some k s.t. $2 \leq k \leq n$, let $F = \{1, \dots, k\}$, and $S = \{k + 1, \dots, n\}$. Let $\mathcal{S}$ be some secret sharing scheme realizing Γ .

Construction 2 would choose party 2 as the leader. It has message complexity $1 + 0 = 2$, since during step 1 party 1 sends a single message to party 2, and during step 2 party 2 only needs to update its own share locally (and thus sends no messages). This matches the lower bound of Theorem 1.

5 Achieving FRI with Non-Interactive Faking

We propose constructions achieving FRI with non-interactive faking. First we describe a construction from PRGs with exponential share size in the number of parties. We then show how share sizes may be reduced assuming virtual black-box (VBB) obfuscation. Finally, we discuss the barriers we encountered when attempting to transfer our VBB construction to the indistinguishability obfuscation setting.

5.1 A First Construction from PRGs

We provide a simple construction that achieves Fake-Real Indistinguishability (FRI) by extending an arbitrary existing secret sharing scheme using a PRG. A caveat of our construction is that it is not efficient for an arbitrary number n of parties due to the shares growing exponentially with n .

In this construction, the dealer generates independent shared randomness for each of the possible sets of fakers.

The dealer only shares the randomness for each of these sets to the parties that form the set of fakers. During the faking algorithm, the fakers use a PRG to transform the randomness provided by the dealer to produce two random strings r_1 and r_2 . The randomness r_1 substitutes the randomness provided by the dealer in the new share, and r_2 is used as randomness to produce a fresh sharing of the fake message. The fakers forward this fresh share with the randomness derived from the PRG to the adversary. We assume a PRG taking a seed of length λ which outputting strings of length $\lambda + \ell_{share}$ where ℓ_{share} is the length of the randomness used by Share,

$$\text{PRG} : \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda \times \{0, 1\}^{\ell_{share}}.$$

Share(m)	Fake(F, m^{fake}, R_i)
1: $(s_i)_{i \in [n]} \leftarrow \text{S.Share}(m)$	1: $R'_i \leftarrow R_i$
2: for $F \in 2^{[n]}$:	2: $r_1 r_2 \leftarrow \text{PRG}(R_i[F])$
3: $r_F \leftarrow \{0, 1\}^\lambda$	3: $R'_i[F] \leftarrow r_1$
4: for $i \in F$:	4: $(s'_i)_{i \in [n]} \leftarrow \text{S.Share}(m^{\text{fake}}; r_2)$
5: $R_i[F] \leftarrow r_F$	5: return $(s'_i, R'_i)_{i \in [n]}$
6: return $(s_i, R_i)_{i \in [n]}$	

Fig. 1. Generic Deniable Secret Sharing with FRI with shares exponential in n .

We will argue that the scheme has FRI and Privacy.

Theorem 7. *The scheme described in Figure 1 has Fake-Real Indistinguishability.*

Proof. We proceed by reduction to the security of the PRG. Fix a partition P_0, P_1 of $[n]$ and messages m_0, m_1 s.t. $|m_0| = |m_1|$. Assume that there exists a PPT adversary $\mathcal{A}$ that wins the FRI-game with advantage ε , i.e.,

$$\Pr \left[\mathcal{A} \left(\begin{array}{c} m_0, m_1, \\ P_0, \\ P_1, \\ \{sh_i\}_{i \in [n]} \end{array} \right) = b \mid \begin{array}{l} b \leftarrow \{0, 1\}; \\ (sh_1^{\text{real}}, \dots, sh_n^{\text{real}}) \leftarrow \text{Share}(m_b); \\ \{sh_i\}_{i \in P_{1-b}} \\ \quad \leftarrow \text{fake}(\{sh_i^{\text{real}}\}_{i \in P_{1-b}}, m_{1-b}); \\ \text{For } i \in P_b : sh_i := sh_i^{\text{real}} \end{array} \right] = \frac{1}{2} + \varepsilon.$$

Hybrid 0: We start with adversary $\mathcal{A}$ in the FRI game described as above.

Hybrid 1: Replace calls to $\text{PRG}(r_{P_{1-b}})$ during **Fake** by a random function R .

These hybrids are indistinguishable due to the pseudorandomness of the PRG, as $r_{P_{1-b}}$ is a uniformly random seed. If the adversary has advantage ε in hybrid 0, it must have at least advantage $\varepsilon - \text{negl}(\lambda)$ in this hybrid. As otherwise $\mathcal{A}$ may be used to construct an adversary with non-negligible advantage in the pseudorandomness game. We may construct such an adversary by guessing ‘real’ in the PRG game when the adversary wins the FRI game, and ‘random’ in the PRG game when the adversary loses.

At this point the shares of P_0 and P_1 are identically distributed, as both sets of shares are generated by the underlying sharing algorithm and the set of randomnesses R_i held by each party are all uniformly random. Thus the adversary cannot have any advantage, implying that ε must be negligible. $\square$

We argue that our scheme has Γ -privacy if the underlying scheme is Γ -private, this clearly follows as the fake shares and additional randomness may be simulated with no knowledge of the underlying real message.

5.2 Replacing Exponential Randomness with Obfuscated Programs

Our previous construction required the dealer to prepare randomness exponential in the number of parties, in the form of randomness r_F for each set $F \in 2^{[n]}$. When extending this approach the first natural question to ask is if these random values for faking can be replaced with pseudorandom values. Doing so quickly poses a few challenges:

1. how can we efficiently provide each party exactly with the randomnesses for the sets that it is a member of?
2. how can fakers replace the randomness they have used during the faking process?

We present an efficient construction resolving these challenges assuming virtual black-box (VBB) obfuscation [BGI⁺01] and symmetric primitives. We prove our construction secure in the oracle access model, where our adversary is restricted to oracle access to our obfuscated programs. Note that VBB obfuscation is known to be impossible as there exist functions which are inherently unobfuscatable [BGI⁺01]; therefore, the scheme presented in this section should be seen as a conceptual stepping stone for future schemes using weaker forms of obfuscation. Later, in section 5.3, we will discuss the barriers encountered when trying to realize the same scheme from indistinguishability obfuscation.

We construct our DSS scheme by giving parties access to a program **Rand** which generates their faking randomness. To prevent the adversary learning the internal state of this program we assume the dealer only provides a VBB obfuscation of the program. This allows us to treat **Rand** as an oracle going forward.

When a party i wants to fake along with a set F , the program Rand must provide a suitable randomness $r = \text{PRF}_h(F)$ derived pseudorandomly from on the set F and a key h . To achieve FRI no party outside F should be allowed to obtain r . We enforce this by giving each party an authenticated ciphertext $c_i = \text{Enc}_K(i)$ of its index i . The program Rand then only outputs $r = \text{PRF}_h(F)$ if given an input c where $(i, h) = \text{Dec}_K(c)$ for $i \in F$. To prevent the adversary from breaking FRI by querying rand on the same input as the fakers did, we allow them to “ratchet” the key contained in their ciphertexts by using the oracle. That is, we add a second output to the PRF $r, h' = \text{PRF}_h(F)$ and encrypt $c'_i \leftarrow \text{Enc}_K((i, h'))$ to the fakers allowing them to present c'_i to the adversary in place of c_i .

We present our formal constructions in Figure 2, where Share and Fake are algorithms and Rand_K is an oracle algorithm parameterised by a key K picked during sharing. Our construction uses a deterministic authenticated encryption scheme [RS06] and a pseudorandom generator,

$$\text{PRF} : \{0, 1\}^{\ell_{\text{seed}}} \times 2^{[n]} \rightarrow \{0, 1\}^{\ell_{\text{share}}} \times \{0, 1\}^{\ell_{\text{seed}}}.$$

We let $\ell_{\text{seed}} = 2\lambda$ to ensure that no PRF evaluation coincides with h chosen during sharing.

$\text{Share}(m)$	$\text{Fake}^{\text{Rand}_K(\cdot, \cdot)}((s_i, c_i), F)$	$\text{Rand}_K(c, F)$
1: $(s_i)_{i \in [n]} \leftarrow \text{S.Share}(m)$	1: $r', c'_i \leftarrow \text{Rand}_K(c_i, F)$	1: $p = (i, h) \leftarrow \text{Dec}_K(c)$
2: $h \leftarrow \{0, 1\}^{\ell_{\text{seed}}}$	2: $(s'_i)_{i \in [n]} \leftarrow \text{S.Share}(m; r')$	2: if $p = \perp \vee i \notin F$:
3: $K \leftarrow \mathcal{K}$	3: return (s'_i, c'_i)	3: return $\perp$
4: for $i \in [n]$:		4: $r', h' \leftarrow \text{PRF}_h(F)$
5: $c_i \leftarrow \text{Enc}_K((i, h))$		5: $c' \leftarrow \text{Enc}_K((i, h'))$
6: return $(s_i, c_i)_{i \in [n]}$		6: return (r', c')

Fig. 2. DSS with FRI given obfuscated randomness.

Theorem 8. *The construction described in figure 2 has (P_0, P_1) -Fake-Real Indistinguishability (Definition 5) in the oracle access model for all partitions P_0, P_1 of $[n]$ assuming a deterministic authenticated encryption scheme (Enc, Dec) and that PRF is pseudorandom.*

Proof. Let $\mathcal{A}$ be a PPT adversary with advantage ϵ in the FRI-game making at most q oracle queries to Rand_K . We proceed through a series of hybrids, each describing modifications to the previous.

Hybrid 0: We start with the FRI game (Definition 5). Recall, the challenger samples a bit $b \leftarrow \{0, 1\}$ and the adversary sees

$$\{(s_i^{\text{real}}, c_i)\}_{i \in P_b} \cup \{(s_i^{\text{fake}}, c_i)\}_{i \in P_{1-b}}$$

where $\{s_i^{real}\}_{i \in [n]} \leftarrow \text{S.Share}(m_b)$ and $\{s_i^{fake}\} \leftarrow \text{S.Share}(m_{1-b}; r')$ for $(r', c'_i) \leftarrow \text{Rand}_K(c_i, P_{1-b})$. At this point $c_i = \text{Enc}_K(i, h_b)$ for $i \in P_{b'}$ where $h_b = h$ and $h_{1-b} = \text{PRF}_{h_b}(P_{1-b})$.

Hybrid 1: Replace all ciphertexts with random strings maintaining a map from plaintext to ciphertexts. For calls to decrypt in **Rand**, check if the ciphertext was previously stored, retrieving the corresponding plaintext if this is the case. If the ciphertext was not previously stored, return $\perp$ directly.

This hybrid is indistinguishable from the previous hybrid due to the security of the deterministic authenticated encryption scheme.

Hybrid 2: We replace the PRF calls with random functions, through a series of $q + 2$ hybrids.

- First replace each call to $\text{PRF}_{h_b}(F)$ with a random function R_{h_b} . This is indistinguishable due to the pseudorandomness of the PRF.
- Now, h_{1-b} (now an output of R_{h_b}) is uniformly random allowing us to replace all calls $\text{PRF}_{h_{1-b}}$ by a random function $R_{h_{1-b}}$.
- Successively for each oracle query $\text{Rand}_k(c_j, F_j)$ for $j \in [q]$, if $(i_j, h_j) = \text{Dec}_K(c_j)$ and $i_j \in F_j$ then replace PRF_{h_j} by random function R_{h_j} . If c_j successfully decrypts it must previously have been encrypted by the oracle in a previous query, this implies that h_j was chosen uniformly at random making the change indistinguishable by pseudorandomness of the PRF.

After the final hybrid, all calls to the PRF have been replaced by truly random functions.

Hybrid 3: Maintain a list of all h' where $(\cdot, h') \leftarrow R_{h''}(F)$ for some call to **Rand**. Abort if any input results in output h' equal h_b to sampled during sharing.

As the adversary is polynomially bounded making at most q queries and R_h is random each h' is uniform in $\{0, 1\}^{\ell_{seed}}$, thus by a union-bound the probability that any output is equal to h_b is negligible. Therefore this hybrid is indistinguishable from the previous.

Hybrid 4: At this point we know the **Rand** oracle will never evaluate R_{h_o} on an input P_{1-b} as no ciphertext of the form (i, h_o) s.t. $i \in P_{1-b}$ may ever successfully decrypt. This could only occur if (i, h_o) had previously been encrypted, which would imply $(\cdot, h_o) \leftarrow R(h', P')$ for some h', P' , which we have ruled out by hybrid 3.

Hybrid 5: In this hybrid we choose $\hat{r}, \hat{h}$ independently of R_{h_o} and define

$$\{(s_i^{fake}, c'_i)\}_{i \in P_{1-b}}$$

as $\{s_i^{fake}\} \leftarrow \text{S.Share}(m_{1-b}; \hat{r})$ and for $i \in P_{1-b}$ let c'_i be the encryption of $(i, \hat{h})$. This hybrid is identically distributed to the previous as R is never queried on (h_o, P_{1-b}) .

After hybrid 4, we may conclude our proof as the distribution seen by the adversary is independent of b , as both sets P_0, P_1 are given shares computed from a uniformly sampled randomness, and ideal encryptions of a random seeds h_b and $\hat{h}$ respectively. $\square$

Privacy of this scheme follows by the same reasoning as our inefficient scheme, fake shares and ciphertexts may be perfectly simulated with no knowledge of the underlying real message. Thus an adversary breaking Γ -privacy of the DSS scheme may be reduced directly to an adversary breaking Γ -privacy of the underlying scheme.

5.3 Challenges when moving to indistinguishability obfuscation

When attempting to transfer our construction to indistinguishability obfuscation we encountered a series of barriers. Unlike VBB, $i\mathcal{O}$ only guarantees the indistinguishability of obfuscated circuits for circuits of the same size with identical functionality [BGI⁺01, SW14]. This weakening is sufficient for $i\mathcal{O}$ to be believed to be realizable from well-founded assumptions [JLS21].

In our VBB construction the key h stored in the ciphertexts effectively encoded the entire history of faking applied to a particular share, through a chain of PRF evaluations. The history we consider is the sequence of faker sets $F_1, \dots, F_k$ used to produce a ciphertext c_k , that is $c_2 = \text{Rand}(c_1, F_1)$, $c_3 = \text{Rand}(c_2, F_2)$, $\dots$, $c_k = \text{Rand}(c_{k-1}, F_k)$. Security relied on the PRF evaluations in this chain not having collided with the initial key, which would result in a ciphertext that would allow evaluating the PRF with the starting key h_b on the set of fakers P_{1-b} . Doing so, the adversary could directly identify the fakers.

When moving to $i\mathcal{O}$ calls to the PRF have to be treated differently, as we cannot easily exclude that there exist cycles when evaluating the PRF in this way. Encoding the entire history $F_1, \dots, F_k$ is also a non-starter, as the adversary may arbitrarily extend this history by applying Rand to the ciphertexts it receives. No ciphertext of fixed size can fit an arbitrary polynomially long history, so some histories must eventually be encoded identically. We present alternatives to encoding the entire history and discuss why they each ultimately prove to be insufficient.

- The first natural approach would be encoding a counter in the ciphertexts like the level system in [CPP20], ensuring that each new call to Rand results in new randomness by incrementing the counter. This solution breaks as if the snitches hold ciphertexts with counter 0 and the fakers hold ciphertexts with counter 1, the adversary can observe that the randomnesses coincide after faking once on behalf of the snitches.
- The counter approach fails because the adversary can detect which set faked by making the values in the ciphertexts coincide if a particular set faked previously. To avoid this one could try to ensure that the state of faker ciphertexts can never coincide with that of the snitches. One way to achieve this is to maintain a list ℓ of n integers in each ciphertext where $\ell[i]$ takes the value of the counter when a p_i was first excluded for a set of fakers. If P_0 fakes, then the $\ell[i] = 0$ for all $i \in P_1$. If the same share is then faked for a smaller set of fakers $P'_0 \subset P_0$ then $\ell[j] = 1$ for all $j \in P_0 \setminus P'_0$ and so on.

Unfortunately, this does not work either as for a set of fake shares from P_b faking for sets $[n]$ and P_b would result in the same state, while this is not true for a set of real shares.

It is worth highlighting that the failure of the approaches described above by no means constitute an impossibility result, simply that our construction does not readily transfer to the iO setting. We leave finding a suitable encoding of the history, or alternative approach to achieving FRI without VBB to future work.

References

- BGG⁺20. Fabrice Benhamouda, Craig Gentry, Sergey Gorbunov, Shai Halevi, Hugo Krawczyk, Chengyu Lin, Tal Rabin, and Leonid Reyzin. Can a public blockchain keep a secret? In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 260–290. Springer, Cham, November 2020.
- BGI⁺01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001.
- BHK⁺22. Fabrice Benhamouda, Shai Halevi, Hugo Krawczyk, Alex Miao, and Tal Rabin. Threshold cryptography as a service (in the multiserver and YOSO models). In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 323–336. ACM Press, November 2022.
- CDK⁺25. Ran Canetti, Ivan Damgård, Sebastian Kolby, Divya Ravi, and Sophia Yakoubov. Deniable secret sharing. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part II*, volume 16269 of *LNCS*, pages 399–427. Springer, Cham, December 2025.
- CPP20. Ran Canetti, Sunoo Park, and Oxana Poburinnaya. Fully deniable interactive encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 807–835. Springer, Cham, August 2020.
- GHK⁺21. Craig Gentry, Shai Halevi, Hugo Krawczyk, Bernardo Magri, Jesper Buus Nielsen, Tal Rabin, and Sophia Yakoubov. YOSO: You only speak once - secure MPC with stateless ephemeral roles. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 64–93, Virtual Event, August 2021. Springer, Cham.
- GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.
- JLS21. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021.
- RS06. Phillip Rogaway and Thomas Shrimpton. Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. Cryptology ePrint Archive, Report 2006/221, 2006.
- SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

A A Stronger Lower Bound for Linear Secret Sharing

Recall that Theorem 1 puts forward a lower bound on the number of messages that an FH faking protocol must send. In particular, certain unqualified subsets $F' \subset F$ must receive incoming messages during faking. It does not directly imply a lower bound on information leaked about the secret, as the messages do not have to leak information about the secret. Here we prove a somewhat stronger version of the theorem in case the underlying secret sharing scheme is linear, stating that these (same) F' -s actually must learn the entire secret. This in particular implies that they must receive an incoming message during the faking protocol.

Theorem 9. *Let Γ be an access structure, and let $\mathcal{S}_\Gamma = (\text{Share}, \text{Rec})$ be a linear secret sharing scheme realizing Γ . Let fake be a faking function such that $(\text{Share}, \text{Rec}, \text{fake})$ is a perfect DSS that satisfies fake hiding (Definition 6).*

Let F be a qualified set of fakers such that $S = \overline{F}$ is an unqualified set (of snitches).

Let Π be a protocol realizing fake — possibly using correlated randomness — for our set of fakers F .

Let $F' \subseteq F$ and let $S' \subseteq S$ be non-empty sets such that $F' \cup S'$ is a minterm. Then, the secret can be reconstructed from the view of F' in Π .

Proof. Assume the faking protocol transforms any sharing (m) into a sharing (f) , such that $(f)_S = (m)_S$. Thus, by linearity, $(f) - (m)$ is a valid sharing of $f - m$. Furthermore, after producing the fake shares (f) , $\overline{sh} = (f) - (m)$ can be locally generated by each party i using its faked share, and its original share. In this sharing, all shares of S are 0's. consider a minterm $F' \cup S'$ where $F' \subseteq F, S' \subseteq S$ where both F', S' are non-empty. By correctness of $\mathcal{S}_\Gamma$, the Rec correctly recovers the secret from the shares of $F' \cup S'$. As the shares of S' in $\overline{sh}$ are 0, F' know all the shares in $\overline{sh}_{F' \cup S'}$, and can correctly recover the secret $f - m$ by themselves, and derive m from it (as they know f).