Deniable Secret Sharing

Ran Canetti[1], Ivan Damgård[2], Sebastian Kolby[2], Divya Ravi[3], and Sophia Yakoubov[2]

[1] Boston University, USA; canetti@bu.edu
[2] Aarhus University, Denmark; {ivan, sk, sophia.yakoubov}@cs.au.dk
[3] University of Amsterdam, Netherlands; d.ravi@uva.nl

Abstract. We introduce *deniable secret sharing* (DSS), which, analogously to deniable encryption, enables shareholders to produce fake shares that are consistent with a target "fake message", regardless of the original secret. In contrast to deniable encryption, in a DSS scheme an adversary sees multiple shares, some of which might be real, and some fake. This makes DSS a more difficult task, especially in situations where the fake shares need to be generated by individual shareholders, without coordination with other shareholders.

We define several desirable properties for DSS, and show both positive and negative results for each. The strongest property is *fake hiding*, which is a natural analogy of deniability for encryption: given a complete set of shares, an adversary cannot determine whether any shares are fake. We show a construction based on Shamir secret sharing that achieves fake hiding as long as (1) the fakers are qualified (number t or more), and (2) the set of *real* shares which the adversary sees is unqualified. Next we show a construction based on indistinguishability obfuscation that relaxes condition (1) and achieves fake hiding even when the fakers are *un*qualified (as long as they comprise more than half of the shareholders).

We also extend the first construction to provide the weaker property of *faker anonymity* for all thresholds. (Faker anonymity requires that given some real shares and some fake shares, an adversary should not be able to tell which are fake, even if it can tell that some fake shares are present.) All of these constructions require the fakers to coordinate in order to produce fake shares.

On the negative side, we first show that fake hiding is unachievable when the fakers are a minority, even if the fakers coordinate. Further, if the fakers do *not* coordinate, then even faker anonymity is unachievable as soon as $t < n$ (namely the reconstruction threshold is smaller than the number of parties), if faking is not unanimous. (If faking *is* unanimous, we show a construction based on indistinguishability obfuscation.)

1 Introduction

A recent line of work [4,16,5] studies how to efficiently outsource the storage of (and computation on) secrets to a large-scale network in the presence of an adaptive adversary. We can give shares of the secret to all the parties in the network, but that can have prohibitive communication cost. The trick is to only

give shares of the secret to *some* of the parties, while hiding their identities to protect them from adaptive corruption. However, what if instead of trying to find — and corrupt — relevant parties, our adversary posts a bounty, and waits for parties who have information to come forward and sell it?

Motivated by the prospect of countering such an attack, we introduce a new primitive which we call *deniable secret sharing* (DSS). In addition to the Share and Rec algorithms, deniable secret sharing is equipped with a Fake algorithm, which enables a shareholder to turn their real secret share into a fake share which *looks* like a real share, but doesn't contribute to the recovery of the secret.

Indeed, if such a deniable secret sharing scheme is used, the adversary will have nothing to gain from offering a bounty for shares, because parties are likely to take her money and give her fake shares. Similarly, if we consider a scenario where an adversary is using a stick rather than a carrot — is demanding at gun-point that a shareholder give up her share — DSS enables the shareholder to appease the adversary without betraying the dealer's trust. In the rest of this paper, we call shareholders who hand over their real shares to the adversary *snitches*, and shareholders who give the adversary fake shares *fakers*.

1.1 Related Work

Deniable secret sharing is analogous to *receiver-deniable encryption* [9], where a message receiver is able to produce a fake secret key that decrypts a given ciphertext to a target fake message. However, in a deniable secret sharing scheme there are multiple shareholders, not all of whom will fake their shares. This introduces additional consistency challenges.

Deniable primitives have proven very useful against adversaries which seek to coerce parties into revealing their private state and inputs after participating in a protocol [10]. Such attacks are a significant concern for functionalities like voting [3,1] where individuals could be forced to vote in a particular way if the protocol allowed obtaining a *receipt* when voting. Protocols defending against such attacks are known as *receipt-free* or *incoercible*. This setting distinguishes itself from the closely related notion of adaptive security, as coerced parties must equivocate their state using a local faking algorithm, rather than relying on a global simulator.

Canetti and Gennaro [10] showed that incoercible MPC protocols exist given trapdoor permutations as long as fewer than half the parties are coerced. Canetti and Poburinnaya [13] later strengthened this to the setting where all parties are coerced, providing an incoercible protocol as long as all parties follow the prescribed faking procedure. Existing incoercible protocols are built using powerful primitives such as fully deniable encryption [12] and indistinguishability obfuscation [2,20]. For DSS we make stronger requirements, considering an adversary that always receives some share from each party and demanding security hold even in the presence of snitches who deviate from the faking algorithm (by giving the adversary their real share).

Anamorphic encryption [19] achieves deniability by alternative means. Rather than constructing novel schemes with the express goal of deniability, anamor-

phic encryption aims to subvert existing cryptosystems, introducing a *double key* which allows hiding an additional message within a ciphertext. Plausible deniability comes from sender and recipient claiming that they were legitimately using the unmodified scheme, where no double key ever existed. This argument does not transfer to primitives such as deniable secret sharing, as the adversary would reasonably expect key material like the double key to exist and demand it be handed over.

Secret sharing with snitching [14] and *traceable secret sharing* [18] both aim to incentivize shareholders not to participate in premature reconstruction of the secret by ensuring that such participation gives the adversary evidence against them, which can be used to implicate them as premature reconstructors. We take the other extreme: rather than making participation proveable, deniable secret sharing aims to make it *undetectable*, even to the adversary gathering the shares. In some scenarios, this is more powerful, since it lifts all incentive to cooperate with an adversary's demands for premature reconstruction.

1.2 Technical Overview

The contributions of this paper are three-fold: we present new definitions, demonstrate lower bounds that preclude schemes that meet these definitions in some contexts, and build constructions that meet the definitions in other contexts.

1.2.1 Definitions A deniable secret sharing (DSS) scheme consists of three algorithms: Share and Rec, as in any secret sharing scheme, and an additional algorithm Fake, which shareholders can run in order to generate a fake share (upon inputting their real share).

Notions of Faking. We consider several notions of faking. The first is *dealer specified faking*, where the dealer has a fake message in mind as a fall-back plan when she creates the shares. She encodes the fake message within the shares, so that when the shareholders apply the Fake algorithm to their shares, they obtain shares of the fake message (without necessarily knowing what that fake message is). We immediately rule dealer specified faking out (Section 1.2.2), and instead focus on the following notions:

- In DSS with *shareholder specified faking*, when running the Fake algorithm, shareholders specify a *fake message* to which they want the resulting fake shares to reconstruct.
- In DSS with *random faking*, no-one chooses the fake message; rather, it is implicitly chosen from a fixed distribution via the randomness of the faking process.
- In DSS with *denial-of-service faking*, there is no fake message at all, and instead reconstruction simply fails if sufficiently many fake shares are present.

On an orthogonal axis, we consider whether shareholders coordinate while faking their shares (that is, whether Fake takes in a single share, or a set of shares).

Properties. We introduce three new properties for DSS.

- A DSS scheme has *fake-real indistinguishability* (FRI) if, given two sets of (possibly fake) shares — of m_0 and of m_1 — it is hard to determine which set of shares is real and which set is fake (and thus what the real message was).
- A DSS scheme has *faker-anonymity* (FA) if, given a set of shares (some of which are fake), it is hard to determine which of them are fake (even if the real message is known).
- A DSS scheme is *fake-hiding* (FH) if a set of shares (some of which are fake) looks exactly like a fresh sharing of the fake message, and the adversary cannot even tell that any faking took place.

As a minimal condition for shareholder specified faking (whether coordinated or uncoordinated) and random faking, we require FRI for the case where one of the two sets is empty, and the other contains all n shares (where n is the number of shareholders). That is, we require that when all shareholders fake their shares, the resulting set of fake shares be indistinguishable from a fresh sharing of the fake message[4]. On the other hand, in the context of denial of service faking, this is unnecessary (since if reconstruction fails, this is naturally distinguishable from a correct reconstruction using real shares).

Intuitively, the standard notion of privacy and our new notion of faker anonymity are orthogonal: privacy protects the message, and faker anonymity protects the fakers. Fake-real indistinguishability is a fall-back notion for both: it creates some uncertainty about the real message *if* the adversary does not know who the fakers are, and it creates some uncertainty about the fakers *if* the adversary does not know the real message. Fake-hiding is a more powerful property that implies both privacy and faker anonymity; interestingly, the relationship of fake-hiding and fake-real indistinguishability is more complex (see Section 2.4.4).

We do not explicitly consider the additional presence of *corrupt* parties who always give the adversary their real share. Corrupt parties differ from snitches, since the adversary does not necessarily know that a snitch's share is real; however, the adversary would be certain that a corrupt party's share is real. FA is still meaningful in the presence of such corrupt parties; in fact, our constructions still attain FA if the snitches are corrupt. However, exploring other flavors of DSS in the presence of corrupt parties is an interesting avenue for future work.

1.2.2 Dealer Specified Faking We can immediately rule out the notion of dealer specified faking. This is because, given a set of n shares, repeatedly applying the Fake algorithm allows the extraction of the fallback fake message, as well as any additional fake messages (if the dealer provided more than one layer of fallbacks). Given knowledge of the number of fallbacks, an adversary can determine whether a set of shares is real by extracting — and counting —

[4] FRI and FH are equivalent for the case where either all of the shares or none of them are fake.

the fallbacks. A way around this is to create uncertainty about the number of fallbacks, but the only way for this to offer reasonable security is to have an exponential upper bound on this number, which precludes an efficient construction, since the sum of the share sizes must scale linearly with the number of fallbacks encoded in the shares.

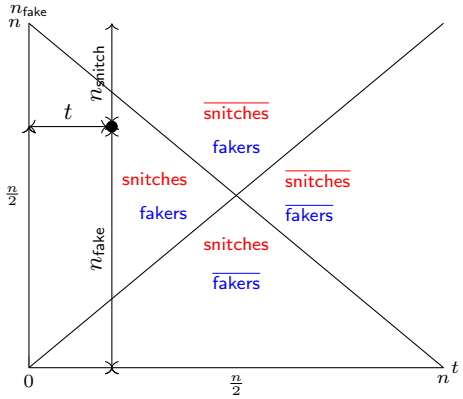
1.2.3 Shareholder Specified Faking In the setting of shareholder specified faking, we have two possibilities: the shareholders either run the Fake algorithm locally, or they coordinate.

Uncoordinated Shareholder Faking. In Section 5.5, we show a DSS scheme with uncoordinated shareholder faking that achieves the minimal condition: FRI when either everyone is a faker or everyone is a snitch. However, if there is at least one faker and at least one snitch, the picture is bleak. If $t < n$, nothing can be achieved when snitches are qualified; fakers run Fake locally, without knowledge of whether the snitches are qualified or not, so they cannot return their real shares without violating privacy. Then, an adversary can identify fake shares by attempting reconstruction with different subsets of the shares it holds. In Section 3, we show that even if snitches are unqualified, we cannot have any notion of deniability (FRI, FA, FH) when the shareholders do not act in unison (that is, if at least one faker and one snitch is present). We move to the coordinated setting for a more interesting picture.

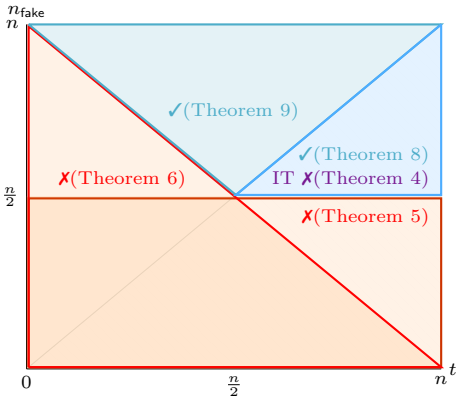
Coordinated Shareholder Faking. Figure 1 summarizes the thresholds t and numbers n_{fake} of fakers for which FRI, FA and FH are achievable. We have two constructions in this setting. The first (Theorem 9) is an information-theoretic construction based on Shamir secret sharing. The Share and Rec algorithms are exactly the ones used by Shamir secret sharing. In order to create fake shares, the fakers do the following:

- If the snitches are unqualified:
  - If the fakers are qualified: they interpolate their Shamir shares to recover the snitches' shares, and return points on a new polynomial that contains the snitches' shares and intersects the y-axis at the target fake message. *Here we have all of FRI, FA and FH.*
  - Otherwise, if the fakers are unqualified: they return random points. *Here we have FRI and FA, but not FH, since the resulting set of snitch and faker shares will not lie on a polynomial of degree $t-1$.*
- Otherwise, if the snitches are qualified (that is, $|\mathsf{snitches}| \geq t$), the fakers known that the adversary will learn the real message anyways, so they return their original (real) shares. *We cannot hope for FH here. However, we have the remaining two properties: FRI and FA.*

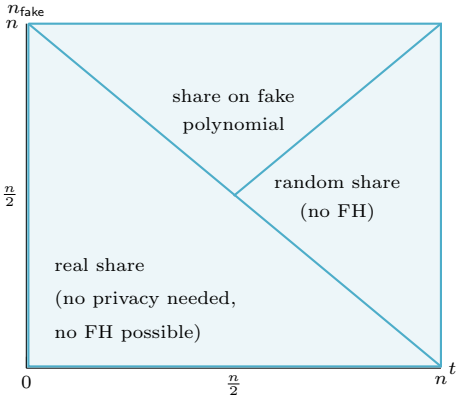
The Fake algorithm from this construction is summarized in Figure 1c. It provides almost the best possible guarantees; the only gap is the lack of FH when both the fakers and the snitches are unqualified. In Theorem 4, we show



(a) Guide to where fakers and snitches are qualified. snitches means the snitches are qualified; $\overline{\text{snitches}}$ means that they are not. fakers means the fakers are qualified; $\overline{\text{fakers}}$ means that they are not. Fake-hiding and faker anonymity are both unachievable when the snitches are qualified.



(b) Landscape of feasibility of fake-hiding.



(c) Both fake-real indistinguishability (FRI) and faker anonymity (FA) are possible for all thresholds (Theorem 9). This figure maps what the Fake algorithm from Theorem 9 outputs.

Fig. 1: Landscape of feasibility for shareholder specified faking with coordination. On each graph, t (the number of shares necessary for reconstruction) is on the x-axis, and n_{fake} (the number of faker shares) is on the y-axis. Figure 1a summarizes where on the graphs each of the fakers and snitches are qualified. Figure 1b summarizes where fake-hiding is feasible. Fake-real indistinguishability and faker anonymity are feasible for all thresholds; Figure 1c summarizes how the construction from Theorem 9 achieves them.

that when both fakers and snitches are unqualified, no *information-theoretic* construction can achieve FH. In Theorem 5, we show that when there are $\frac{n}{2}$ or fewer fakers, even a computational construction cannot achieve FH.

We complete the picture with a construction that uses obfuscation and achieves FH as long as more than half of the shareholders are fakers, and the snitches are not qualified (Theorem 8).

Optimally Fake-Hiding Coordinated Shareholder DSS from Obfuscation. In our optimally fake-hiding construction, there is a common reference string in the form of an obfuscated program which holds a secret authenticated encryption key. Each secret share is an encryption of the message and the shareholder identity, decryptable only by the program. In order to reconstruct or to generate fake shares, the program takes as input a set of shares. In order to reconstruct, the program checks that the set has t or more valid ciphertexts. If there are enough, the program decrypts the ciphertexts. Some of these ciphertexts might decrypt to a fake message, and a counter that indicates how "new" the fake message is; the program returns either the newest fake message, or, if all of the messages are real, it returns the real message.

In order to generate fake shares, the program checks that the set of shares it is given contains more than $\max(\frac{n}{2}, n-t)$ valid ciphertexts. If it does, the program returns encryptions of the fake message — together with an incremented counter — as the fake shares. Requiring more than half of the shares in order to fake ensures that sequential applications of faking, on potentially different subsets of shares, will always result in a strict ordering of fake messages.

1.2.4 Random and Denial of Service Faking

Random Faking. When the fakers don't coordinate and threshold $t = n$, additive secret sharing — where to fake, a shareholder chooses a fresh random share — gets us all of privacy, FRI, FA and FH. For $t < n - 1$, the impossibility of uncoordinated random faking can be shown in much the same way as the impossibility of uncoordinated shareholder faking (Appendix D.2).

When shareholders coordinate, we can modify the Shamir construction for coordinated shareholder faking to get a construction with random faking that gets privacy, FRI and FA for all thresholds, and FH for all thresholds where FH is possible. When the snitches are qualified, as before, the fakers return their real shares. When the snitches are unqualified, whether the fakers are themselves qualified or not, they choose a random value ρ and a random degree-$(t-1)$ polynomial f such that $f(i) = 0$ for all snitches i, and $f(0) = \rho$. Each faker i computes their fake share by adding $f(i)$ to their real share.

Denial of Service (DoS) Faking. In the context of DoS faking, our goal is to get reconstruction to fail if sufficiently many shares are fake. Strong properties like FH are not relevant here, but FRI and FA are. In settings where random faking is possible, we can bootstrap a random faking construction to get a DoS faking

construction: First, map messages to a subspace of a random distribution dist in such a way that a randomly chosen element of dist will only be a valid message with negligible probability. Then, use a random faking construction with FRI and/or FA for distribution dist.

For $t < n - 1$, random faking is impossible unless the fakers coordinate. However, if the snitches are unqualified, we can get DoS faking even if the fakers do *not* coordinate. This can again be built from Shamir sharing: each faker simply takes a random value as its fake share. All of the (snitch and faker) shares look like independent random values, giving us both FRI and FA.

1.3 Future Work and Open Problems

It would be interesting to see what changes when some of the shareholders are *corrupt*. Such corrupt shareholders would naturally give the adversary their true shares, but the adversary would additionally have certainty that the corrupt shareholders' shares are, in fact, real. Another avenue for future work is additionally considering a dealer's deniability (rather than just the shareholders').

2 Definitions

In this section, we formally define deniable secret sharing.

2.1 Notation and Syntax

Parameters. Our schemes are parametrized by the following:

n**:** Number of shares.
n_{fake}**:** Number of fakers.
t**:** The number of shares necessary for reconstruction. Any fewer shares should reveal nothing about the secret.

Throughout the paper, we assume that each of the n parties delivers one share to the coercing or bribing adversary (and is thus either a faker or a snitch); so, the number of snitches is $n - n_{\mathsf{fake}}$.

Types of Faking. Having ruled out dealer-specified faking (Section 1.2.2), we consider three flavors of faking, differing based on where the fake message comes from:

Shareholder-specified faking, where each faker alters her share with a specific target fake message in mind.
Random faking, where there is no target fake message; rather, the fake message is random (from some distribution).
Denial-of-service faking, where there is no fake message; rather, the fakers aim to prevent the coercer from learning the original message, without intending to fool him into thinking that everyone snitched.

Coordination. Another parameter is whether the fakers communicate with one another or not. Each of the above flavors of faking by default requires faking to be a local process; however, each can be modified to allow the fakers to coordinate. In some cases, this coordination can even be independent of the fakers' shares, and can thus be done *before* sharing.

Algorithms. A deniable secret sharing scheme comprises the following algorithms, where elements present only for shareholder-specified faking appear highlighted in *grey*.

$\mathsf{Share}(m) \to (s_1, \ldots, s_n)$ is the secret sharing algorithm.
$\mathsf{Rec}(\{s_i\}_{i \in \mathcal{Q}}) \to m$ takes in a set of shares belonging to parties $\mathcal{Q}$, and outputs a secret (as long as $|\mathcal{Q}| \geq t$).
$\mathsf{Fake}(s_i, m^{fake}) \to s_i^{fake}$ takes in a share, and outputs a fake share.

When faker coordination is allowed, the Fake algorithm takes in a set of shares instead of a single share, as follows:

$\mathsf{Fake}(\{s_i\}_{i \in \mathsf{fakers}}, m^{fake}) \to \{s_i^{fake}\}_{i \in \mathsf{fakers}}$

Matrix Notation. We will use the following pictorial notation: columns of a matrix refer to the shares held by disjoint sets of parties. The first row is all m's if m was shared. (Because the first row refers to the initial sharing, it will always consist of one replicated entry.) Subsequent rows refer to fakings (and, for rows three and on, re-fakings).

As an example, the following matrix describes the case where 0 was shared to $\mathsf{partyset}_0 \cup \mathsf{partyset}_1$, and $\mathsf{partyset}_1$ then faked their shares to 1:

$$\begin{bmatrix} 0 & 0 \\ & 1 \end{bmatrix}$$

2.2 Properties

Informally, we would like the following properties:

Correctness: t or more honest shares enable the recovery of the secret.
Privacy: Fewer than t real shares reveal nothing about secret. This should hold even in the presence of fake shares.
Fake-Real Indistinguishability (FRI): If snitches report their real shares and fakers report fake shares, this should be indistinguishable from the mirror case (where snitches fake and fakers report their real shares).
Faker-Anonymity (FA): The adversary should not be able to tell which shares are fake and which are real.
Fake-Hiding (FH): The adversary should not be able to tell whether anyone faked.

Privacy vs. FRI vs. FA vs. FH. Intuitively, privacy protects the message, FA protects the identities of the fakers, and FRI protects both. Privacy protects the message even if the identities of the fakers are known; FRI does not. However, unlike privacy, FRI can be meaningful even if t or more real shares are available (that is, if the snitches are qualified), as a fallback: even if the adversary is able to recover the real message, FRI guarantees that she won't be convinced that it is the real message and not a fake.

FA and FRI are in some sense orthogonal. FRI does not protect the fakers if the adversary knows (though some external channel) which message is real, while FA does. On the other hand, FA does not protect the message; it makes no guarantee that the adversary cannot be sure that a reconstructed message is real. Finally, FRI is only meaningful when both the set of fakers and the set of snitches could plausibly be either, while FA can hold even in a setting where a majority of fakers is guaranteed.

Fake-hiding is the strongest guarantee; when it is achievable, it implies privacy, FRI *and* FA. However, we only consider FH and FA when the snitches are unqualified (see Remark 1), while FRI is interesting even when the snitches are qualified.

Remark 1. Both FA and FH are uninteresting when the snitches are qualified. This is because, if the fakers do not coordinate (and thus do not know how many fellow fakers they have, and whether the snitches will be qualified or not), FA and FH both contradict privacy: an adversary can reconstruct the secret by looking at the qualified set of snitch shares, so by FA and FH should also reconstruct the same secret by looking at a different set of shares of the same size that includes faker shares. However, if such a set — containing fewer than t real snitch shares — returns the secret as well, this is a contradiction of privacy (since both faking and reconstruction should do the same thing whether or not other, unused snitch shares exist).

On the other hand, if the fakers do coordinate (and thus know that the snitches will be qualified), the Fake algorithm can simply return the original, real shares, trivially achieving both FA and FH.

Figure 2 gives a pictorial summary (in our matrix notation) of the properties (other than correctness) of deniable secret sharing. More details can be found below.

2.2.1 Privacy Informally, a DSS scheme has *privacy* if an unqualified set of real shares reveals nothing about the shared message, even in the presence of additional fake shares.

Definition 1 (t-Privacy). *A scheme is* t-private *if the following holds. For any:*

- *partition of the parties into* fakers *and* $\mathsf{snitches}$ *s.t.* $|\mathsf{snitches}| < t$*;*
- *messages* $m_0, m_1, \{m_i^{fake}\}_{i \in \mathsf{fakers}}$ *s.t.* $|m_0| = |m_1| = |m_i^{fake}|$ *for all* $i \in \mathsf{fakers}$*;*

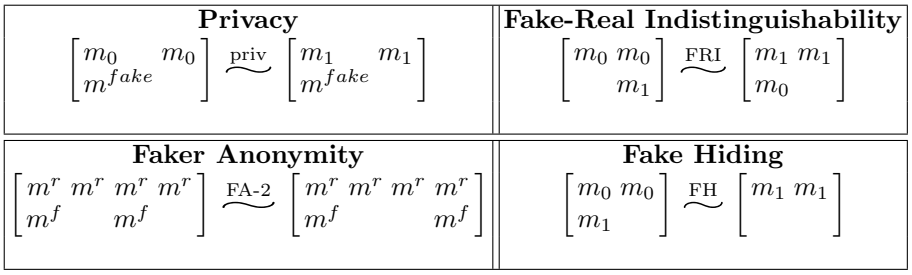


Fig. 2: The Properties of Deniable Secret Sharing

– *PPT adversary* $\mathcal{A}$,

Run the following experiment:

– $b \leftarrow \{0,1\}$;
– $(s_1^{real}, \ldots, s_n^{real}) \leftarrow \mathsf{Share}(m_b)$;
– *For* $i \in \mathsf{fakers}$: $s_i \leftarrow \mathsf{Fake}(s_i^{real}, m_i^{fake})$;
– *For* $i \in \mathsf{snitches}$: $s_i := s_i^{real}$.

The adversary's distinguishing advantage should be negligible:

$$\Pr[\mathcal{A}(m_0, m_1, \{m_i^{fake}\}_{i \in \mathsf{fakers}}, \mathsf{fakers}, \mathsf{snitches}, \{s_i\}_{i \in [n]}) = b] = \frac{1}{2} + \mathsf{negl}$$

In the case of shareholder specified faking, our privacy definition allows the fake shares to be formed with respect to different target messages, which is desirable in practice. One could also consider a weaker notion of privacy, which restricts the fake messages to be the same: $m_i^{fake} = m^{fake}$ for some m^{fake}, for all $i \in \mathsf{fakers}$. All of our lower bounds in Section 3 hold even with respect to this weaker definition of privacy.

Privacy in Matrix Notation. Pictorially, privacy can be represented as follows, where the left-most column refers to an unqualified set snitches and the right-most column refers to fakers:

$$\begin{bmatrix} m_0 & m_0 \\ m^{fake} & \end{bmatrix} \overset{\text{priv}}{\sim} \begin{bmatrix} m_1 & m_1 \\ m^{fake} & \end{bmatrix}$$

Privacy for Other Flavors. In order to adapt this definition to random or denial-of-service faking, we remove the message m^{fake} from the first bullet point. In order to adapt it to the setting where fakers coordinate, we use the version of the Fake algorithm that takes in and returns a set of shares rather than a single share, and we only give it a single fake message (m^{fake}).

2.2.2 FRI: Fake-Real Indistinguishability Informally, a DSS scheme has *fake-real indistinguishability* if, given a partition of the shareholders into fakers and snitches, it is hard to tell which is which.

Definition 2 ($(\mathsf{partyset}_0, \mathsf{partyset}_1)$-Fake-Real Indistinguishability). *A scheme is* $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-fake-real indistinguishable *if the following holds. Limit* $\mathsf{partyset}_0, \mathsf{partyset}_1$ *to be a partition of* $[n]$*. For any:*

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

Run the following experiment:

- $b \leftarrow \{0,1\}$*;*
- $(s_1^{real}, \ldots, s_n^{real}) \leftarrow \mathsf{Share}(m_b)$*;*
- *For* $i \in \mathsf{partyset}_b$*:* $s_i := s_i^{real}$*;*
- *For* $i \in \mathsf{partyset}_{1-b}$*:* $s_i \leftarrow \mathsf{Fake}(s_i^{real}, m_{1-b})$.

The adversary's distinguishing advantage should be negligible:

$$\Pr[\mathcal{A}(m_0, m_1, \mathsf{partyset}_0, \mathsf{partyset}_1, \{s_i\}_{i\in[n]}) = b] = \frac{1}{2} + \mathsf{negl}$$

Definition 3 (FRI: Fake-Real Indistinguishability). *A scheme is Fake-Real Indistinguishable if it is* $(\mathsf{fakers}, \mathsf{snitches})$*-Fake-Real Indistinguishable for every partition of* $[n]$ *into* fakers *and* snitches*.*

FRI in Matrix Notation. Pictorially, FRI can be represented as follows, where the left-most column refers to $\mathsf{partyset}_0$ and the right-most column refers to $\mathsf{partyset}_1$:

$$\begin{bmatrix} m_0 & m_0 \\ & m_1 \end{bmatrix} \overset{\text{FRI}}{\sim} \begin{bmatrix} m_1 & m_1 \\ m_0 & \end{bmatrix}$$

FRI for Other Flavors. This definition does not apply to denial-of-service faking. In order to adapt it to random faking, we parametrize the definition by a distribution dist, and do not quantify over all m_0, m_1. Instead, we draw $m_b \leftarrow \mathsf{dist}$, and allow the faking process to implicitly determine m_{1-b}. We do not give either message to the adversary, since depending on the number of fakes one of them may be undefined.

In order to adapt this definition to the setting where fakers coordinate, we use the version of Fake algorithm that takes in and returns a set of shares rather than a single share.

2.2.3 FA: Faker Anonymity We give a few versions of faker anonymity. Definition 4 considers a concrete pair of potential faker sets; the adversary should not know which one faked. (The two sets may overlap, so there may be parties who fake in both cases, as well as parties who fake in neither case.) Definition 5 limits the pair of faker sets to differ by only one party. Definition 6 extends Definition 5 to any pair of potential faker sets of a given size that only differ by one party; Definition 7 lifts the size restruction.

Definition 4 ($(\mathsf{fakers}_0, \mathsf{fakers}_1)$-Faker Anonymity). *A scheme is* $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-faker anonymous *if the following holds. Limit* fakers_0 *and* fakers_1 *to be subsets of* $[n]$*. For any:*

- *messages* m^{real}, m^{fake} *s.t.* $|m^{real}| = |m^{fake}|$*;*
- *PPT adversary* $\mathcal{A}$*,*

Run the following experiment:

- $b \leftarrow \{0,1\}$*;*
- $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(m^{real})$*;*
- *For* $i \in \mathsf{fakers}_b$*:* $s_i \leftarrow \mathsf{Fake}(s_i, m^{fake})$*.*

The adversary's distinguishing advantage should be negligible:

$$\Pr[\mathcal{A}(m^{real}, m^{fake}, \mathsf{fakers}_0, \mathsf{fakers}_1, \{s_i\}_{i\in[n]}) = b] = \frac{1}{2} + \mathsf{negl}$$

Definition 5 (FA-2: $(\mathsf{fakers}, i^*, j^*)$-Faker Anonymity). *A scheme is* $(\mathsf{fakers}, i^*, j^*)$*-faker anonymous for* $\mathsf{fakers} \subset [n]$*, and* $i^*, j^* \notin \mathsf{fakers}$ *if it is* $(\mathsf{fakers} \cup \{i^*\}, \mathsf{fakers} \cup \{j^*\})$*-faker anonymous.*

Definition 6 (t_{FA}-Faker Anonymity). *A scheme is* t_{FA}*-faker anonymous if it is* $(\mathsf{fakers}, i^*, j^*)$*-faker anonymous (Definition 5) for every* i^*, j^* *and* $\mathsf{fakers} \subset [n] \setminus \{i^*, j^*\}$ *where* $|\mathsf{fakers}| = t_{FA} - 1$*.*

Definition 7 (Faker Anonymity). *A scheme is faker enonymous if it is* $(\mathsf{fakers}, i^*, j^*)$*-Faker Anonymous (Definition 5) for every* i^*, j^* *and* $\mathsf{fakers} \subset [n] \setminus \{i^*, j^*\}$*.*

FA in Matrix Notation. Pictorially, FA-2 (Definition 5) can be represented as follows, where

- column 1 refers to fakers,
- column 2 refers to $[n] \setminus (\mathsf{fakers} \cup \{i^*, j^*\})$,
- column 3 refers to party i^*, and
- column 3 refers to party j^*.

$$\begin{bmatrix} m^r & m^r & m^r & m^r \\ m^f & & m^f & \end{bmatrix} \underbrace{\qquad}_{\text{FA-2}} \begin{bmatrix} m^r & m^r & m^r & m^r \\ m^f & & & m^f \end{bmatrix}$$

FA for Other Flavors. In order to adapt this definition to denial-of-service faking, we remove m^{fake} everywhere. In order to adapt this definition to random faking, we additionally remove the quantification over all m^{real}. We instead parametrize the definition by a distribution dist, and draw $m^{real} \leftarrow \mathsf{dist}$. The faking process implicitly determines m_{fake}. We don't give m_{fake} to the adversary, since depending on the number of fakes it may be undetermined.

In order to adapt this definition to the setting where fakers coordinate, we use the version of Fake algorithm that takes in and returns a set of shares rather than a single share.

2.2.4 FH: Fake-Hiding Informally, a DSS scheme has *fake hiding* if, given a set of shares, it is hard to tell whether any of them are fake.

Definition 8 (fakers-Fake Hiding). *A scheme is* fakers-fake hiding *if the following holds. Limit* fakers *to be a subset of* $[n]$*. For any:*

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

Run the following experiment:

- $b \leftarrow \{\mathsf{real}, \mathsf{fake}\}$*;*
- *If* $b = \mathsf{real}$*:* $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(m_1)$*;*
- *If* $b = \mathsf{fake}$*:*
  - $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(m_0)$*;*
  - *for* $i \in \mathsf{fakers}$*:* $s_i \leftarrow \mathsf{Fake}(s_i, m_1)$*.*

The adversary's distinguishing advantage should be negligible:

$$\Pr[\mathcal{A}(m_0, m_1, \mathsf{fakers}, \{s_i\}_{i\in[n]}) = b] = \frac{1}{2} + \mathsf{negl}$$

Definition 9 (t_{FH}-Fake Hiding). *A scheme is* t_{FH}*-Fake Hiding if it is* (fakers)*-fake hiding for every subset* fakers *of* $[n]$ *where* $|\mathsf{fakers}| > t_{FH}$*.*

FH in Matrix Notation. Pictorially, FH can be represented as follows, where column 1 refers to fakers, and column 2 refers to snitches:

$$\begin{bmatrix} m_0 & m_0 \\ m_1 & \end{bmatrix} \overset{\text{FH}}{\sim} \begin{bmatrix} m_1 & m_1 \\ & \end{bmatrix}$$

FH for Other Flavors. This definition does not apply to denial-of-service faking. In order to adapt this definition to random faking, we additionally remove the quantification over all m^{real}, m^{fake}. We instead parametrize the definition by a distribution dist, and draw a single message $m \leftarrow \mathsf{dist}$. Depending on the value of b we either apply faking or we don't. The faking process implicitly determines the second message. We no longer give the adversary either message, since one message is guaranteed to be extractable for both values of b, and giving him both would make the game trivial.

In order to adapt this definition to the setting where fakers coordinate, we use the version of Fake algorithm that takes in and returns a set of shares rather than a single share.

2.2.5 Deniable Secret Sharing In order for it deniable secret sharing with shareholder specified or random faking to be meaningful, a minimal condition is that a set of shares *all* of which are fake should look like a set of real shares.

Definition 10. *A scheme* (Share, Rec, Fake) *is a threshold* t *deniable secret sharing with shareholder specified or random faking if it has* t*-privacy (Definition 1) and* $(\emptyset, [n])$*-FRI (Definition 2).*

Of course, this really is minimal; in practice, we would want stronger notions of deniability, and protection for fakers in the presence of snitches. Additional desirable properties include more general FRI, faker anonymity (Definitions 7 and 6) and fake hiding (Definition 9).

2.3 Proofs with the Matrix Notation

Throughout the following sections we will use the matrix notation as a convenient shorthand for our proofs. In the uncoordinated setting security properties may be used to show that sets of shares with various degrees of faking are indistinguishable from one another by replacing earlier fakes.

Consider a partition of our parties into $\mathsf{partyset}_0$ and $\mathsf{partyset}_1$. We may show that two cases are indistinguishable given $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI. In the first m_1 is shared, with the shares for $\mathsf{partyset}_1$ being faked to m_0 and then re-faked to m_0. In the second, m_0 is shared initially, after which the shares of $\mathsf{partyset}_0$ are faked to m_1, while shares for $\mathsf{partyset}_1$ are faked to m_0. Pictorially, for columns $\mathsf{partyset}_0$ and $\mathsf{partyset}_1$,

$$\text{case 1}: \begin{bmatrix} m_1 & m_1 \\ & m_0 \\ & m_0 \end{bmatrix}, \quad \text{case 2}: \begin{bmatrix} m_0 & m_0 \\ m_1 & m_0 \end{bmatrix}.$$

In our notation $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI may be represented as,

$$\begin{bmatrix} m_1 & m_1 \\ & m_0 \end{bmatrix} \underbrace{\qquad}_{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_0 & m_0 \\ m_1 & \end{bmatrix}.$$

Suppose an adversary is able to distinguish case 1 and 2. Such an adversary would break $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI. An adversary distinguishing the cases may be transformed into one breaking $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI, simply by applying "fake to m_0" to the shares for $\mathsf{partyset}_1$. Therefore, $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI implies case 1 and 2 are indistinguishable. In our proofs we will represent the argument above visually as:

$$\begin{bmatrix} m_1 & m_1 \\ & m_0 \\ & m_0 \end{bmatrix} \underbrace{\qquad}_{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_0 \end{bmatrix}.$$

In the above, m_0 shown in blue is the extra fake to m_0 added in the reduction. More generally, indistinguishability is preserved for any polynomial sequence of fakes applied on top of the cases of $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI. In the coming sections we will on occasion use color to clarify which applications of fake are being interchanged.

2.4 Relationships

In this section, we study the relationships amongst the definitions which we present above.

2.4.1 FA Self Implications If a scheme has t_{FA}-FA then any two sets of t_{FA} fakers will be indistinguishable.

Lemma 1. *A scheme has* $(\mathsf{fakers}_0, \mathsf{fakers}_1)$*-faker anonymity for all* $\mathsf{fakers}_0, \mathsf{fakers}_1$ *where* $|\mathsf{fakers}_0| = |\mathsf{fakers}_1| = t_{FA}$ *if it is* t_{FA}*-Faker Anonymous (Definition 6).*

See Appendix A.1 for a proof.

FA for a smaller thresholds implies FA for larger thresholds in the uncoordinated setting.

Lemma 2. *In the uncoordinated setting a scheme has* t_{FA}*-FA if it has* t'_{FA}*-FA for* $t'_{FA} < t_{FA}$*.*

This follows as an adversary may perform additional faking to transform one case to the other. For a proof see Appendix A.2.

2.4.2 FH Implies FA When the snitches are unqualified (which is the only setting in which FH makes sense), FH implies FA.

Lemma 3. *A scheme which is* t_{FH}*-fake hiding is also* t_{FH}*-faker anonymous.*

Proof (of Lemma 3). For all distinct $i^*, j^* \in [n]$, and $\mathsf{fakers} \subset [n] \setminus \{i^*, j^*\}$ where $|\mathsf{fakers}| \geq t_{FH} - 1$, we must show that the scheme is $(\mathsf{fakers}, i^*, j^*)$-faker anonymous. Let $\mathsf{snitches} = [n] \setminus (\mathsf{fakers} \cup \{i^*, j^*\})$ be the remaining parties.

In the matrix notation below, from left to right we let the columns represent sets $\mathsf{fakers}, \mathsf{snitches}, \{i^*\}, \{j^*\}$. We apply t_{FH}-fake hiding:

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & m_1 & \end{bmatrix} \overbrace{\sim\sim\sim}^{(\mathsf{fakers} \cup \{i^*\})\text{-FH}} \begin{bmatrix} m_1 & m_1 & m_1 & m_1 \end{bmatrix}$$

$$\begin{bmatrix} m_1 & m_1 & m_1 & m_1 \end{bmatrix} \overbrace{\sim\sim\sim}^{(\mathsf{fakers} \cup \{j^*\})\text{-FH}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & & m_1 \end{bmatrix}.$$

Given m_0, m_1 and the shares, no adversary can distinguish a fresh sharing of m_1 from a sharing of m_0 where $\mathsf{fakers} \cup \{i^*\}$ have faked to m_1. The same holds for $\mathsf{fakers} \cup \{j^*\}$, implying the scheme is $(\mathsf{fakers}, i^*, j^*)$-Faker Anonymous. □

2.4.3 Privacy and Faker Anonymity Intuitively, these properties aim to hide different things. Privacy aims to hide the original message when enough shares are faked, while faker anonymity aims to hide which parties have faked.

In fact, it is possible to construct schemes which have privacy, but not faker anonymity and vice versa. To achieve privacy one could simply take any regular secret sharing scheme and make faking output the share $\perp$. This scheme does not have faker anonymity as the faked shares are clearly identifiable. On the other hand, if the faking procedure is just the identity, causing fakers to send their original share, then privacy is not achieved, but the fakers are perfectly anonymous.

2.4.4 FH and FRI Only constructions that are symmetric with respect to the $n_{\mathsf{fake}} = \frac{n}{2}$ line in the graphs in Figure 1 can achieve FRI. Because shareholder specified constructions with fake hiding are limited to occupying the top half of such graphs (Theorem 6), a single construction cannot achieve these two properties.

3 Uncoordinated Shareholder Faking: Lower Bounds

In this section, we show that very little is possible when shareholders are the ones to specify the fake message, but do not coordinate. In particular, when faking is not unanimous (that is, there is at least one faker and at least one snitch), we rule out DSS with FRI (Section 3.1) and DSS with FA (Section 3.2). Since FH implies FA, this also (indirectly) rules out FH. (When faking *is* unanimous, in Section 5.5 we show a DSS scheme with FH based on indistinguishability obfuscation.)

3.1 Fake-Real Indistinguishability

Theorem 1 rules out DSS with FRI when neither the fakers nor the snitches are qualified. Theorem 2 similarly rules out DSS with FRI when one of the two sets is qualified.

Theorem 1 (Lower Bound on FRI When Neither Set Has t Parties). *Say we have DSS with uncoordinated shareholder-specified faking (Definition 10). Consider a specific partition of the parties into non-empty sets* $\mathsf{partyset}_0$ *and* $\mathsf{partyset}_1$ *such that* $|\mathsf{partyset}_0| < t$ *and* $|\mathsf{partyset}_1| < t$*. Then, the scheme does not have* $(\mathsf{partyset}_0, \mathsf{partyset}_1)$*-FRI.*

Proof (of Theorem 1). Recall that DSS with shareholder-specified faking must have (a) t-privacy (Definition 1), and (b) $(\emptyset, [n])$-FRI (Definition 2; if *everyone* fakes, the set of fake shares should be indistinguishable from a fresh sharing).

Towards contradiction, assume the scheme has $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI: that is, we have

$$\begin{bmatrix} m_0 & m_0 \\ m_1 & \end{bmatrix} \sim \begin{bmatrix} m_1 & m_1 \\ & m_0 \end{bmatrix}$$

for messages m_0, m_1 of the same length.

Lemma 4. *Consider messages* m_2, m_3, m_4 *s.t.* $|m_2| = |m_3| = |m_4|$.

The following two sets of shares (depicted using our matrix notation) reconstruct to m_4*:*

$$\begin{bmatrix} m_2 & m_2 \\ m_3 & \\ m_4 & \end{bmatrix}, \begin{bmatrix} m_2 & m_2 \\ & m_3 \\ & m_4 \end{bmatrix}$$

Proof (of Lemma 4).

Below, we examine only the case of the left-hand matrix; the same holds for the right-hand matrix by symmetry. We can say the following:

$$\begin{bmatrix} m_2 & m_2 \\ m_3 & \end{bmatrix} \overset{\text{privacy}}{\sim} \begin{bmatrix} m_4 & m_4 \\ m_3 & \end{bmatrix} \overset{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}}{\sim} \begin{bmatrix} m_3 & m_3 \\ & m_4 \end{bmatrix} \overset{\text{privacy}}{\sim} \begin{bmatrix} m_4 & m_4 \\ & m_4 \end{bmatrix}$$

Note that the left-most set of shares contains no information about m_4. Since

$$\begin{bmatrix} m_4 & m_4 \\ m_4 & m_4 \end{bmatrix} \overset{(\emptyset, [n])\text{-FRI}}{\sim} \begin{bmatrix} m_4 & m_4 \end{bmatrix}$$

which reconstructs to m_4 (by correctness), we can conclude that $\begin{bmatrix} m_2 & m_2 \\ m_3 & \\ m_4 & \end{bmatrix}$ also reconstructs to m_4 with overwhelming probability, and, in fact, $\mathsf{partyset}_0$ can cause reconstruction to output an arbitrary message of their choice by re-faking their shares to that message.

Now, we will use Lemma 4 to prove Theorem 1.

Consider the following indistinguishability, which is true by FRI (which we assumed for contradiction).

$$\begin{bmatrix} m_1 & m_1 \\ & m_0 \\ & m_0 \end{bmatrix} \overset{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}}{\sim} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_0 \end{bmatrix}$$

The definition of FRI implies the indistinguishability for the matrix entries in black; given those shares, the adversary can always alter the matrix by re-faking, giving us the entries in blue. Indistinguishability should still hold. By Lemma 4, the left-hand side should reconstruct to m_0. (Note that Lemma 4 does not restrict the messages considered to be different; we can use it here with $m_2 = m_1$, and $m_3 = m_4 = m_0$.) Therefore, the right-hand side should reconstruct to m_0 as well.

We can similarly consider the following indistinguishability.

$$\begin{bmatrix} m_0 & m_0 \\ m_0 & \\ m_1 & \end{bmatrix} \overset{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}}{\sim} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_0 \end{bmatrix}$$

By Lemma 4 with $m_2 = m_3 = m_0$ and $m_4 = m_1$, the left-hand side should reconstruct to m_0, and therefore the right-hand side should as well.

This gives us a contradiction, since we have now shown that $\begin{bmatrix} m_0 & m_0 \\ m_1 & m_0 \end{bmatrix}$ must reconstruct to both m_0 and m_1. □

Theorem 2 (Lower Bound on FRI in the Uncoordinated Setting When One Set Has at Least t Parties). *Say we have DSS with uncoordinated*

shareholder-specified faking (Definition 10). Consider a specific partition of the parties into non-empty sets $\mathsf{partyset}_0$ *and* $\mathsf{partyset}_1$ *such that* $|\mathsf{partyset}_1| \geq t$*. Then, the scheme does not have* $(\mathsf{partyset}_0, \mathsf{partyset}_1)$*-FRI.*

Proof (of Theorem 2). We assume towards contradiction that our scheme has $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI.

Consider messages m_0, m_1, m_2 s.t. $|m_0| = |m_1| = |m_2|$ and $m_1 \neq m_2$. For $b \in \{0, 1\}$, we further partition the sets $\mathsf{partyset}_b$ into $\mathsf{partyset}_b^0$ and $\mathsf{partyset}_b^1$, such that $|\mathsf{partyset}_1^1| = t - 1$ and $|\mathsf{partyset}_0^1| = 1$. Note, $\mathsf{partyset}_0^0$ may be empty. In matrix notation we will now have four columns representing

$$\mathsf{partyset}_0^0, \mathsf{partyset}_0^1, \mathsf{partyset}_1^0, \mathsf{partyset}_1^1,$$

from left to right. We will show that for

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_2 & m_2 \end{bmatrix}$$

reconstructing from the shares of $\mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^1$ must both give m_1 and m_2, providing a contradiction. By construction $\mathsf{partyset}_0^1$ and $\mathsf{partyset}_1^1$ will together provide sufficient shares to reconstruct, while neither is qualified individually.

First we show the shares of $\mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^1$ must reconstruct to m_1:

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_2 & m_2 \end{bmatrix} \overbrace{\qquad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_2 & m_2 & m_2 & m_2 \\ m_0 & m_0 & & \\ m_1 & m_1 & & \end{bmatrix}$$

At this point we cannot exploit privacy with respect to the shares from $\mathsf{partyset}_1$ as they are qualified. However, we may consider the related case where the members of $\mathsf{partyset}_1^0$ have also faked, i.e.

$$\begin{bmatrix} m_2 & m_2 & m_2 & m_2 \\ m_0 & m_0 & m_0 & \\ m_1 & m_1 & & \end{bmatrix} \overbrace{\qquad}^{\text{Privacy}} \begin{bmatrix} m_1 & m_1 & m_1 & m_1 \\ m_0 & m_0 & m_0 & \\ m_1 & m_1 & & \end{bmatrix}.$$

Naturally, the behaviour of reconstruction excluding the shares of $\mathsf{partyset}_1^0$ must be invariant, regardless of the behavior of $\mathsf{partyset}_1^0$. (This is true since in the uncoordinated setting faking is an entirely local procedure. If a particular reconstruction output is required when $\mathsf{partyset}_1^0$ have faked, it must also occur when they have not.) Applying $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI, we see

$$\begin{bmatrix} m_1 & m_1 & m_1 & m_1 \\ m_0 & m_0 & m_0 & \\ m_1 & m_1 & & \end{bmatrix} \overbrace{\qquad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & m_1 \\ & & m_0 & \end{bmatrix},$$

and by $(\emptyset, [n])$-FRI, it follows,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & m_1 \\ & & m_0 & \end{bmatrix} \overbrace{\qquad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} m_1 & m_1 & m_1 & m_1 \\ & & m_0 & \end{bmatrix}.$$

The shares of $\mathsf{partyset}_0^1$ and $\mathsf{partyset}_1^1$ are not affected by the faking by $\mathsf{partyset}_1^0$. Therefore, by correctness, reconstructing from the shares in $\mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^1$ must give m_1 with overwhelming probability.

The case for m_2 follows analogously,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_2 & m_2 \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_1 & m_1 & m_1 & m_1 \\ & & m_0 & m_0 \\ & & m_2 & m_2 \end{bmatrix}.$$

Then,

$$\begin{bmatrix} m_1 & m_1 & m_1 & m_1 \\ m_0 & & m_0 & m_0 \\ & & m_2 & m_2 \end{bmatrix} \overbrace{\quad}^{\text{Privacy}} \begin{bmatrix} m_2 & m_2 & m_2 & m_2 \\ m_0 & & m_0 & m_0 \\ & & m_2 & m_2 \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_2 & m_2 & m_2 & m_2 \\ m_0 & & & \end{bmatrix}.$$

Once again,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_2 & m_2 & m_2 & m_2 \\ m_0 & & & \end{bmatrix} \overbrace{\quad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} m_2 & m_2 & m_2 & m_2 \\ m_0 & & & \end{bmatrix}.$$

Showing that the shares of $\mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^1$ must reconstruct to m_2 with overwhelming probability, by correctness. This gives a contradiction. □

3.2 Faker Anonymity

Theorem 12 rules out faker anonymity (FA) when all but one party is a faker. Corollary 1 then rules out FA for any number of fakers, which follows since FA for t fakers always implies FA for $t+1$ fakers.

Theorem 3 (Lower Bound on FA in the Uncoordinated Setting). *Say we have DSS with uncoordinated shareholder-specified faking (Definition 10). Then, the scheme does not have $(n-1)$-FA (Definition 6).*

Proof (of Theorem 12). Assume toward contradiction the scheme has $(n-1)$-FA. We begin with some simple observations. For any partition of $[n]$ into $\mathsf{fakers}, \{i\}, \{j\}, \{k\}$, represented left to right, $(n-1)$-FA gives,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & \end{bmatrix} \overbrace{\quad}^{(\mathsf{fakers} \cup \{i\}, j, k)\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & & m_1 \end{bmatrix} \overbrace{\quad}^{(\mathsf{fakers} \cup \{k\}, i, j)\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & m_1 & m_1 \end{bmatrix}.$$

Applying the indistinguishability twice it follows that

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & \\ & m_1 & & \end{bmatrix} \overbrace{\quad}^{(\mathsf{fakers} \cup \{j\}, i, k)\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & m_1 \end{bmatrix}$$

$$= \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & m_1 \end{bmatrix} \overbrace{\quad}^{(\mathsf{fakers} \cup \{i\}, k, j)\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & \\ & & m_1 & \end{bmatrix}.$$

For the remainder of this proof will use (i,j)-FA as a shorthand for this indistinguishability. As a step towards our contradiction, for columns $\{i\}, [n] \setminus \{i\}$ we will show we will show an equivalence which we call FA*:

$$\begin{bmatrix} & m_0 & m_0 \\ & m_1 & m_1 \\ n\Big\{ & m_0 & \\ & \vdots & \\ & m_0 & \end{bmatrix} \overset{\text{FA}^*}{\sim} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_1 \end{bmatrix}.$$

Jumping ahead, this results in a contradiction, in the case where all shares have been faked once to m_1 and then n times to m_0. By FRI in the extreme $(\emptyset, [n])$ case this should be indistinguishable from a fresh sharing of m_0. At the same time FA* implies faking the share of a party n times to m_0 should be indistinguishable from not applying these fakes, in which case the shares should be indistinguishable from a fresh sharing of m_1. Pictorially,

$$\begin{bmatrix} m_0 & m_0 \end{bmatrix} \overset{\substack{(n+1)\times \\ (\emptyset,[n])\text{-FRI}}}{\sim} \begin{bmatrix} & m_0 & m_0 \\ & m_1 & m_1 \\ n\Big\{ & m_0 & m_0 \\ & \vdots & \vdots \\ & m_0 & m_0 \end{bmatrix} \overset{\text{FA}^*}{\sim} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_1 \\ & m_0 \\ & \vdots \\ & m_0 \end{bmatrix} \overset{\substack{(n-1)\times \\ \text{FA}^*}}{\sim} \begin{bmatrix} m_0 & m_0 \\ m_1 & m_1 \end{bmatrix} \overset{(\emptyset,[n])\text{-FRI}}{\sim} \begin{bmatrix} m_1 & m_1 \end{bmatrix}.$$

To obtain the desired contradiction we must now prove FA*. Without loss of generality, consider columns $\{1\}, \{2, \ldots, n-1\}, \{n\}$. Our strategy will be to move the fakings to m_0 below the fakes to m_1, allowing them to be eliminated by FRI. Using faker anonymity, we may move the fakes to m_1:

$$\begin{bmatrix} & m_0 & m_0 & m_0 \\ & m_1 & m_1 & m_1 \\ n\Big\{ & m_0 & & \\ & \vdots & & \\ & m_0 & & \end{bmatrix} \overset{(\{2,\ldots,n-1\},1,n)\text{-FA}}{\sim} \begin{bmatrix} & m_0 & m_0 & m_0 \\ n\Big\{ & m_0 & m_1 & m_1 \\ & \vdots & & m_1 \\ & m_0 & & \end{bmatrix}.$$

Our next step is to move the fakes for m_0. However, before we do this we must first apply FRI:

$$\begin{bmatrix} & m_0 & m_0 & m_0 \\ n\Big\{ & m_0 & m_1 & m_1 \\ & \vdots & & m_1 \\ & m_0 & & \end{bmatrix} \overset{(\emptyset,[n])\text{-FRI}}{\sim} \begin{bmatrix} & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 \\ n\Big\{ & m_0 & m_1 & m_1 \\ & \vdots & & m_1 \\ & m_0 & & \end{bmatrix}.$$

Now we are ready to shift the first faking to m_0 under the fakings to m_1:

$$\begin{bmatrix} & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 \\ n\begin{cases} \end{cases} & m_0 & m_1 & m_1 \\ & \vdots & & m_1 \\ & m_0 & & \end{bmatrix} \xrightarrow{(\{2,\ldots,n-1\},1,n)\text{-FA}} \begin{bmatrix} & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 \\ n{-}1\begin{cases} \end{cases} & m_0 & m_1 & m_0 \\ & \vdots & & m_1 \\ & m_0 & & m_1 \end{bmatrix}.$$

We wish to move the fakes applied to party 1 to create a whole row of fakes to m_0, under the fakes to m_1. In the matrix notation, we split the middle column in two, now having columns:

$$\{1\}, [2, n-2], \{n-1\}, \{n\}.$$

We may apply our observation from earlier, applying $(1, n-1)$-FA to move one fake from party 1 to party $n-1$:

$$\begin{bmatrix} & m_2 & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 & m_0 \\ n{-}1\begin{cases} \end{cases} & m_0 & m_1 & m_1 & m_0 \\ & \vdots & & & m_1 \\ & m_0 & & & m_1 \end{bmatrix} \xrightarrow{(1,n-1)\text{-FA}} \begin{bmatrix} & m_2 & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 & m_0 \\ n{-}2\begin{cases} \end{cases} & m_0 & m_1 & m_0 & m_0 \\ & \vdots & & m_1 & m_1 \\ & m_0 & & & m_1 \end{bmatrix}.$$

This process may in fact be repeated, moving one fake at a time to each party. Splitting our columns into

$$\{1\}, [2, i-1], \{i\}, [i+1, n-1], \{n\},$$

this may be shown as:

$$\begin{bmatrix} & m_2 & m_2 & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 & m_0 & m_0 \\ i\begin{cases} \end{cases} & m_0 & m_1 & m_1 & m_0 & m_0 \\ & \vdots & & & m_1 & m_1 \\ & m_0 & & & & m_1 \end{bmatrix} \xrightarrow{(1,i)\text{-FA}} \begin{bmatrix} & m_2 & m_2 & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 & m_0 & m_0 \\ i{-}1\begin{cases} \end{cases} & m_0 & m_1 & m_0 & m_0 & m_0 \\ & \vdots & & m_1 & m_1 & m_1 \\ & m_0 & & & & m_1 \end{bmatrix}.$$

By this sequence of hybrids, the applications of faking to m_0 have now been distributed across the parties. If we once again consider columns

$$\{1\}, [2, n-1], \{n\},$$

this may be visualised as

$$\begin{bmatrix} & m_2 & m_2 & m_2 \\ & m_0 & m_0 & m_0 \\ n\begin{cases} \end{cases} & m_0 & m_1 & m_1 \\ & \vdots & & m_1 \\ & m_0 & & \end{bmatrix} \sim \begin{bmatrix} m_2 & m_2 & m_2 \\ m_0 & m_0 & m_0 \\ m_0 & m_0 & m_0 \\ & m_1 & m_1 \\ & & m_1 \end{bmatrix}.$$

We obtain the final indistinguishability by moving the fakes to m_1 back.

$$\underbrace{\quad}_{2\times(\emptyset,[n])\text{-FRI}}\begin{bmatrix} m_0 & m_0 & m_0 \\ & m_1 & m_1 \\ & & m_1 \end{bmatrix} \underbrace{\quad}_{([2,n-1],1,n)\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 \end{bmatrix}.$$

FA* therefore follows, completing our proof. □

If a scheme with uncoordinated shareholder-specified faking has t-FA then it must also have $(n-1)$-FA (Lemma 2). Since, by Theorem 12, this is impossible, Corollary 1 follows.

Corollary 1. *Say we have a DSS scheme with uncoordinated shareholder-specified faking (Definition 10). Then, the scheme does not have t-FA (Definition 5) for any $t < n$.*

4 Coordinated Shareholder Faking: Lower Bounds for Fake-Hiding

In this section, we show that even when the shareholders coordinate, DSS with fake-hiding is not always achievable. We start with Theorem 4, which rules out information-theoretic DSS with fake-hiding if the fakers are not qualified. We then rule out even computational DSS with fake-hiding if the fakers are not a majority (Theorem 5) and if the snitches are qualified (Theorem 6).

Theorem 4 (Information-Theoretic Lower Bound on FH). *If there are fewer than t fakers, then no information theoretic DSS scheme with coordinated shareholder-specified faking can achieve fake-hiding.*

Proof. Consider the set of shares $(s_1, \ldots, s_n)$, which are either the real sharing of m_1 or a sharing of m_0 where fakers used fake shares instead. Let $\overline{S}$ be the set of snitch shares (known to be real) and S^* denote the unqualified set of shares (which could be either fake or real). We analyze each of the possibilities for S^*.

1. Suppose S^* comprises of real shares. Then, by information-theoretic privacy and the fact that S^* is unqualified, the following must hold: Fix the shares in S^*. Then, for any fixed message m there exist the same number r of possible ways of choosing the remaining shares, such that it corresponds to a sharing of m. If there are a total of k possible messages, then there are rk total ways of choosing the remaining shares such that it corresponds to a valid sharing.

2. Suppose S^* comprises of fake shares. These must have been derived by the fakers from their *original* (real) shares, say $\tilde{S}$. We note that for a scheme with fake hiding, any of the rk versions of the remaining shares that define a valid sharing together with $\tilde{S}$ must now result in a valid sharing of m_1 together with S^*. This is because any of these rk versions could be potential snitch shares. Therefore, for the fixed S^*, there must be rk possible ways of choosing the remaining shares such that it corresponds to a valid sharing of m_1.

Based on the above, we note that an unbounded adversary looking at S^* can count the number of complementary shares yielding m_1, which tells it whether S^* is real or fake. There would be r such ways in the former case, but rk in the latter. The above argument assumes perfect fake hiding, but it can be extended to the statistical case as well. This is because the distribution of messages yielded by the complementary shares must be statistically close to uniform distribution in the real case, but is biased towards m_1 in the fake case (as it results in m_1 with overwhelming probability).

Theorem 5 (Lower Bound on FH when $|\mathsf{fakers}| \leq \frac{n}{2}$). *If $t_{FH} < \frac{n}{2}$ then no DSS scheme with coordinated or uncoordinated shareholder-specified faking with correctness can achieve t_{FH}-Fake Hiding.*

Proof. This follows by the observation that there may be two competing fakes, which both should be reconstructed to. Consider a partition of $[n]$ into $\mathsf{partyset}_0$ and $\mathsf{partyset}_1$, both of which are of size at least t_{FH}. Such a partition must exist as $t_{FH} < n/2$.

Throughout this proof we will let column 1 refer to $\mathsf{partyset}_0$ and column 2 refer to $\mathsf{partyset}_1$. If $\mathsf{partyset}_0$ fakes to m_1 and $\mathsf{partyset}_1$ fakes to m_2 we arrive at a contradiction, shown pictorially:

$$\begin{bmatrix} m_0 & m_0 \\ m_1 & m_2 \end{bmatrix} \overset{(\mathsf{partyset}_0)\text{-FH}}{\sim\!\sim} \begin{bmatrix} m_1 & m_1 \\ & m_2 \end{bmatrix} \overset{(\mathsf{partyset}_1)\text{-FH}}{\sim\!\sim} \begin{bmatrix} m_2 & m_2 \end{bmatrix}$$

while,

$$\begin{bmatrix} m_0 & m_0 \\ m_1 & m_2 \end{bmatrix} \overset{(\mathsf{partyset}_1)\text{-FH}}{\sim\!\sim} \begin{bmatrix} m_2 & m_2 \\ m_1 & \end{bmatrix} \overset{(\mathsf{partyset}_0)\text{-FH}}{\sim\!\sim} \begin{bmatrix} m_1 & m_1 \end{bmatrix}.$$

In this case, correctness implies that the shares should both reconstruct to m_2 and m_1 with overwhelming probability, a contradiction when $m_2 \neq m_1$. (For schemes with binary messages the problem remains for $m_2 = m_0$.)

Theorem 6 (Lower Bound on FH when $|\mathsf{snitches}| \geq t$). *If a DSS scheme has reconstruction threshold t, then it cannot be correct and t_{FH}-Fake Hiding for $t_{FH} \leq n - t$.*

Proof. Let fakers and $\mathsf{snitches}$ be a partition of $[n]$, where $|\mathsf{fakers}| \geq t_{FH}$ and $|\mathsf{snitches}| = t$. (This is possible as $t_{FH} \leq n - t$.) Consider the case where the message m_0 is shared, and the fakers then fake to m_1. By t_{FH}-Fake Hiding, this must be indistinguishable from a fresh sharing of m_1:

$$\begin{bmatrix} m_0 & m_0 \\ m_1 & \end{bmatrix} \overset{(\mathsf{fakers})\text{-FH}}{\sim\!\sim} \begin{bmatrix} m_1 & m_1 \end{bmatrix}$$

where any subset of at least t shares reconstructs to m_1. However, by correctness the shares of $\mathsf{snitches}$ must reconstruct to m_0, giving a contradiction.

5 Coordinated Shareholder Faking from Indistinguishability Obfuscation

In this section, we show that DSS with coordinated shareholder faking with fake hiding is feasible in the computational setting when $n_{\mathsf{fake}} < t$, unlike in the information-theoretic case (as shown by the impossibility in Theorem 4). More specifically, using computational assumptions and indistinguishability obfuscation we are able to construct a secret sharing scheme which allows $n_{\mathsf{fake}} < t$, albeit still requiring $n_{\mathsf{fake}} > n/2$ and $n_{\mathsf{fake}} + t > n$, as imposed by Theorem 5 and Theorem 6 respectively.

When $n_{\mathsf{fake}} < t$ the fakers are unable to appropriately correlate their shares (at least information theoretically) with those held by the snitches, allowing an adversary to detect any faking. We wish to restrict the adversary by constructing an obfuscated program for sharing, reconstructing, and faking, only allowing the adversary to manipulate its shares as specified by the program. As the program cannot be stateful, the shares themselves must contain all the information required to perform these operations. Clearly, these messages must be encrypted as sending them in the clear would render the obfuscated program pointless. Our impossibility results provide insight when designing our programs.

Reconstruction from snitches. In Theorem 6 we observe that the original message cannot be hidden if it is possible to reconstruct using only the snitch shares. If we want to be tight to this bound, then exactly one faker share must be enough to force reconstruction to the fake message.

Competing fakes. When reconstructing from shares where faking has occurred multiple times Theorem 5 shows the importance of imposing an ordering on the fakes. With no ordering it may be ambiguous which of the faked messages should be reconstructed to, contradicting fake hiding.

We will define programs which when obfuscated allow deniable secret sharing. The programs communicate to themselves by sending messages encapsulated under asymmetrically constrained encryption (Definition 15). For correctness these shares must collectively have enough information to specify the message m itself. As reconstruction is controlled by the program we can simply include m in every share along with the index of its recipient.

We ensure that shares from different instances cannot be mixed by deriving a session identifier τ from the provided sharing randomness. (An injective one way function (Definition 13) may be used to ensure that no sessions collide.)

An ordering is imposed on fakes by introducing a level system in the style of [12] as inspired by [6], essentially introducing a counter ℓ describing the extent of faking thus far, where real shares start at level $\ell = 0$. This gives us the plaintext $p_i = (\tau, m, \ell, i)$. When a set of shares of level at most ℓ are input to the faking algorithm, the new shares produced will be of level $\ell + 1$. We ensure the most recent fake takes precedence by reconstructing the message of the share with the highest level. The latest fake will have this highest level, since sequential applications of fake must always overlap (as we have $n_{\mathsf{fake}} > n/2$).

For shares to be of fixed size, regardless of the extent of faking, levels must always remain within some fixed range $[0, T]$. If the adversary were able to reach the upper bound T by repeated applications of fake it would clearly be able to distinguish a set of shares with level 0 from a set of level 1. In prior work [6,12] it has been shown that if T is exponentially large an adversary cannot distinguish these worlds. However, these existing approaches require T hybrids, necessitating sub-exponential hardness assumptions.

We observe that our setting is somewhat relaxed and therefore allows an alternate approach. Rather than starting levels counter at 0 we start at a pseudorandom point 0_τ derived from the session identifier τ using a PRF. This allows us to rely on the statistical closeness of uniform values in $[0, T]$ and $[1, T+1]$ for exponential T.

5.1 Circuit Obfuscation

In [2] Barak *et al.* proposed the notion of *indistinguishability obfuscation*, allowing a circuit to be transformed into an obfuscated form perfectly preserving functionality, but hiding implementation details. Security requires the obfuscations of any two circuits of the same size with identical functionalities to be computationally indistinguishable. Several years later Garg *et al.* [15] proposed the first candidate construction for general (boolean) circuits.

Definition 11 (Indistinguishability Obfuscation[15]). *A uniform PPT machine* iO *is called an indistinguishability obfuscator for the circuit class* $\{\mathcal{C}_\lambda\}$ *if the following hold,*

- *For all* $\lambda \in \mathbb{N}$ *and all circuits* $C \in \mathcal{C}_\lambda$ *and all inputs* x

$$\Pr[C' \leftarrow \mathsf{iO}(\lambda, C) : C(x) = C'(x)] = 1.$$

- *For any PPT distinguisher* D *there exists a negligible function* negl *such that for all* λ*, and any two circuits* $C_0, C_1 \in \mathcal{C}_\lambda$ *where* $C_0(x) = C_1(x)$ *for all* x *then*

$$|\Pr[D(\mathsf{iO}(\lambda, C_0))] - \Pr[D(\mathsf{iO}(\lambda, C_1))]| \leq \mathsf{negl}(\lambda).$$

We will further require a weak variant of extractability obfuscation introduced in [8]. In contrast to indistinguishability obfuscation, this considers programs which differ at a small, polynomially bounded, number of inputs. Such programs are not guaranteed to be indistinguishable after obfuscation, however the existence of a distinguisher with polynomial advantage implies an extractor which efficiently finds an input at which the programs differ, also with polynomial advantage.

Definition 12 (Weak Extractability Obfuscation [8, Definition 6.1]). *A uniform PPT machine* $\mathcal{O}$ *is a weak extractability obfuscator for a class of turing machines* $\mathcal{M} = \{\mathcal{M}_k\}$ *if it satisfies the following. For every PPT adversary* $\mathcal{A}$ *and polynomial* $p(k)$*, there exists a PPT algorithm* E *and polynomials*

$p_E(k), t_E(k)$ for which the following holds. For every polynomial $d(k)$ and sufficiently large k, and ever pair $M_0, M_1 \in \mathcal{M}_k$ differing on at most $d(k)$ inputs, and every auxiliary input z,

$$\Pr\left[b \leftarrow \{0,1\}; \tilde{M} \leftarrow \mathcal{O}(1^k, M_b) : \mathcal{A}(1^k, \tilde{M}, M_0, M_1, z) = b\right] \geq \frac{1}{2} + \frac{1}{p(k)}$$

$$\implies \Pr\left[x \leftarrow E(1^k, M_0, M_1, z) : M_0(x) \neq M_1(x)\right] \geq \frac{1}{p_E(k)}$$

where the runtime of E is $t_E(k, d(k))$.

For the circuits we are interested in indistinguishability obfuscation and weak extractability obfuscation are closely linked. Informally, if finding an input at which two programs differ helps solve a hard computational problem then obfuscating the two programs with iO is sufficient to achieve indistinguishability, despite their functionalities differing.

Theorem 7 ([8, Theorem 6.2]). *Let* iO *be an indistinguishability obfuscator for P/poly, then* iO *is also a weak extractability obfuscator for P/poly.*

5.2 Indistinguishability Obfuscation Friendly primitives

To construct our deniable secret sharing scheme from indistinguishability obfuscation we will follow the *punctured programs* paradigm introduced by Sahai and Waters [20].

Puncturable Pseudorandom Functions. Puncturable pseudorandom functions, realised by the GGM PRF [17], enable puncturing the key K of a PRF F for a polynomially bounded set S. The punctured key $K\{S\}$ allows evaluating the PRF at all points outside S, while all evaluations $F_K(x)$ for $x \in S$ appear pseudorandom even given $K\{S\}$.

Definition 13 (Puncturable Pseudorandom Function). *For input size $n = n(\lambda)$ and output size $m = m(\lambda)$ a puncturable pseudorandom function (PPRF) family is defined by a tuple of PPT algorithms $\{\mathsf{Sample}, \mathsf{Puncture}, \mathsf{Eval}\}$. Using $F_K(x)$ as a shorthand for $\mathsf{Eval}(K, x)$, a PPRF must satisfy the two following properties.*
Functionality preserved under puncturing: *For any $S \subset \{0,1\}^n$ where $|S| = \mathsf{poly}(\lambda)$ and $x \notin S$,*

$$\Pr\left[F_K(x) = F_{K\{S\}}(x) \mid K \leftarrow \mathsf{Sample}(1^\lambda); K\{S\} \leftarrow \mathsf{Puncture}(K, S)\right] = 1.$$

Pseudorandomness at punctured points: *For any $S \subset \{0,1\}^n$ where $|S| = \mathsf{poly}(\lambda)$, and any PPT adversary $\mathcal{A}$,*

$$\left|\Pr\left[\mathcal{A}\left(K\{S\}, (F_K(x_i))_{x_i \in S}\right) = 1\right] - \Pr\left[\mathcal{A}\left(K\{S\}, U^{|S|}_{m(\lambda)}\right) = 1\right]\right| = \mathsf{negl}(\lambda),$$

Injective OWF. As shown in [7] injective one-way functions can be constructed from iO and one-way functions.

Definition 14 (Injective One-Way Function [7]). *For polynomially bounded length functions k, τ, let*

$$\{\mathsf{OWF}_K : \{0,1\}^\lambda \to \{0,1\}^{\tau(\lambda)}\}_{K \in \{0,1\}^{k(\lambda)}}$$

be an injective one-way function family if it has an efficient key sampling algorithm Sample *such that*

- *for every $K \in \{0,1\}^{k(\lambda)}$ the function OWF_K is injective, and*
- *for every polysize adversary $\mathcal{A}$,*

$$\Pr\left[K \leftarrow \mathsf{Sample}(1^\lambda); x \leftarrow \{0,1\}^\lambda : \mathcal{A}(K, \mathsf{OWF}_K(x)) = x\right] = \mathsf{negl}(\lambda).$$

5.3 Asymmetrically Constrained Encryption

Introduced in [11] asymmetrically constrained encryption (ACE) is a deterministic authenticated encryption scheme which allows puncturing both at encryption and decryption. The encryption key may be constrained for a message m to obtain $EK\{m\}$ which does not allow encrypting m. The decryption key may also be punctured, possibly for different messages, where $DK\{m\}$ does not allow decrypting the encryption of m. Informally, an ACE scheme must satisfy:

- *Equivalence of constrained keys:* Constrained keys, $EK\{S\}, DK\{S\}$, should be equivalent to their unconstrained counterparts on all messages outside the set S.
- *Unique ciphertexts:* There should only be one ciphertext which decrypts to a message m under a particular decryption key DK.
- *Security of Constrained Decryption:* An adversary given $EK\{U\}$ should not be able to distinguish two decryption keys $DK\{S_0\}$ and $DK\{S_1\}$ where $S_0 \subseteq S_1 \subseteq U$. The adversary may also receive ciphertexts for chosen messages $m \notin S_1 \setminus S_0$.
- *Selective Indistinguishability of Ciphertexts:* Given $EK\{U\}, DK\{S\}$ an adversary should not be able to distinguish, the ciphertexts for two messages $m_0, m_1 \in U \cap S$, even in the presence of encryptions of other messages.

For the sake of completeness, we recall ACE formally as described by [12] in Appendix C.1.

5.4 The iO Construction

We construct a secret sharing scheme for messages $m \in \mathcal{M}$ where $|\mathcal{M}|$ is polynomially bounded in λ. The scheme may be composed in parallel in the straightforward way for an exponential message space. Let $T \in \mathbb{N}$ be a positive integer which is exponentially large in λ, and $\mathcal{R} = \{0,1\}^\lambda$ be a set of random tapes for Share. The obfuscated programs are produced as follows.

– Generate a key for an injective one-way function

$$\mathsf{G} : \mathcal{K}_{\mathsf{OWF}} \times \mathcal{R} \to \{0,1\}^{\tau(\lambda)}$$

as $K_1 \leftarrow \mathsf{OWF.Sample}(1^\lambda)$.

– Further sample a key K_2 for the puncturable PRF

$$\mathsf{H} : \mathcal{K}_{\mathsf{PPRF}} \times \{0,1\}^{\tau(\lambda)} \to [T]$$

as $K_2 \leftarrow \mathsf{PPRF.Sample}(1^\lambda)$.

– For the asymmetrically constrained encryption generate a secret key $SK \leftarrow \mathsf{ACE.Setup}(1^\lambda, 1^m, 1^s)$, setting m large enough that the plaintext space $\{0,1\}^m$ may encode all

$$(\tau, m, \ell, i) \in \Big(\{0,1\}^{\tau(\lambda)} \times \mathcal{M} \times [0, T+1] \times [n]\Big).$$

In our reductions we will only ever constrain ACE by message sets polynomial in n and $|\mathcal{M}|$, allowing $s = \mathsf{poly}(m, n, |\mathcal{M}|)$. Use SK to produce encryption and decryption keys $EK \leftarrow \mathsf{GenEK}(SK, C_\emptyset); DK \leftarrow \mathsf{GenDK}(SK, C_\emptyset)$.

– Let $C = (\mathsf{Share}, \mathsf{Fake}, \mathsf{Rec})$ be the circuit allowing the evaluation of the circuits specified in Figure 3. For some appropriate padding pad let $C' \leftarrow \mathsf{iO}(1^\lambda, \mathsf{pad}(C))$. Instantiate t_{fake} s.t. $t_{\mathsf{fake}} > \max\{\frac{n}{2}, n - t\}$.

The circuit C' may either be produced by a trusted setup and given as a common reference string, or by the dealer at the time of sharing: distributing the obfuscated programs with the shares. To share a secret m simply sample randomness $\rho \leftarrow \mathcal{R}$ and compute

$$\{(i, s_i)\}_{i \in [n]} \leftarrow C'.\mathsf{Share}(m; \rho),$$

obtaining shares s_i for $i \in [n]$. Reconstruction and faking similarly proceed in the straightforward way by invoking the corresponding obfuscated programs.

At a very high-level, when faking (Fake) takes as input a set of shares, say S, of size at least t_{fake}, it checks that the shares at the highest level (say ℓ) are consistent with respect to the message and if so, it updates *each* of the shares in S to level $\ell + 1$ and corresponding to the fake message. Reconstruction (Rec), on the other hand, takes as input any qualified set of shares and simply returns the message corresponding to the highest level. Intuitively, this approach prevents adversary from gaining information about whether faking occurred due to the following: even if the adversary repeatedly attempts Fake with different potential sets S, any attempt at Rec would always correspond to the message it used in the latest faking. This is because any two consecutive faking attempts must have a common party (as $t_{\mathsf{fake}} > n/2$ holds) which will correspond to the highest level. Further since shares of at least t_{fake} parties will have the highest level, Rec which requires t shares must necessarily include a share of the highest level (otherwise $n \geq t + t_{\mathsf{fake}}$ must hold, which contradicts our assumption). We can thus infer that any attempts to Rec must reconstruct the message corresponding to the most recent fake.

$\mathsf{Share}(m;\rho)$

1 : $\tau \leftarrow \mathsf{G}_{K_1}(\rho)$ // Generate session id
2 : $0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$ // Starting level
3 : **for** $i \in [n]$:
4 : $s_i \leftarrow \mathsf{Enc}_{EK}((\tau, m, 0_\tau, i))$
5 : **return** $\{(i, s_i)\}_{i\in[n]}$

$\mathsf{Rec}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$

1 : $(\tau, m, \ell, \mathsf{partyset}')$
$\leftarrow \mathsf{Decrypt}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$
2 : **if** $|\mathsf{partyset}'| \le t$ **then return** $\perp$
3 : **return** m

$\mathsf{Fake}(\{(i, s_i)\}_{i\in\mathsf{partyset}}, m^{fake})$

1 : $(\tau, m, \ell, \mathsf{partyset}')$
$\leftarrow \mathsf{Decrypt}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$
2 : **if** $|\mathsf{partyset}'| \le t_{\mathsf{fake}}$ **then return** $\perp$
3 : **for** $i \in \mathsf{partyset}'$:
4 : $s'_i \leftarrow \mathsf{Enc}_{K_2}((\tau, m^{fake}, \ell + 1, i))$
5 : **return** $\{(i, s'_i)\}_{i\in\mathsf{partyset}'}$

$\mathsf{Decrypt}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$ (Subcircuit only)

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $p \leftarrow \mathsf{Dec}_{DK}(s_i)$
4 : **if** $p = \perp$ **then continue**
5 : **else parse** $(\tau_i, m_i, \ell_i, j) := p$
6 : $0_{\tau_i} \leftarrow \mathsf{H}_{K_2}(\tau_i)$
7 : **if** $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
8 : $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
9 : $j \leftarrow \mathsf{argmax}_{i\in\mathsf{partyset}'}\ \ell_i$
10 : $\tau \leftarrow \tau_j;\ m \leftarrow m_j;\ \ell \leftarrow \ell_j$
// Check all shares are derived from
11 : // the same sharing.
12 : **if** $\exists\ i \in \mathsf{partyset}', \tau_i \neq \tau$
13 : **then return** $(\perp, \perp, \perp, \emptyset)$
// Check consistency accross the highest level.
14 : **if** $\exists\ i \in \mathsf{partyset}', \ell_i = \ell \wedge (m_i \neq m)$
15 : **then return** $(\perp, \perp, \perp, \emptyset)$
16 : **return** $(\tau, m, \ell, \mathsf{partyset}')$

Fig. 3: Circuits for coordinated faking from iO, using Decrypt as a subcircuit.

Theorem 8. *Let* $t_{\mathsf{fake}} > \max\{n/2, n - t\}$. *Then this construction is* n_{fake}*-Fake-Hiding (Definition 9) for* $n_{\mathsf{fake}} \ge t_{\mathsf{fake}}$ *if* G *is an injective one-way function,* H *is a puncturable pseudorandom function,* ACE *is an asymmetrically constrained encryption scheme, and* iO *is an indistinguishability obfuscator.*

We provide a high level sketch of our proof strategy, postponing a formal proof to Appendix C.2.

Proof sketch. We will gradually modify the branch where faking has occurred, $b = \mathsf{fake}$, until it is distributed identically to the branch with no faking, $b = \mathsf{real}$. For randomness ρ^* the adversary receives the ciphertexts,

$$\{\mathsf{Enc}_{EK}((\tau^*, m_0, 0_{\tau^*}, i))\}_{i\in\mathsf{snitches}}, \quad \{\mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*} + 1, i))\}_{i\in\mathsf{fakers}}$$

where $\tau^* = \mathsf{G}_{K_1}(\rho^*)$ and $0_{\tau^*} = \mathsf{H}_{K_2}(\tau^*)$.

We must ensure that the real shares obtained by the adversary from the snitches cannot be used to reconstruct the real secret. When faking has taken place we know the $n - t_{\mathsf{fake}}$ snitch shares are insufficient to reconstruct. However, the adversary may be able to reconstruct if it can obtain more level 0_{τ^*} shares, e.g. by finding the randomness used during the original sharing. We may

prevent this by modifying Share to output $\perp$ when $\tau^* = \mathsf{G}_{K_1}(\rho)$. This alters the functionality of Share, preventing the use of iO security directly. Instead, using Theorem 7 it may be shown that an adversary distinguishing the two programs may be used to break the one-wayness of G by finding a preimage of τ^*.

Perhaps counterintuitively, our strategy will start by increasing the level of the snitch shares, so all shares are of level $0_{\tau^*} + 1$. Allowing the levels to be addressed at once, later in the proof. Our goal is to substitute the ciphertexts for $i \in \mathsf{snitches}$ as

$$s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_0, 0_{\tau^*}, i)) \Longrightarrow \hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*} + 1, i)).$$

However, before selective indistinguishability may apply, we must first have to constrain the encryption and decryption key for the involved plaintexts, preventing any trivial distinguishers.

1. Due to the previous modifications to Share, constraining encryption of all plaintexts in $\{(\tau^*, m, 0_{\tau^*}, i)\}_{i\in[n],m\in\mathcal{M}}$ has no effect on functionality, and is indistinguishable by iO security.
2. Decryption may be constrained similarly, taking care to appropriately hardcode the decryption of $s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_0, 0_{\tau^*}, i))$ for $i \in \mathsf{snitches}$. This must be done in two steps, as security of constrained decryption does not allow constraining plaintexts for which the adversary receives ciphertexts.
3. At this point we may be sure that the only shares which decrypt with level 0_{τ^*} are those in $\{s_i^*\}_{i\in\mathsf{snitches}}$. Since $t_{\mathsf{fake}} > n/2$, we know $n - t_{\mathsf{fake}}$ shares are insufficient for faking, the programs will never encrypt shares of level $0_{\tau^*} + 1$ for the adversary. We may therefore restrict encryption and then decryption for the set,

   $$\{(\tau^*, m, 0_{\tau^*} + 1, i)\}_{i\in[n],m\in\mathcal{M}} \setminus \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i\in\mathsf{fakers}}$$

   respectively applying iO security and the security of constrained decryption once again.
4. At this point we may modify the hardcoded decryption:

   $$(\tau^*, m_0, 0_{\tau^*}, i) \Longrightarrow (\tau^*, m_1, 0_{\tau^*} + 1, i)$$

   for $i \in \mathsf{snitches}$. Importantly, the functionalities of Fake and Rec are invariant of this change, as $\{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i\in\mathsf{fakers}}$ are the only other plaintexts of level $0_{\tau^*} + 1$ that will ever decrypt.
   Any call to Fake and Rec with shares of level $\ell > 0_{\tau^*} + 1$ will clearly be unaffected by this change. This leaves calls with a mix of levels 0_{τ^*} and $0_{\tau^*} + 1$. Calls of this type with at least t_{fake} or t shares must contain at least one share of level $0_{\tau^*} + 1$, as $n_{\mathsf{fake}} \geq t_{\mathsf{fake}} > \max\{n/2, n - t\}$. Increasing the other shares from level 0_{τ^*} to $0_{\tau^*} + 1$ has no effect on the maximal level, preserving the functionality of the $\mathsf{Decrypt}$ subcircuit and therefore also the functionalities of Fake and Rec

5. Having appropriately constrained encryption and decryption the ciphertexts s_i^* may be exchanged with $\hat{s}_i$, relying on selective indistinguishability. After this change we may undo the hardcoded decryption without affecting functionality, and relax the constraints on decryption to only be for $\{(\tau^*, m, 0_{\tau^*}, i)\}_{i\in[n],m\in\mathcal{M}}$.

We are now ready to address the levels. The constraints which remain on decryption enforce that all shares in Rec and Fake have levels in the range $[0_{\tau^*}+1, T]$. Meanwhile, the ciphertexts given to the adversary are

$$\{\mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}+1, i))\}_{i\in[n]}.$$

If we puncture the key for H at τ^*, then it follows by pseudorandomness that this starting level $0_{\tau^*}+1$ is indistinguishable from uniform in $[1, T+1]$. In the branch without faking the starting level is indistinguishable from uniform over $[0, T]$. For exponential T these distributions are statistically close, allowing the level to be replaced:

$$\{\mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}, i))\}_{i\in[n]}.$$

All that remains is to undo the puncturing and constrianing of keys, and removing the check for $\tau^* = \mathsf{G}_{K_1}(\rho)$ from Share. Once the programs are restored the branch for $b = \mathsf{fake}$ will be identical to $b = \mathsf{real}$.

5.5 Extreme FRI and Privacy Without Coordination

The construction in Section 5.4 may be modified to allow uncoordinated faking and achieve $(\emptyset, [n])$-FRI. Simply alter Fake to only require one share. Given a ciphertext, the program decrypts it, increments the level by one and replaces the message with the new m^{fake}. Reconstruction should require all shares to be of the same level and have the same message, needing at least t shares to ensure t-privacy.

To have $(\emptyset, [n])$-FRI the adversary must be unable to distinguish the distributions

$$\{\mathsf{Enc}_{EK}((\tau^*, m, 0_{\tau^*}, i))\}_{i\in[n]}$$

and

$$\{\mathsf{Enc}_{EK}((\tau^*, m, 0_{\tau^*}+1, i))\}_{i\in[n]}.$$

This may be shown by a similar argument as for the proof of Theorem 8, if G is an injective one-way function, H is a puncturable PRF, ACE is an asymmetrically constrained encryption scheme, and iO is an indistinguishability obfuscator. Note that this construction does not achieve any notion of deniability when $\mathsf{fakers} \neq \emptyset$ AND $\mathsf{snitches} \neq \emptyset$.

References

1. Joël Alwen, Rafail Ostrovsky, Hong-Sheng Zhou, and Vassilis Zikas. Incoercible multi-party computation and universally composable receipt-free voting. In

Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 763–780, Santa Barbara, CA, USA, August 16–20, 2015.

2. 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, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18, Santa Barbara, CA, USA, August 19–23, 2001.
3. Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *26th Annual ACM Symposium on Theory of Computing*, pages 544–553, Montréal, Québec, Canada, May 23–25, 1994. ACM Press.
4. 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: 18th Theory of Cryptography Conference, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 260–290, Durham, NC, USA, November 16–19, 2020.
5. 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: 29th Conference on Computer and Communications Security*, pages 323–336, Los Angeles, CA, USA, November 7–11, 2022. ACM Press.
6. Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 1480–1498, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.
7. Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 474–502, Tel Aviv, Israel, January 10–13, 2016.
8. Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 52–73, San Diego, CA, USA, February 24–26, 2014.
9. Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 90–104, Santa Barbara, CA, USA, August 17–21, 1997.
10. Ran Canetti and Rosario Gennaro. Incoercible multiparty computation (extended abstract). In *37th Annual Symposium on Foundations of Computer Science*, pages 504–513, Burlington, Vermont, October 14–16, 1996. IEEE Computer Society Press.
11. Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Indistinguishability obfuscation of iterated circuits and RAM programs. Cryptology ePrint Archive, Report 2014/769, 2014.
12. Ran Canetti, Sunoo Park, and Oxana Poburinnaya. Fully deniable interactive encryption. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 807–835, Santa Barbara, CA, USA, August 17–21, 2020.

13. Ran Canetti and Oxana Poburinnaya. Towards multiparty computation withstanding coercion of all parties. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 410–438, Durham, NC, USA, November 16–19, 2020.
14. Stefan Dziembowski, Sebastian Faust, Tomasz Lizurej, and Marcin Mielniczuk. Secret sharing with snitching. In *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 840–853. ACM Press, November 2024.
15. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.
16. 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, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 64–93, Virtual Event, August 16–20, 2021.
17. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. On the cryptographic applications of random functions. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 276–288, Santa Barbara, CA, USA, August 19–23, 1984.
18. Vipul Goyal, Yifan Song, and Akshayaram Srinivasan. Traceable secret sharing and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 718–747, Virtual Event, August 16–20, 2021.
19. Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. Cryptology ePrint Archive, Report 2022/639, 2022.
20. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

A Omitted details from Section 2

A.1 Proof of Lemma 1

Proof (of Lemma 1). For all messages m^{real}, m^{fake} s.t. $|m^{real}| = |m^{fake}|$, and subsets fakers of $[n]$, we define the random variable

$$\mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}) := (m^{real}, m^{fake}, \mathsf{fakers}, \{s_i\}_{i\in[n]})$$

where the distribution of the shares is taken over the random coins of the process

- $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(m^{real})$;
- For $i \in \mathsf{fakers}$: $s_i \leftarrow \mathsf{Fake}(s_i, m^{fake})$.

We will show that if a scheme has t_{FA}-Faker Anonymity then

$$\mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}_0) \approx \mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}_1)$$

for any pair of partitions s.t. $|\mathsf{fakers}_0| = |\mathsf{fakers}_1| \geq t_{FA}$. First, let

$$\mathsf{addedfakers} = \mathsf{fakers}_1 \setminus \mathsf{fakers}_0, \quad \mathsf{removedfakers} = \mathsf{fakers}_0 \setminus \mathsf{fakers}_1.$$

Clearly, if $|\mathsf{fakers}_0| = |\mathsf{fakers}_1|$ then $|\mathsf{addedfakers}| = |\mathsf{removedfakers}|$. Let k be the size of these sets, and define an arbitrary ordering on their elements

$$\mathsf{addedfakers} = \{a_1, \ldots, a_k\}, \ \mathsf{removedfakers} = \{r_1, \ldots, r_k\}$$

For $i = 0, \ldots, k$, where $\mathsf{fakers}^i = (\mathsf{fakers} \setminus \{r_1, \ldots, r_i\}) \cup \{a_1, \ldots, a_i\}$,

$$\mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}^{i-1}) \approx \mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}^{i})$$

would be implied exactly by $((\mathsf{fakers}_0 \setminus \{r_1, \ldots, r_i\}) \cup \{a_1, \ldots, a_{i-1}\}, r_i, a_i)$-FA. As we assume t_{FA} -Faker Anonymity as $|(\mathsf{fakers} \setminus \{r_1, \ldots, r_i\}) \cup \{a_1, \ldots, a_{i-1}\}| \geq t_{FA} - 1$. Indistinguishability follows by a simple hybrid argument, giving

$$\mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}_0 = \mathsf{fakers}^0) \approx \mathcal{D}(m^{real}, m^{fake}, \mathsf{fakers}_1 = \mathsf{fakers}^k).$$

A.2 Proof of Lemma 2

Proof (of Lemma 2). Assume our scheme has t'_{FA}-FA and consider any partition of $[n]$ into fakers, i, j. Let m_0, m_1 be any two messages of the same length. For the sake of our proof, further partition fakers into $\mathsf{fakers}^{t'_{FA}}$ and $\mathsf{fakers}^{n-(t'_{FA}+2)}$, where $|\mathsf{fakers}^{t'_{FA}}| = t'_{FA}$. From left to right, we let our columns be,

$$\mathsf{fakers}^{t'_{FA}}, \mathsf{fakers}^{n-(t'_{FA}+2)}, \{i\}, \{j\}.$$

By assumption,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & m_1 & \end{bmatrix} \overbrace{\quad}^{t'_{FA}\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & & m_1 \end{bmatrix},$$

$(n-1)$-FA follows as,

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & m_1 & \end{bmatrix} \overbrace{\quad}^{t'_{FA}\text{-FA}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & m_1 & & m_1 \end{bmatrix}.$$

An adversary distinguishing in the case of $(n-1)$-FA may be used to break t'_{FA}-FA, simply by faking for the shares of $\mathsf{fakers}^{n-(t'_{FA}+2)}$ to m_1 first. Therefore, the scheme must have $(n-1)$-FA.

Proof (of Lemma 3). For all distinct $i^*, j^* \in [n]$, and $\mathsf{fakers} \subset [n] \setminus \{i^*, j^*\}$ where $|\mathsf{fakers}| \geq t_{FH} - 1$, we must show that the scheme is $(\mathsf{fakers}, i^*, j^*)$-faker anonymous. Let $\mathsf{snitches} = [n] \setminus (\mathsf{fakers} \cup \{i^*, j^*\})$ be the remaining parties.

In the matrix notation below, from left to right we let the columns represent sets $\mathsf{fakers}, \mathsf{snitches}, \{i^*\}, \{j^*\}$. We apply t_{FH}-fake hiding:

$$\begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & m_1 & \end{bmatrix} \overbrace{\quad}^{(\mathsf{fakers} \cup \{i^*\})\text{-FH}} \begin{bmatrix} m_1 & m_1 & m_1 & m_1 \end{bmatrix}$$

$$\begin{bmatrix} m_1 & m_1 & m_1 & m_1 \end{bmatrix} \overbrace{\qquad}^{(\mathsf{fakers} \cup \{j^*\})\text{-FH}} \begin{bmatrix} m_0 & m_0 & m_0 & m_0 \\ m_1 & & & m_1 \end{bmatrix}.$$

Given m_0, m_1 and the shares, no adversary can distinguish a fresh sharing of m_1 from a sharing of m_0 where $\mathsf{fakers} \cup \{i^*\}$ have faked to m_1. The same holds for $\mathsf{fakers} \cup \{j^*\}$, implying the scheme is $(\mathsf{fakers}, i^*, j^*)$-Faker Anonymous. □

B Coordinated Shareholder Faking: Information-Theoretic Construction

In this section, we describe the information theoretic DSS scheme with coordinated shareholder faking.

Theorem 9. *There is an information theoretic DSS scheme with coordinated shareholder faking that*

1. *achieves fake-hiding when $n - n_{\mathsf{fake}} < t$ and $n_{\mathsf{fake}} \geq t$.*
2. *achieves fake-real indistinguishability and faker anonymity for all thresholds.*

Proof. Consider the following construction based on Shamir-secret sharing (say, over a finite field $\mathbb{F}_q$ of integers modulo q, with prime $q > n$).

1. $\mathsf{Share}(m; \rho)$: Computes $(s_1, \ldots, s_n) \leftarrow \mathsf{Shamir.Share}(m, t)$ as a shamir-sharing of the message $m \in \mathbb{F}_q$ with threshold t. Here, $\mathsf{Shamir.Share}$ involves sampling a polynomial $p(x)$ of degree $(t-1)$ such that $p(0) = m$ and setting $s_i = p(i) \mod q$. Return $\{(i, s_i)\}_{i \in [n]}$.

2. $\mathsf{Fake}(\{s_i\}_{i \in \mathsf{fakers}}, m^{fake})$:
   - If $n - |\mathsf{fakers}| \geq t$ (that is, the snitches are qualified), return the real shares $\{s_i\}_{i \in \mathsf{fakers}}$.
   - Else: If $|\mathsf{fakers}| \geq t$
     (a) Use Lagrange interpolation to compute the $(t-1)$-degree polynomial $p(x)$ that is consistent with $\{s_i\}_{i \in \mathsf{fakers}}$.
     (b) Next, sample a uniform $(t-1)$-degree polynomial $p'(x)$ such that $p'(j) = p(j)$ for each $j \in [n] \setminus \mathsf{fakers}$ and $p'(0) = m^{fake}$.
     (c) Set $s'_i = p'(i)$ for $i \in \mathsf{fakers}$. Return $\{(i, s'_i)\}_{i \in \mathsf{fakers}}$.
   - Else (i.e. when $|\mathsf{fakers}| < t$ and $n - |\mathsf{fakers}| < t$, sample $s'_i \in \mathbb{F}_q$ uniformly at random for each $i \in \mathsf{fakers}$. Return $\{(i, s'_i)\}_{i \in \mathsf{fakers}}$.

3. $\mathsf{Rec}(\{s_i\}_{i \in \mathsf{partyset}})$: Computes $m' \leftarrow \mathsf{Shamir.Rec}(\{s_i\}_{i \in \mathsf{partyset}}, t)$. Here, $\mathsf{Shamir.Rec}$ involves using Lagrange interpolation to identify a $(t-1)$-degree polynomial $p''(x)$ consistent with $\{s_i\}_{i \in \mathsf{partyset}}$. Return $p''(0)$ if such a polynomial exists and $\perp$ otherwise.

We analyze the properties achieved by the above construction below.

– When $n - |\mathsf{fakers}| \geq t$ (that is, the snitches are qualified), the construction achieves FRI and FA. Since the fakers are also providing their real shares, these properties follow directly. Note that privacy is applicable only for settings where snitches are not qualified, therefore this does not contradict privacy. Further, fake hiding is also not applicable to this setting (Theorem 6).

– Next, when $n - |\mathsf{fakers}| < t$ and $|\mathsf{fakers}| \geq t$ (that is, the snitches are unqualified and fakers are qualified), the construction achieves FRI, FA and fake hiding as well. This is because the fake shares provided by the coordinated fakers are such that they are consistent with the real shares given by the snitches. Therefore, the adversary's view is identically distributed in the cases where a real sharing of m^{fake} occurred or faking occurred.

– Lastly, when $n - |\mathsf{fakers}| < t$ and $|\mathsf{fakers}| < t$ (that is, both snitches and fakers are unqualified), the construction achieves FRI and FA as the randomly sampled fake shares cannot be distinguished from the real shares. However, this does not achieve fake hiding, as the reconstruction will output $\perp$ when faking occurs. It follows from Theorem 4 that we could not hope for information-theoretic fake hiding in this setting.

C Omitted details from Section 5

C.1 Formal ACE security definitions

We restate the formal security definitions of ACE as described in [12]. Throughout the following definitions we will use Δ as the binary operator for the symmetric difference of two sets, i.e. $S_0 \Delta S_1 = (S_1 \setminus S_0) \cup (S_0 \setminus S_1)$.

Definition 15 (Asymmetrically Constrained Encryption [11]). *An asymmetrically constrained encryption (ACE) scheme consists of the tuple of algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{GenEK}, \mathsf{GenDK}, \mathsf{Enc}, \mathsf{Dec})$ *with the syntax:*

Setup $\mathsf{Setup}(1^\lambda, 1^n, 1^s) \to SK$*, is a randomised algorithm taking unary arguments security parameter* λ*, the message length* n*, and circuit succinctness* s*. It outputs a secret key* SK *for which encryption and decryption keys may be derrived from. We let the message space* $\mathcal{M} = \{0,1\}^n$*.*

Constrained Key Generation *Let* S *be a set* $S \subset \mathcal{M}$ *whose membership is deciable by a circuit* C_S*. The set is considered* admissible *if* $|C_S| \leq s$ *where* s *is the succinctness parameter from* Setup*. The succinctness parameter restricts the complexity of the sets for which keys may be constrianed.*

$\mathsf{GenEK}(SK, C_S) \to EK\{S\}$ *Given the secret key* SK *and a circuit* C_S *for an admissible set* S*, outputs a constrained encryption key* $EK\{S\}$*.*

$\mathsf{GenDK}(SK, C_S) \to DK\{S\}$ *Given the secret key* SK *and a circuit* C_S *for an admissible set* S*, outputs a constrained decryption key* $DK\{S\}$*.*

When keys are constrained for the empty set we will simply write EK, DK.

Encryption $\mathsf{Enc}(EK', m) \to c/\perp$*, A deterministic algorithm taking a possibly constrained key EK' and message $m \in \mathcal{M}$, outputting a ciphertext c or reject $\perp$.*

Decryption $\mathsf{Dec}(DK', c) \to m/\perp$*, A deterministic algorithm taking a possibly constrained key DK' and ciphertext c, outputting a message $m \in \mathcal{M}$ or reject $\perp$.*

An ACE scheme has the security properties, Correctness (Appendix C.1.1) Security of Constrained Decryption (Appendix C.1.2), and Selective Indistinguishability of Ciphertexts (Appendix C.1.3).

C.1.1 Correctness of ACE [12] A correct ACE scheme satisfies,

(a) *Correctness of Decryption:* For all sets $S, S' \subset \mathcal{M}$ and messages $m \in \mathcal{M}$, $m \notin S \cup S'$,

$$\Pr\left[SK \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{Dec}(\mathsf{GenDK}(SK, C_S), \mathsf{Enc}(\mathsf{GenEK}(SK, C_{S'}), m)) = m\right] = 1$$

(b) *Equivalence of constrained encryption:* For all sets $S, S' \subset \mathcal{M}$ and messages $m \in \mathcal{M}$, $m \notin S \Delta S'$,

$$\Pr\left[SK \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{Enc}(\mathsf{GenEK}(SK, C_S), m) = \mathsf{Enc}(\mathsf{GenEK}(SK, C_\emptyset), m)\right] = 1.$$

(c) *Unique Ciphertexts* With overwhelming probability over $SK \leftarrow \mathsf{Setup}(1^\lambda)$ and $s \subset \mathcal{M}$ for $DK = \mathsf{GenDK}(SK, C_S)$ it holds for c, c' that if $\mathsf{Dec}(DK, c) = \mathsf{Dec}(DK, c') \neq \perp$ then $c = c'$.

(d) *Safety of Constrained Decryption:* For all c, and all $S \subset \mathcal{M}$,

$$\Pr\left[SK \leftarrow \mathsf{Setup}(1^\lambda); DK = \mathsf{GenDK}(SK, C_S) : \mathsf{Dec}(DK, c) \in S\right] = 0$$

(e) *Equivalence of Constrained Decryption:* If $\mathsf{Dec}(DK\{S\}, c) = m \neq \perp$ and $m \notin S$ then $\mathsf{Dec}(DK\{S'\}, c) = m$.

C.1.2 Security of Constrained Decryption for ACE [12] For all PPT adversaries $\mathcal{A}$, the adversary initially outputs circuits (C_{S_0}, C_{S_1}, U) to the challenger, specifying sets $S_0, S_1, U \leftarrow \mathcal{A}$ subject to $S_0 \Delta S_1 \subseteq U \subseteq \mathcal{M}$.

The adversary also provides polynomially many messages $m_1, \ldots, m_t$ where $m_i \notin S_0 \Delta S_1$. The challenger samples $b \leftarrow \{0, 1\}$ and performs the following,

- $SK \leftarrow \mathsf{Setup}(1^\lambda)$
- $DK\{S_b\} \leftarrow \mathsf{GenDK}(SK, C_{S_b})$
- $EK \leftarrow \mathsf{GenEK}(SK, C_\emptyset)$
- For every $i \in [t], c_i \leftarrow \mathsf{Enc}(EK, m_i)$
- $EK\{U\} \leftarrow \mathsf{GenEK}(SK, C_U)$

The challenger then sends $(EK\{U\}, DK\{S_b\}, \{c_i\}_{i \in [t]})$. Finally, the adversary outputs $b' \in \{0, 1\}$. Let $\mathbf{Adv}_\mathcal{A} = |\Pr[b - b'] - 1/2|$, we require $\mathbf{Adv}_\mathcal{A} = |S_1 \setminus S_0| \cdot \mathsf{negl}(\lambda)$. That is an adversary will have negligible advantage for all polynomial $|S_1 \setminus S_0|$.

C.1.3 Selective Indistinguishability of Ciphertexts for ACE [12] For all $S, U \subseteq \mathcal{M}$, for all $m_0^*, m_1^* \in S \cap U$ and all $m_1, \ldots, m_t \in \mathcal{M} \setminus \{m_0^*, m_1^*\}$, the distrbution

$$EK\{S\}, DK\{U\}, c_0^*, c_1^*, c_1, \ldots c_t$$

is indistinguishable from

$$EK\{S\}, DK\{U\}, c_1^*, c_0^*, c_1, \ldots c_t$$

with randomness taken over $SK \leftarrow \mathsf{Setup}(1^\lambda); EK \leftarrow \mathsf{GenEK}(SK, C_S); DK\{U\} \leftarrow \mathsf{GenDK}(SK, C_U)$, giving ciphertexts $c_b^* \leftarrow \mathsf{Enc}(EK, m_b^*)$ and $c_i \leftarrow \mathsf{Enc}(EK, m_i)$.

C.2 Security proof for Theorem 8

Proof. We will prove that our construction is t_{fake}-Fake Hiding (Definition 9). Our proof proceeds through a series of hybrids, modifying the case when faking occurs ($b = \mathsf{fake}$) until it is identical to the case where all shares are real ($b = \mathsf{real}$).

HYBRID-0. When $b = \mathsf{fake}$, the original sharing is run for message $m^* = m_0$ using randomness ρ^*. The adversary is given ciphertexts

$$\{s_i^* = \mathsf{Enc}_{EK}((\tau^*, m^*, 0_{\tau^*}, i))\}_{i \in \mathsf{snitches}} \quad \{s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*} + 1, i))\}_{i \in \mathsf{fakers}}$$

where $\tau^* = \mathsf{G}_{K_1}(\rho^*)$ and $0_{\tau^*} = \mathsf{H}_{K_2}(\tau^*)$.

HYBRID-1. Add a check making Share return $\perp$ if $\tau = \tau^* = \mathsf{G}_{K_1}(m^*, \rho^*)$. Note this changes the functionality of the program, but only when sharing with randomness ρ^* as G is injective.

$\mathsf{Share}(m; \rho)$

1 : $\tau \leftarrow \mathsf{G}_{K_1}(\rho)$
2 : **if** $\tau = \tau^*$ **then return** $\perp$
3 : $0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$
4 : **for** $i \in [n]$:
5 : $\quad s_i \leftarrow \mathsf{Enc}_{K_2}((\tau, m, 0, i))$
6 : **return** $\{(i, s_i)\}_{i \in [n]}$

HYBRID-0 $\approx$ HYBRID-1: As G is injective, then only the inputs $\{(m; \rho^*)\}_{m \in \mathcal{M}}$ to s which will cause $\tau = \tau^* = \mathsf{G}(\rho^*)$. That is there are exactly $|\mathcal{M}| = \mathsf{poly}(\lambda)$ inputs for which the programs differ between the hybrids.

By Theorem 7 ([8, Theorem 6.2]), any distinguisher of HYBRID-0 and HYBRID-1 with polynomial advantage implies an efficient extractor finding a point at which the programs differ. This extractor runs in time polynomial in the circuit size of the programs, and the number of differing inputs, $|\mathcal{M}| = \mathsf{poly}(\lambda)$.

Given an extractor finding the point at which Share differs between HYBRID-0 and HYBRID-1, we may construct an adversary which breaks the one-wayness

of G. Observe, for HYBRID-1 the challenger does not need to know what the preimage of τ^* is to provide the programs and ciphertexts to the adversary, as the ciphertexts are,

$$\{\mathsf{Enc}_{EK}((\tau^*, m^*, 0_{\tau^*}, i))\}_{i\in\mathsf{snitches}}, \quad \{\mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}+1, i))\}_{i\in\mathsf{fakers}}.$$

We may set τ^* to be the challenge of the one-wayness game for G (Definition 14), i.e. $\tau^* = \mathsf{G}(\hat{\rho})$ where $\hat{\rho} \leftarrow \mathcal{R}$. This is distributed identically to HYBRID-1. An extractor successfully finding where this modified Share differs from that of HYBRID-0 clearly wins the one-wayness game. This follows as it finds $(m'; \rho')$ where $\mathsf{G}(\rho') = \tau^*$. Thus if G is an injective one-way function and iO is an indistinguishability obfuscator, these hybrids are indistinguishable.

HYBRID-2. At this point we are sure that the adversary may never independently produce shares with session identifier τ^* and level 0_{τ^*}. We proceed to constrain the encryption key in the Share and Fake programs for the set

$$S = \{(\tau^*, m, 0_{\tau^*}, i)\}_{i\in[n], m\in\mathcal{M}}.$$

If the message space $\mathcal{M}$ is of polynomial (or constant) size then S will also be polynomially bounded.

$\mathsf{Share}(m;\rho)$	$\mathsf{Fake}(\{(i,s_i)\}_{i\in S}, m^{fake})$
$\tau \leftarrow \mathsf{G}_{K_1}(\rho)$	$(\tau, m, \ell, \mathsf{partyset}') \leftarrow \mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$
if $\tau = \tau^*$ **then return** $\perp$	**if** $\lvert\mathsf{partyset}'\rvert \le t_{\mathsf{fake}}$ **then return** $\perp$
$0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$	**for** $i \in \mathsf{partyset}'$:
for $i \in [n]$:	$s'_i \leftarrow \mathsf{Enc}_{EK\{S\}}((\tau_j, m^{fake}, \ell+1, i))$
$s_i \leftarrow \mathsf{Enc}_{EK\{S\}}((\tau, m, 0, i))$	**return** $\{(i, s'_i)\}_{i\in\mathsf{partyset}'}$
return $\{(i,s_i)\}_{i\in[n]}$	

HYBRID-1 $\approx$ HYBRID-2: Due to the check introduced back in HYBRID-1 we may be sure that Share never produces an encryption of any element in S, as all encryptions are of $(\tau, m, 0_{\tau^*}, i)$ for $\tau \neq \tau^*$.

For input shares to Fake with session identifier τ^*, the $\mathsf{Decrypt}$ sub-circuit will ensure that all levels are in the interval $[0_{\tau^*}, T]$. If ℓ is the maximal level of the input shares, then Fake will always produce encryptions of level $\ell+1$. It follows that any encryptions of (τ^*, m, ℓ', i) will have $\ell' \in [0_{\tau^*}+1, T+1]$. We may conclude that Fake never encrypts an element of S.

As neither Share nor Fake ever encrypts an element of S the hybrids are indistinguishable by equivalence of constrained encryption (Section C.1.1) and the security of iO.

HYBRID-3. Our next step is to ensure that the adversary cannot make the programs decrypt ciphertexts with session identifier τ^* of level 0_{τ^*}, other than those provided to it by parties in $\mathsf{snitches}$. To do so constrain decryption for

$$S' = S \setminus \{(\tau^*, m^*, 0_{\tau^*}, i)\}_{i\in\mathsf{snitches}}.$$

Note, we cannot yet constrain decryption for the entire set S, as we need to produce ciphertexts for the messages in $\{(\tau^*, m^*, 0_{\tau^*}, i)\}_{i\in\mathsf{snitches}}$ in form of the shares given to the adversary.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $\quad p \leftarrow \mathsf{Dec}_{DK\{S'\}}(s_i)$
$\quad\quad\vdots$

Hybrid-2 $\approx$ Hybrid-3: In Hybrid-2 we constrained the encryption key for the set S. Security of constrained decryption allows changing a decryption key from $DK\{S_0\}$ to $DK\{S_1\}$ under the condition that $S_0 \Delta S_1 \subseteq S$ where S is the set the encryption key is constrained for. In our case $S_0 = \emptyset$ and $S_1 = S'$, satisfying $\emptyset \Delta S' = S' \subseteq S$. Crucially the messages $m_i \in \{(\tau^*, m^*, 0_{\tau^*}, i)\}_{i\in\mathsf{snitches}}$, which are encrypted to produce shares s_i^* for $i \in \mathsf{fakers}$ are not included in $S_0 \Delta S_1 = S'$. Indistinguishability follows by the security of constrained decryption, see Section C.1.2.

Hybrid-4. We wish to constrain decryption further, to include the plaintexts in $\{(\tau^*, m^*, 0_{\tau^*}, i)\}_{i\in\mathsf{snitches}}$. To achieve this we must first hardcode the decryption for the shares of level 0_{τ^*} which the adversary has received: $U = \{s_i^*\}_{i\in\mathsf{snitches}}$.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i\in\mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $\quad$ **if** $\exists j \in \mathsf{snitches}, s_i = s_j^* \in U$ **then** $p \leftarrow (\tau^*, m^{real}, 0, j)$
4 : $\quad$ **else** $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
$\quad\quad\vdots$

Hybrid-2 $\approx$ Hybrid-4: By correctness of decryption of ACE, hardcoding the decryption of ciphertexts in U will preserve the functionality. Moreover, due to unique ciphertexts (Section C.1.1), with overwhelming probability over the choice of DK no ciphertexts other than those in U will decrypt to a plaintext in $\{(\tau^*, m^*, 0_{\tau^*}, i)\}_{i\in\mathsf{snitches}}$. By equivalence of constrained decryption (Section C.1.1) it follows that constraining DK by S rather than S' will preserve functionality with overwhelming probability. With functionality preserved, we may apply iO security to conclude indistinguishability of the hybrids.

Hybrid-5. Further, constrain encryption during faking and sharing for the set

$$Z = \{(\tau^*, m, 0_{\tau^*} + 1, i)\}_{i\in[n], m\in\mathcal{M}} \setminus \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i\in\mathsf{fakers}}.$$

Giving,

$\mathsf{Share}(m;\rho)$	$\mathsf{Fake}(\{(i,s_i)\}_{i\in S}, m^{fake})$
$\tau \leftarrow \mathsf{G}_{K_1}(\rho)$	$(\tau, m, \ell, \mathsf{partyset}') \leftarrow$
if $\tau = \tau^*$ **then return** $\perp$	$\quad \mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$
$0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$	**if** $\lvert\mathsf{partyset}'\rvert \le t_{\mathsf{fake}}$ **then return** $\perp$
for $i \in [n]$:	**for** $i \in \mathsf{partyset}'$:
$\quad s_i \leftarrow \mathsf{Enc}_{EK\{S\cup Z\}}((\tau, m, 0, i))$	$\quad s_i' \leftarrow \mathsf{Enc}_{EK\{S\cup Z\}}((\tau_j, m^{fake}, \ell+1, i))$
return $\{(i,s_i)\}_{i\in[n]}$	**return** $\{(i,s_i')\}_{i\in\mathsf{partyset}'}$

Note Z will always be a set of messages in the plaintext space, as all levels in $[0, T+1]$ are permitted,

$$(\tau, m, \ell, i) \in \left(\{0,1\}^{\tau(\lambda)} \times \mathcal{M} \times [0, T+1] \times [n]\right).$$

HYBRID-4 $\approx$ HYBRID-5: The functionality of Share is unchanged, by equivalence of constrained encryption, as it would never encrypt any plaintext in Z.

We may now consider Fake. After constraining decryption in $\mathsf{Decrypt}$, the elements of

$$U = \{s_i^* = \mathsf{Enc}_{EK}((\tau^*, m^*, 0_{\tau^*}, i))\}_{i\in\mathsf{snitches}}$$

are the only ciphertexts with session identifier τ^* and level 0_{τ^*} which will successfully decrypt. By assumption, we know $\lvert\mathsf{snitches}\rvert \le t_{\mathsf{fake}}$, therefore for $\tau = \tau^*$, the level ℓ cannot be 0_{τ^*} while $\lvert\mathsf{partyset}'\rvert > t_{\mathsf{fake}}$ holds. Therefore, Fake will never encrypt an element of Z, allowing its functionality to be preserved by equivalence of constrained encryption (Section C.1.1). Indistinguishability simply follows by the security of iO as the functionalities of Share and Fake are unchanged.

HYBRID-6. Constrain decryption further from S to $S \cup Z$,

$\mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$
$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
$\quad$ **if** $\exists i \in \mathsf{partyset}_0, s_i = s_i^*$ **then** $p \leftarrow (\tau^*, m_1, 0_{\tau^*}, i)$
$\quad$ **else** $p \leftarrow \mathsf{Dec}_{DK\{S\cup Z\}}(s_i)$
$\vdots$

HYBRID-5 $\approx$ HYBRID-6: Indistinguishability follows by the security of constrained decryption (Section C.1.2). As required $S\Delta(S \cup Z) = Z \subseteq S \cup Z$. Furthermore, the messages for faker and snitch shares $\{(\tau^*, m_1, 0_{\tau^*}+1, i)\}_{i\in\mathsf{fakers}}$ are not contained in Z.

HYBRID-7. Modify the hardcoded decryption to be of level $0_{\tau^*}+1$ and change the message to be m_1.

Before.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  if $\exists j \in \mathsf{snitches}, s_i = s_j^*$ **then**
    $p \leftarrow (\tau^*, m^{real}, 0_{\tau^*}, j)$
  else $p \leftarrow \mathsf{Dec}_{DK\{S \cup Z\}}(s_i)$
⋮

After.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  if $\exists j \in \mathsf{snitches}, s_i = s_j^*$ **then**
    $p \leftarrow (\tau^*, m^{fake}, 0_{\tau^*} + 1, j)$
  else $p \leftarrow \mathsf{Dec}_{DK\{S \cup Z\}}(s_i)$
⋮

Hybrid-6 $\approx$ Hybrid-7: To employ the security of iO we must argue that the above modification has no effect on the functionality of the program. We may restrict ourselves to inputs to decrypt where at least one share $s_i = s_j^*$. If any share has a session identifier different from τ^* both Fake and Rec would output $\perp$ regardless of this change. Similarly, if any share is provided with the wrong party index $\perp$ will always be returned.

Assuming all shares have session identifier τ^* we may move on to considering the level of the shares. If any share has level greater than $0_{\tau^*} + 1$ changing the level and message when decrypting s_i^* will have no effect on functionality, as $\mathsf{Decrypt}$ only checks the messages for shares of the highest level.

This leaves us considering inputs where all shares are of level 0_{τ^*} or $0_{\tau^*} + 1$. Safety of constrained decryption tells us that decryption with $DK\{S \cup Z\}$ we will never output a plaintext in $S \cup Z$ (Section C.1.1). Prior to the changes in the hybrid the only shares which may decrypt of level 0_{τ^*} are $s_i^* \in U$, this follows as decryption is constrained for $S = \{(\tau^*, m, 0_{\tau^*}, i)\}_{i \in [n], m \in \mathcal{M}}$. As $|U| \leq t_{\mathsf{fake}}$ and $|U| \leq t$, any input containing only ciphertexts from U will result in Rec and Fake returning $\perp$. This is unaffected by the changes between these hybrids.

Finally, we are left with the case where shares $s_i^* \in U$ are mixed with shares of level $0_{\tau^*} + 1$. Recall, decryption is also constrained for

$$Z = \{(\tau^*, m, 0_{\tau^*} + 1, i)\}_{i \in [n], m \in \mathcal{M}} \setminus \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}.$$

This means that prior to the changes the only plaintexts which could decrypt of level $0_{\tau^*} + 1$ are in $W = \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}$. Let

$$V = \{(\tau^*, m_0, 0_{\tau^*}, i)\}_{i \in \mathsf{snitches}}, \qquad V' = \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{snitches}}.$$

Consider a set P of plaintexts drawn from $W \cup V$ and P' from $W \cup V'$ taking plaintexts with identical indices I, i.e. for $i \in I$ if $i \in \mathsf{snitches}$ then $(\tau^*, m_0, 0_{\tau^*}, i) \in P$. By inspection the functionality of $\mathsf{Decrypt}$ is identical between P and P', as it will return $(\tau^*, m_1, 0_{\tau^*} + 1, I)$ in either case. We may conclude that the functionality of the program between the two hybrids is identical.

Hybrid-8. One-by-one, replace the ciphertexts

$$s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_0, 0_{\tau^*}, i)) \quad \text{with} \quad \hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*} + 1, i))$$

for $i \in \mathsf{snitches}$. Ciphertexts are replaced in both the obfuscated programs (in Decrypt) and where they are given to the adversary directly.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : **if** $\exists i \in \mathsf{snitches}, s_i = \hat{s}_i$ **then** $p \leftarrow (\tau^*, m^{real}, 0_{\tau^*} + 1, i)$
4 : **else** $p \leftarrow \mathsf{Dec}_{DK\{S \cup Z\}}(s_i)$

$\vdots$

Hybrid-7 $\approx$ Hybrid-8: For selective indistinguishability of ciphertexts for ACE to apply (Section C.1.3) we require that both encryption and decryption are constrained for the plaintexts we wish to swap. Both encryption and decryption are constrained for the set $S \cup Z$. The plaintext $(\tau^*, m_0, 0_{\tau^*}, i)$ for s_i^* is in S, while the plaintext $(\tau^*, m_1, 0_{\tau^*} + 1, i)$ for $\hat{s}_i$ is in Z, implying $s_i^* \approx \hat{s}_i$.

Hybrid-9. In Decrypt change decryption key from being constrained by $S \cup Z$ to $S \cup Z'$ for $Z' = Z \setminus \{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}$.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : **if** $\exists i \in \mathsf{snitches}, s_i = \hat{s}_i$ **then** $p \leftarrow (\tau^*, m^{real}, 0_{\tau^*} + 1, i)$
4 : **else** $p \leftarrow \mathsf{Dec}_{DK\{S \cup Z'\}}(s_i)$

$\vdots$

Hybrid-8 $\approx$ Hybrid-9: The set $\{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}$ being removed from puncturing is exactly the set of plaintexts for the ciphertexts $\hat{s}_i$ for $i \in \mathsf{fakers}$. With overwhelming probability over the choice of secret key the ciphertexts $\hat{s}_i$ for $i \in \mathsf{fakers}$ are the only ciphertexts for the plaintexts in $\{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}$. In the case where ciphertexts are unique changing Z to Z' does not affect the functionality of Decrypt as no ciphertexts distinct from $\hat{s}_i$ for $i \in \mathsf{fakers}$ will decrypt to anything in $\{(\tau^*, m_1, 0_{\tau^*} + 1, i)\}_{i \in \mathsf{fakers}}$. Indistinguishability follows be the security of iO.

Hybrid-10. Reduce the constraints on the decryption key, going from $S \cup Z'$ to S.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : **if** $\exists i \in \mathsf{snitches}, s_i = \hat{s}_i$ **then** $p \leftarrow (\tau^*, m_0, 0_{\tau^*} + 1, i)$
4 : **else** $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$

$\vdots$

Hybrid-9 $\approx$ Hybrid-10: Indistinguishability follows by the security of constrained decryption (Section C.1.2). Encryption is constrained for $S \cup Z$, satisfying $(S \cup Z')\Delta S = Z' \subseteq S \cup Z$. Furthermore, none of the ciphertexts provided as shares to the adversary have plaintexts in $Z' = (S \cup Z')\Delta S$.

Hybrid-11. Remove the special case for $\hat{s}_i$ where $i \in \mathsf{snitches}$, allowing these to be decrypted normally,

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$

$\vdots$

Hybrid-10 $\approx$ Hybrid-11: By correctness of decryption and equivalence of constrained decryption (Section C.1.1) the ciphertexts $\hat{s}_i$ will be correctly decrypted after this modification. As such the functionality of $\mathsf{Decrypt}$ is unchanged, with indistinguishability following by the security of iO.

Hybrid-12. Puncture the key for PRF H on input τ^*, and hardcoding this output in Share and $\mathsf{Decrypt}$. Let $0_{\tau^*} = \mathsf{H}_{K_2}(\tau^*)$.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
  if $p = \bot$ **then continue**
  parse $(\tau_i, m_i, \ell_i, j) := p$
  if $\tau_i = \tau^*$
    if $j \neq i \vee \ell_i \notin [0_{\tau^*}, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
  else
    $0_{\tau_i} \leftarrow \mathsf{H}_{K_2\{\tau^*\}}(\tau_i)$
    if $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
$j \leftarrow \mathsf{argmax}_{i \in \mathsf{partyset}'}\ \ell_i$
⋮

$\mathsf{Share}(m; \rho)$

$\tau \leftarrow \mathsf{G}_{K_1}(\rho)$
if $\tau = \tau^*$ **then return** $\bot$
$0_\tau \leftarrow \mathsf{H}_{K_2\{\tau^*\}}(\tau)$
for $i \in [n]$:
  $s_i \leftarrow \mathsf{Enc}_{K_2}((\tau, m, 0, i))$
return $\{(i, s_i)\}_{i \in [n]}$

HYBRID-11 $\approx$ HYBRID-12: The functionality of s is unchanged as then functionality of H is preserved under puncturing (Definition 13) and H is never evaluated on τ^*. The functionality of $\mathsf{Decrypt}$ is similarly identical as functionality is preserved under puncturing and $0_{\tau^*} = \mathsf{H}_{K_2}(\tau^*)$. Indistinguishability follows by the security of iO.

HYBRID-13. Replace 0_{τ^*} by randomly sampled $\ell_0 \leftarrow [0, T]$, both in $\mathsf{Decrypt}$ and the ciphertexts (shares) provided to the adversary.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
  if $p = \perp$ **then continue**
  parse $(\tau_i, m_i, \ell_i, j) := p$
  if $\tau_i = \tau^*$
    if $j \neq i \vee \ell_i \notin [\ell_0, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
  else
    $0_{\tau_i} \leftarrow \mathsf{H}_{K_2\{\tau^*\}}(\tau_i)$
    if $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
$j \leftarrow \mathsf{argmax}_{i \in \mathsf{partyset}'}\ \ell_i$
⋮

$$\{\hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_0, \ell_0 + 1, i))\}_{i \in \mathsf{snitches}},$$

$$\{s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_0, \ell_0 + 1, i))\}_{i \in \mathsf{fakers}},$$

$$S = \{(\tau^*, m, \ell_0, i)\}_{i \in [n], m \in \mathcal{M}}.$$

HYBRID-12 $\approx$ HYBRID-13: Pseudorandomness at punctured points of H implies that these hybrids are indistinguishable.

HYBRID-14. For $\tau_i = \tau^*$ increase the lower bound of the accepted level range by one.

$\mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
  if $p = \perp$ **then continue**
  parse $(\tau_i, m_i, \ell_i, j) := p$
  if $\tau_i = \tau^*$
    if $j \neq i \vee \ell_i \notin [\ell_0{+}1, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
  else
    $0_{\tau_i} \leftarrow \mathsf{H}_{K_2\{\tau^*\}}(\tau_i)$
    if $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
$j \leftarrow \mathsf{argmax}_{i\in\mathsf{partyset}'}\ \ell_i$

$\vdots$

Hybrid-13 $\approx$ Hybrid-14: We argue this does not affect the functionality of Decrypt. Recall,

$$S = \{(\tau^*, m, \ell_0, i)\}_{i\in[n], m\in\mathcal{M}}.$$

As the decryption key is punctured by S, no ciphertext can decrypt to a plaintext of the form (τ^*, m, ℓ_0, i), allowing $\ell_i \notin [\ell_0, T]$ to be changed to $\ell_i \notin [\ell_0{+}1, T]$ without affecting the functionality. Indistinguishability follows by the security of iO.

Hybrid-15. Let $\ell'_0 = \ell_0 + 1$, observe ℓ'_0 is uniform in $[1, T+1]$. Replace ℓ'_0 by $\ell''_0 \leftarrow [0, T]$.

Before.

$\mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
  if $p = \perp$ **then continue**
  parse $(\tau_i, m_i, \ell_i, j) := p$
  if $\tau_i = \tau^*$
    if $j \neq i \vee \ell_i \notin [\ell'_0, T]$ **then**
      continue
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
  else

$\vdots$

$\{\hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, \ell'_0, i))\}_{i\in\mathsf{snitches}}$
$\{s^*_i = \mathsf{Enc}_{EK}((\tau^*, m_1, \ell'_0, i))\}_{i\in\mathsf{fakers}}$
$S = \{(\tau^*, m, \ell'_0 - 1, i)\}_{i\in[n], m\in\mathcal{M}}$

After.

$\mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$

$\mathsf{partyset}' \leftarrow \emptyset$
for $i \in \mathsf{partyset}$
  $p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
  if $p = \perp$ **then continue**
  parse $(\tau_i, m_i, \ell_i, j) := p$
  if $\tau_i = \tau^*$
    if $j \neq i \vee \ell_i \notin [\ell''_0, T]$ **then**
      continue
    $\mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
  else

$\vdots$

$\{\hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, \ell''_0, i))\}_{i\in\mathsf{snitches}}$
$\{s^*_i = \mathsf{Enc}_{EK}((\tau^*, m_1, \ell''_0, i))\}_{i\in\mathsf{fakers}}$
$S = \{(\tau^*, m, \ell''_0 - 1, i)\}_{i\in[n], m\in\mathcal{M}}$

HYBRID-14 $\approx$ HYBRID-15: For T an exponentially large in the security parameter the distributions of ℓ'_0 and ℓ''_0 are statistically close and therefore indistinguishable.

HYBRID-16. Restore the PRF output $0_{\tau^*} = \mathsf{H}_{K_2}(\tau^*)$ in place of ℓ''_0 in the program and ciphertexts.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $\quad p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
4 : $\quad$ **if** $p = \perp$ **then continue**
5 : $\quad$ **parse** $(\tau_i, m_i, \ell_i, j) := p$
6 : $\quad$ **if** $\tau_i = \tau^*$
7 : $\quad\quad$ **if** $j \neq i \vee \ell_i \notin [0_{\tau^*}, T]$ **then continue**
8 : $\quad\quad \mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
9 : $\quad$ **else**
10 : $\quad\quad 0_{\tau_i} \leftarrow \mathsf{H}_{K_2\{\tau^*\}}(\tau_i)$
11 : $\quad\quad$ **if** $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
12 : $\quad\quad \mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
13 : $j \leftarrow \mathsf{argmax}_{i \in \mathsf{partyset}'}\ \ell_i$

$$\vdots$$

$$\{\hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}, i))\}_{i \in \mathsf{snitches}}$$
$$\{s^*_i = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}, i))\}_{i \in \mathsf{fakers}}$$
$$S = \{(\tau^*, m, 0_{\tau^*} - 1, i)\}_{i \in [n], m \in \mathcal{M}}$$

HYBRID-15 $\approx$ HYBRID-16: Indistinguishability follows by the pseudorandomness of H at punctured points (Definition 13).

HYBRID-17. Restore the key for H to its unpunctured state, removing the seperate branch for τ^*.

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $\quad p \leftarrow \mathsf{Dec}_{DK\{S\}}(s_i)$
4 : $\quad$ **if** $p = \bot$ **then continue**
5 : $\quad$ **parse** $(\tau_i, m_i, \ell_i, j) := p$
6 : $\quad 0_{\tau_i} \leftarrow \mathsf{H}_{K_2}(\tau_i)$
7 : $\quad$ **if** $j \neq i \vee \ell_i \notin [0_{\tau_i}, T]$ **then continue**
8 : $\quad \mathsf{partyset}' \leftarrow \mathsf{partyset}' \cup \{i\}$
9 : $j \leftarrow \mathsf{argmax}_{i \in \mathsf{partyset}'}\ \ell_i$

$\vdots$

HYBRID-16 $\approx$ HYBRID-17: These changes do not affect the functionality of Decrypt as the functionality of H is preserved under puncturing. Indistinguishability follows from the security of iO.

HYBRID-18. Remove constraints on the decryption key for the set

$$S = \{(\tau^*, m, 0_{\tau^*} - 1, i)\}_{i \in [n], m \in \mathcal{M}}.$$

$\mathsf{Decrypt}(\{(i, s_i)\}_{i \in \mathsf{partyset}})$

1 : $\mathsf{partyset}' \leftarrow \emptyset$
2 : **for** $i \in \mathsf{partyset}$
3 : $\quad p \leftarrow \mathsf{Dec}_{DK}(s_i)$
4 : $\quad$ **if** $p = \bot$ **then continue**

$\vdots$

HYBRID-17 $\approx$ HYBRID-18: Indistinguishability follows by the security of constrained decryption, as encryption is constrained for the set S and no ciphertexts with plaintexts in S are given to the adversary.

HYBRID-19. Remove constraints on the encryption key for the set

$$S = \{(\tau^*, m, 0_{\tau^*} - 1, i)\}_{i \in [n], m \in \mathcal{M}}.$$

$\mathsf{Share}(m;\rho)$	$\mathsf{Fake}(\{(i,s_i)\}_{i\in S}, m^{fake})$
$\tau \leftarrow \mathsf{G}_{K_1}(\rho)$	$(\tau, m, \ell, \mathsf{partyset}') \leftarrow \mathsf{Decrypt}(\{(i,s_i)\}_{i\in\mathsf{partyset}})$
if $\tau = \tau^*$ **then return** $\perp$	**if** $\lvert\mathsf{partyset}'\rvert \le t_{\mathsf{fake}}$ **then return** $\perp$
$0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$	**for** $i \in \mathsf{partyset}'$:
for $i \in [n]$:	$s_i' \leftarrow \mathsf{Enc}_{EK}((\tau_j, m^{fake}, \ell+1, i))$
$s_i \leftarrow \mathsf{Enc}_{EK}((\tau, m, 0_\tau, i))$	**return** $\{(i,s_i')\}_{i\in\mathsf{partyset}'}$
return $\{(i,s_i)\}_{i\in[n]}$	

Hybrid-18 $\approx$ Hybrid-19: The functionality of Share is clearly unchanged as no plaintext in S is ever encrypted. The same holds true for Fake as the level ℓ given by Decrypt for a session identifier τ must be in the range $[0_\tau, T]$. Indistinguishability then follows from the security of iO.

Hybrid-20. Remove the check for $\tau = \tau^*$.

$\mathsf{Share}(m;\rho)$
$\tau \leftarrow \mathsf{G}_{K_1}(\rho)$
~~**if** $\tau = \tau^*$ **then return** $\perp$~~
$0_\tau \leftarrow \mathsf{H}_{K_2}(\tau)$
for $i \in [n]$:
$s_i \leftarrow \mathsf{Enc}_{K_2}((\tau, m, 0_\tau, i))$
return $\{(i,s_i)\}_{i\in[n]}$

Having restored the programs and transformed the ciphertexts we are finally left in the world with where no faking occurs,

$$\{\hat{s}_i = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}, i))\}_{i\in\mathsf{snitches}}$$
$$\{s_i^* = \mathsf{Enc}_{EK}((\tau^*, m_1, 0_{\tau^*}, i))\}_{i\in\mathsf{fakers}}.$$

Hybrid-19 $\approx$ Hybrid-20: Indistinguishability follows by the same reasoning as for Hybrid-0 $\approx$ Hybrid-1. When G is injective the programs differ at $|\mathcal{M}|$ points, a distinguisher with polynomial advantage allows efficiently finding one of these points, breaking the one-wayness of G. Using Theorem 7 the hybrids are indistinguishable if G is an injective one-way function and iO is an indistinguishability obfuscator. □

D Random Faking

In this section, we show a few simple deniable secret sharing schemes with random faking. In Section D.1 and Section D.2, we consider the setting where fakers don't coordinate. In Section D.3 we consider the coordinated setting.

The notion of random faking is defined with respect to a distribution dist. We consider dist to be the uniform distribution; messages from other distributions can be encoded as uniform elements.

D.1 Uncoordinated Random Faking: Full-Threshold Construction

Consider the following construction for $t = n$, which uses additive secret sharing i.e. an n-out-of-n secret sharing scheme over any finite group, such as $\mathbb{Z}_q$ (set of integers modulo q).

$\mathsf{Share}(m)$**:** Additively share $m \in \mathbb{Z}_q$ to get $s_1, \ldots, s_n$, where the shares are sampled uniformly at random subject to $\sum_{i=1}^{n} s_i \mod q = m$. Return $\{(i, s_i)\}_{i \in [n]}$.

$\mathsf{Fake}(s_i)$**:** Choose a random value ρ_i, and let $s'_i = s_i + \rho_i \mod q$. Return s'_i.

$\mathsf{Rec}(s_1, \ldots, s_n)$**:** Return $m' = \sum_{i \in \{1,\ldots,n\}} s_i \mod q$.

This construction clearly offers correctness and privacy (Definition 1), as well as all of fake-real indistinguishability (Definition 3), faker anonymity (Definition 7), and even fake-hiding (Definition 9). This is because the uniform distribution of shares and the full threshold ensures that there is no way for the adversary to be able to distinguish fake shares from the real ones. However, this approach works only for full threshold and it is not clear how to modify it for $t < n$.

D.2 Uncoordinated Random Faking: Lower Bounds

In the following theorem we prove that schemes with uncoordinated random faking cannot achieve FRI for unqualified sets. We specifically prove this for thresholds $t \leq n - 2$, in this case the smallest n where $[n]$ can be partitioned into two unqualified sets is 6, for $t = 4$.

Theorem 10 (Lower Bound on FRI in the Uncoordinated Setting When Neither Set Has t Parties). *Say we have a scheme with uncoordinated random faking with $n \geq 6$ shareholders, which requires $(\emptyset, [n])$-FRI (Definition 2; if* everyone *fakes, the set of fake shares should be indistinguishable from a fresh sharing) and has a privacy threshold $t \leq n-2$. Consider some $k < t$, then the scheme cannot have $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI for all partitions $\mathsf{partyset}_0, \mathsf{partyset}_1$ where $|\mathsf{partyset}_0| = k$.*

Proof. Assume for contradiction that the scheme has $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI for all partitions $\mathsf{partyset}_0, \mathsf{partyset}_1$ where $|\mathsf{partyset}_0| = k$.

Consider some specific partition $\mathsf{partyset}_0, \mathsf{partyset}_1$ where $|\mathsf{partyset}_0| = k$. First, partition $\mathsf{partyset}_0$ further into $\mathsf{partyset}_0^0$ and $\mathsf{partyset}_0^1$, s.t. $|\mathsf{partyset}_0^0| = 1$ and $|\mathsf{partyset}_0^0| = k-1$. Then partition $\mathsf{partyset}_1$ into four sets, $\mathsf{partyset}_1^0, \mathsf{partyset}_1^1$, $\mathsf{partyset}_1^2, \mathsf{partyset}_1^3$,

$$|\mathsf{partyset}_1^0| = t - k, \quad |\mathsf{partyset}_1^1| = |\mathsf{partyset}_1^2| = 1, \quad |\mathsf{partyset}_1^3| = n - t - 2.$$

Observe, the sets

$$S_1 = \mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^0 \cup \mathsf{partyset}_1^1 \quad \text{and} \quad S_2 = \mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^0 \cup \mathsf{partyset}_1^2$$

are qualified, while may be $\mathsf{partyset}_1^3$ empty. In our matrix notation we now consider columns $\mathsf{partyset}_0^0, \mathsf{partyset}_0^1$ followed by $\mathsf{partyset}_1^0, \mathsf{partyset}_1^1, \mathsf{partyset}_1^2, \mathsf{partyset}_1^3$, from left to right. By FRI we have,

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f} \\ \mathsf{f}\,\mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \end{bmatrix}.$$

Similarly, for $S = \mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^1$ and $R = \mathsf{partyset}_0^0 \cup \mathsf{partyset}_1^0 \cup \mathsf{partyset}_1^2 \cup \mathsf{partyset}_1^3$,

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\quad\mathsf{f} \\ \mathsf{f}\quad\mathsf{f} \end{bmatrix} \overbrace{\quad}^{(S,R)\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \end{bmatrix}.$$

Let us study the shares

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\quad\mathsf{f} \\ \mathsf{f}\quad\mathsf{f} \end{bmatrix}$$

more closely. As they are indistinguishable from a fresh sharing, all subsets of size t must reconstruct to the same message. In particular, reconstructing from the sets S_1 and S_2, must give the same message. To reach our contradiction, consider the shares with additional faking for $\mathsf{partyset}_0^0$,

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f}\quad\mathsf{f} \\ \mathsf{f}\,\mathsf{f}\quad\mathsf{f} \end{bmatrix}.$$

Reconstructing from S_1 and S_1 must exhibit identical behaviour here, as $\mathsf{partyset}_0^0$ is disjoint from S_1 and S_2.

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f}\quad\mathsf{f} \\ \mathsf{f}\,\mathsf{f}\quad\mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f}\,\mathsf{f} \\ \mathsf{f} \\ \mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f} \\ \mathsf{f} \end{bmatrix}.$$

We had previously shown that the shares of S_1 and S_2 must reconstruct to the same message. Shown visually, the shares marked in blue for S_1 on the left and in blue for S_2 on the right

$$\begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f} \\ \mathsf{f} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r}\,\mathsf{r} \\ \mathsf{f} \\ \mathsf{f} \end{bmatrix}$$

must reconstruct to the same message. By correctness S_2 will allow recovering the original message. This contradicts privacy as S_1 contains fewer than t real shares. Formally, the adversary could recover the original message for the following shares,

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & & \mathsf{f} & \mathsf{f} & \mathsf{f} & \end{bmatrix}$$

given only $t - 1 = |\mathsf{partyset}_0^1 \cup \mathsf{partyset}_1^0|$ real shares, by faking for $\mathsf{partyset}_1^1$ and reconstructing for S_1. □

Theorem 11 (Lower Bound on FRI in the Uncoordinated Setting When One Set Has at Least t Parties). *Say we have a scheme with uncoordinated random faking, which requires* $(\emptyset, [n])$*-FRI (Definition 2; if* everyone *fakes, the set of fake shares should be indistinguishable from a fresh sharing) and has a privacy threshold* $t < n$*. Consider a specific partition of the parties into non-empty sets* $\mathsf{partyset}_0$ *and* $\mathsf{partyset}_1$ *such that* $|\mathsf{partyset}_1| \geq t$*. Then, the scheme does not have* $(\mathsf{partyset}_0, \mathsf{partyset}_1)$*-FRI.*

Proof. Partition $\mathsf{partyset}_1$ into $\mathsf{partyset}_1^0$ and $\mathsf{partyset}_1^0$, such that the parties in $\mathsf{partyset}_0 \cup \mathsf{partyset}_1^0$ are qualified, i.e. $|\mathsf{partyset}_0 \cup \mathsf{partyset}_1^0| = t$. We consider columns $\mathsf{partyset}_0$, $\mathsf{partyset}_1^0$ and $\mathsf{partyset}_1^0$ and assume $(\mathsf{partyset}_0, \mathsf{partyset}_1)$-FRI for contradiction,

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & & \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \\ & \mathsf{f} & \mathsf{f} \end{bmatrix}.$$

In the random faking case, the real message is sampled within the security game, and the fake message is chosen implicitly by the faking algorithm. We must therefore consider distributions of shares where the message is taken from some distribution itself. Observe the following,

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & & \\ \mathsf{f} & & \end{bmatrix} \overbrace{\quad}^{(\mathsf{partyset}_0, \mathsf{partyset}_1)\text{-FRI}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & \mathsf{f} & \mathsf{f} \end{bmatrix} \overbrace{\quad}^{(\emptyset, [n])\text{-FRI}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \end{bmatrix}.$$

In the case of a real sharing correctness requires that any subset of at least t shares reconstruct to the same message. By privacy this cannot be true for:

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & & \\ \mathsf{f} & & \end{bmatrix}.$$

Suppose we attempted to reconstruct from the set $\mathsf{partyset}_0 \cup \mathsf{partyset}_1^0$, by our reasoning above this should reconstruct to the original message with overwhelming probability. This poses a problem as faking should hide the message when the snitches (in this case $\mathsf{partyset}_0$) are unqualified. If the distribution is non-trivial, meaning there are at least two messages which occur with non-negligible

probability, this would allow the adversary a non-negligible advantage in distinguishing the distributions,

$$\begin{bmatrix} m_0 & m_0 & m_0 \\ \mathsf{f} & & \mathsf{f} \\ \mathsf{f} & & \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} m_1 & m_1 & m_1 \\ \mathsf{f} & & \mathsf{f} \\ \mathsf{f} & & \end{bmatrix},$$

contradicting privacy. □

Theorem 12. *Say we have a scheme with uncoordinated random faking, which requires* $(\emptyset, [n])$*-FRI (Definition 2) with threshold* $t \leq n-2$*. Then, the scheme does not have* $(n-1)$*-FA (Definition 5).*

Proof. Assume towards contradiction that the scheme has $(n-1)$-FA. Consider distinct indices $i, j, k \in [n]$. Let $\mathsf{partyset}_0 \subset [n] \setminus \{i, j, k\}$ be a set s.t. $|\mathsf{partyset}_0| = t-1$. For the remainder of this proof, our columns will be

$$\mathsf{partyset}_0, \{i\}, \{j\}, \{k\}, \mathsf{partyset}_1 = [n] \setminus (\{i, j, k\} \cup \mathsf{partyset}_0).$$

Note, $\mathsf{partyset}_1$ may be empty. The sets $\mathsf{partyset}_0 \cup \{i\}$ and $\mathsf{partyset}_0 \cup \{j\}$ are both qualified. By extreme FRI, if a fake is applied to every share, the resulting distribution must be indistinguishable from a fresh sharing, where all qualified subsets reconstruct to the same message.

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} \end{bmatrix} \overbrace{\qquad}^{(\emptyset,[n])\text{-FRI}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \end{bmatrix}.$$

We observe the following sequence of indistinguishabilities,

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} \\ & & \mathsf{f} & & \end{bmatrix} \overbrace{\qquad}^{([n]\setminus\{j\},[n]\setminus\{k\})\text{-FA}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} \\ & & \mathsf{f} & & \end{bmatrix} \overbrace{\qquad}^{(\emptyset,[n])\text{-FRI}} \begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ & & \mathsf{f} & & \end{bmatrix}.$$

In the leftmost case $\mathsf{partyset}_0 \cup \{i\}$ and $\mathsf{partyset}_0 \cup \{j\}$ must reconstruct to the same value. This contradicts privacy, as the shares of $\mathsf{partyset}_0 \cup \{j\}$ (columns shown in blue) must allow recovering the original message for the shares,

$$\begin{bmatrix} \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} & \mathsf{r} \\ & \mathsf{f} & \mathsf{f} & \mathsf{f} & \mathsf{f} \end{bmatrix}.$$

□

D.3 Coordinated Random Faking

With coordination, it is possible to get random faking with privacy, FRI and FA for all thresholds, and FH for all thresholds where FH is possible. Consider the following construction for arbitrary threshold t based on Shamir-secret sharing (say, over a finite field $\mathbb{F}_q$ of integers modulo q, with prime $q > n$).

$\mathsf{Share}(m)$**:** Computes $(s_1, \ldots, s_n) \leftarrow \mathsf{Shamir.Share}(m, t)$ as a shamir-sharing of the message $m \in \mathbb{F}_q$ with threshold t. Here, $\mathsf{Shamir.Share}$ involves sampling a polynomial $p(x)$ of degree $(t-1)$ such that $p(0) = m$ and setting $s_i = p(i) \mod q$. Return $\{(i, s_i)\}_{i \in [n]}$.

$\mathsf{Fake}(\{s_i\}_{i \in \mathsf{fakers}})$**:** If $n - |\mathsf{fakers}| \geq t$ (that is, the snitches are qualified), return the real shares $\{s_i\}_{i \in \mathsf{fakers}}$. Else,

1. Choose a random value ρ.
2. Choose a polynomial $p'(x)$ of degree $t-1$ such that $p'(0) = \rho$ and $p'(i) = 0$ for $i \in [n] \setminus \mathsf{fakers}$.
3. For $i \in \mathsf{fakers}$, let $s'_i = s_i + p'(i) \mod q$.

$\mathsf{Rec}(\{s_i\}_{i \in \mathsf{partyset}})$**:** Computes $m' \leftarrow \mathsf{Shamir.Rec}(\{s_i\}_{i \in \mathsf{partyset}}, t)$. Here, $\mathsf{Shamir.Rec}$ involves using Lagrange interpolation to identify a $(t-1)$-degree polynomial $p''(x)$ consistent with $\{s_i\}_{i \in \mathsf{partyset}}$. Return $p''(0)$ if such a polynomial exists and $\perp$ otherwise.

This construction clearly offers correctness and privacy which follows directly from the properties of Shamir secret sharing. Next, we observe that if snitches are qualified, similar to the construction in Theorem 9, the fakers gain nothing from altering their shares. In such a setting, while FH is impossible to achieve (Theorem 6), we obtain fake-real indistinguishability and faker anonymity by outputting original shares.

However, in the more interesting case where snitches are unqualified, this construction offers correctness, privacy, fake-real indistinguishability, faker anonymity and fake-hiding (when possible). This is because the fakers' fake shares together with the snitches' real shares constitutes a valid sharing of $m + \rho$. Notably, this construction works even if the fakers coordinate before sharing occurs, since the polynomial $p'(x)$ does not depend on the fakers' shares (only their identities need to be known).

E Denial-of-Service Faking

In this section, we discuss denial-of-service (DoS) faking. For denial-of-service faking, the fake-hiding guarantee does not make any sense; it is unreasonable to hope to hide the fact that someone faked their share, when the goal of faking is to cause reconstruction to detectably abort. For the same reason, extreme FRI is not applicable as well. Instead, in the context of DoS faking, the focus is on (non-extreme) fake-real indistinguishability and faker anonymity.

We can get both of these properties by generically compiling any random faking scheme. We turn random faking into a denial-of-service by restricting the set of valid messages to be a sparse subset of the distribution from which the random message is drawn (e.g. by requiring that a valid message have λ trailing zeros). We say that if a message outside that subset is reconstructed, this is equivalent to an abort. The message yielded by random faking should be

uniform in the distribution, and will almost certainly not be in the sparse subset of valid messages, giving us DoS faking.

In Section D.3, we saw an optimal random faking construction with coordination, which worked as long as the snitches were unqualified. However, when the fakers can't coordinate, Section D.1 only shows how to get the desired properties when $t = n$. However, it is possible to achieve DoS faking when $t < n$, as we elaborate below.

E.1 Uncoordinated DoS Faking

Consider the following construction for arbitrary threshold t based on Shamir-secret sharing (say, over a finite field $\mathbb{F}_q$ of integers modulo q, with prime $q > n$), that achieves fake-real indistinguishability and faker anonymity as long as the snitches are unqualified ($|\mathsf{snitches}| < t$):

$\mathsf{Share}(m)$**:** Computes $(s_1, \ldots, s_n) \leftarrow \mathsf{Shamir.Share}(m, t)$ as a shamir-sharing of the message $m \in \mathbb{F}_q$ with threshold t. Here, $\mathsf{Shamir.Share}$ involves sampling a polynomial $p(x)$ of degree $(t-1)$ such that $p(0) = m$ and setting $s_i = p(i) \mod q$. Return $\{(i, s_i)\}_{i \in [n]}$.

$\mathsf{Fake}(s_i)$**:** Return a random share $s_i' \in \mathbb{F}_q$.

$\mathsf{Rec}(\{s_i\}_{i \in \mathsf{partyset}})$**:** Computes $m' \leftarrow \mathsf{Shamir.Rec}(\{s_i\}_{i \in \mathsf{partyset}}, t)$. Here, $\mathsf{Shamir.Rec}$ involves using Lagrange interpolation to identify a $(t-1)$-degree polynomial $p''(x)$ consistent with $\{s_i\}_{i \in \mathsf{partyset}}$. Return $p''(0)$ if such a polynomial exists and $\perp$ otherwise.

This construction clearly offers correctness and privacy which follows directly from the properties of Shamir secret sharing. Next, we observe that if snitches are unqualified and faking occurs, since all the shares are randomly sampled from $\mathbb{F}_q$, it is not possible for the adversary to distinguish the fake shares from the real ones. This allows us to achieve fake-real indistinguishability and faker anonymity. We point that once the snitches *are* qualified, we lose both fake-real indistinguishability and faker anonymity, but that cannot be helped.