Electronic Colloquium on Computational Complexity, Report No. 200 (2018)

Leakage-Resilient Secret Sharing

Ashutosh Kumar [*] Raghu Meka [†] Amit Sahai [‡]

Abstract

In this work, we consider the natural goal of designing secret sharing schemes that ensure security against a powerful adaptive adversary who may learn some "leaked" information about *all* the shares. We say that a secret sharing scheme is *p-party leakage-resilient*, if the secret remains statistically hidden even after an adversary learns a bounded amount of *leakage*, where each bit of leakage can depend *jointly* on the shares of an *adaptively* chosen subset of p parties. A lot of works have focused on designing secret sharing schemes that handle *individual* and (mostly) *non-adaptive* leakage for (some) *threshold* secret sharing schemes ([DP07, DDV10, LL12, ADKO15, GK18a, BDIR18]).

- We give an unconditional compiler that transforms any standard secret sharing scheme with arbitrary access structure into a p-party leakage-resilient one for p logarithmic in the number of parties. This yields the first secret sharing schemes secure against adaptive and joint leakage for more than two parties.
- As a natural extension, we initiate the study of *leakage-resilient non-malleable secret sharing* and build such schemes for general access structures. We empower the computationally unbounded adversary to adaptively leak from the shares and then use the leakage to tamper with each of the shares arbitrarily and independently. Leveraging our p-party leakage-resilient schemes, we also construct such non-malleable secret sharing schemes: any such tampering either preserves the secret or completely 'destroys' it. This improves upon the non-malleable secret sharing scheme of Goyal and Kumar (CRYPTO 2018) where no leakage was permitted. Leakage-resilient non-malleable codes can be seen as 2-out-of-2 schemes satisfying our guarantee and have already found several applications in cryptography [LL12, ADKO15, GKP$^+$18, GK18a, CL18, OPVV18].
- Our constructions rely on a clean connection we draw to communication complexity in the well-studied number-on-forehead (NOF) model and rely on functions that have strong communication-complexity lower bounds in the NOF model (in a black-box way). We get efficient p-party leakage-resilient schemes for p upto $O(\log n)$ as our share sizes have exponential dependence on p. We observe that improving this dependence from $2^{O(p)}$ to $2^{o(p)}$ will lead to progress on longstanding open problems in complexity theory.

[*] UCLA. a@ashutoshk.com
[†] UCLA. raghum@cs.ucla.edu
[‡] UCLA. sahai@cs.ucla.edu

ISSN 1433-8092

1 Introduction

Shamir [Sha79] and Blakley [Bla79] initiated the study of secret sharing schemes by constructing threshold secret sharing schemes that allow any set of t parties, out of n parties total, to reconstruct the secret. Furthermore, crucially, the secret is hidden given less than t shares. For the sake of exposition, in this introduction we will focus only on t-out-of-n schemes, whereas our results will also apply to more general access patterns. We set up some basic notation using the definition below:

Definition 1. *Let* $\mathbf{Share} : \{0,1\}^k \to (\{0,1\}^\ell)^n$ *be any efficient randomized algorithm mapping k bit secrets into n shares, each of length ℓ bits. Let* $\mathbf{Rec} : (\{0,1\}^\ell)^n$ *be a (deterministic) algorithm that maps a collection of shares back to a possible secret. The pair* $(\mathbf{Share}, \mathbf{Rec})$ *is called a t-out-of-n secret sharing scheme (SS) mapping k bit secrets into ℓ bit shares if:*

- ***Perfect correctness** : Any t-out-of-n shares can be used to reconstruct the secret correctly. For any secret $a \in \{0,1\}^k$, for any set $T \subseteq [n]$ with $|T| \geq t$,*

$$\Pr\left[\mathbf{Rec}\big(\mathbf{Share}(a)_T\big) = a\right] = 1$$

  where the probability is over the randomness of the sharing function and $\mathbf{Share}(a)_T$ *denotes the restriction of the n shares produced by* $\mathbf{Share}(a)$ *to the ones identified by the set T.*

- ***Perfect secrecy** : Less than t shares reveal no information about the underlying secret. More formally, for any two secret $a, b \in \{0,1\}^k$, any set $U \subseteq [n]$ with $|U| < t$,* $\mathbf{Share}(a)_U$ *is identically distributed to* $\mathbf{Share}(b)_U$*.*

Secret sharing schemes, while originally envisioned with only the goal of secrecy formulated above, have been strengthened in various ways, such as by adding verifiability [RBO89], robustness [CDF$^+$08], or functionality [BGI15]. In this work, our focus is on a substantially stronger secrecy goal—*leakage-resilience*.

Leakage-resilience has a long history in cryptography. Motivated by the fascinating goal of securing circuit computation against an adversary who probes the values of internal wires of the cirucuit, Ishai, Sahai, and Wagner [ISW03] initiated the study of private circuits. Micali and Reyzin [MR04] put forward a very general model for such side-channel attacks. Subsequently, a lot of primitives in cryptography were made leakage-resilient [DP07, DP08, ADW09, ADN$^+$10, DDV10, GR15, LL12, ADKO15, GIM$^+$16, GK18a, BDIR18].

Focusing now on leakage resilience in the context of secret sharing, Dziembowski and Pietrzak [DP07] developed an *intrusion-resilient* secret sharing scheme using alternating extractors [CG88]. Davì, Dziembowski and Venturi [DDV10] constructed the first 2-out-of-2 secret sharing scheme that statistically hides the secret even after an *adaptive* adversary executes a bounded communication leakage protocol on the two shares. Liu and Lysyanskaya [LL12] and [ADKO15] constructed leakage-resilient non-malleable codes, which can be seen as 2-out-of-2 leakage-resilient schemes which also feature non-malleability against such bounded leakage-protocols. Recently, Goyal and Kumar [GK18a, GK18b] constructed 'non-malleable secret sharing schemes' (NMSS) for general access structures by first designing a 2-out-of-n secret sharing scheme that hides the secret even when a non-adaptive adversary learns some bounded amount of information independently from each of the n shares. Using tools from Fourier analysis developed for additive combinatorics, Benhamouda, Degwekar, Ishai and Rabin [BDIR18] showed that Shamir's t-out-of-n secret sharing scheme for large values of $t = n - o(\log n)$ is leakage-resilient against a non-adaptive adversary who independent leaks bounded amount of information from each share individually. (See below for further discussion of these works and formal definitions.)

Therefore the focus of recent literature on leakage-resilient secret sharing is on handling *individual* and (mostly) *non-adaptive* leakage for *threshold* secret sharing schemes. In this work, we aim to develop a more comprehensive theory of leakage resilient secret sharing. To this end, our main focus will be on handling *joint* leakage from multiple shares at once. For example, in our model, we would allow the adversary to specify an arbitrary leakage function f and obtain the output $f(share_1, share_2)$, where $share_1$ is the share given to Party 1, and $share_2$ is the share given to Party 2. Furthermore, we consider *adaptive* leakage for secret sharing schemes that support *general access structures.*

In particular, we model this by viewing leakage as (an adversary) running a communication protocol, where in each round, a group of at most p parties (out of a total of n) get together and compute a message based on all messages in the transcript so far, and the set of shares known to all the parties in the group. This process continues until a limit of at most μ bits have been communicated (or *leaked*). We call such protocols *bounded collusion protocols* (BCPs), and we will call the set of protocols obeying the restrictions above (p, n, μ)-BCP or *p-party collusion protocols* when n, μ are not too important.

The above definition is motivated by the fundamental *Number-on-Forehead* (NOF) model [CFL83, BNS92] from communication complexity (see book of [KN06]). In particular, the $p = n - 1$ case corresponds exactly to the well-studied NOF-model and $p = 1$ corresponds to the well-studied *number-in-hand* (NIH) model [PVZ12, BEO$^+$13, BO15]. Our model is also particularly well suited in the context of secret sharing: for instance, for t-out-of-n threshold secret sharing schemes, leakage-resilience is not possible if t or more parties can collude as they can just compute the secret. Thus, for the case of (t, n)-threshold schemes, resilience against $(t-1)$-party collusion protocols is the best one could hope for.

Even more generally, our work is guided by the following question: *Given a class of communication protocols $\mathcal{P}$, can we design $\mathcal{P}$-resilient secret sharing schemes in that the secret is statistically hidden from an adversary who sees the entire transcript of a protocol from $\mathcal{P}$ executed on the shares?*

The above discussion leads us to the main notion of leakage-resilience secret sharing schemes that we study (see Section 2 for a more formal definition):

Definition 2. *(**p-party leakage resilience**) Let (*$\mathbf{Share}, \mathbf{Rec}$*) be a t-out-of-n secret sharing scheme that shares k bit secrets into n shares each of bit-length ℓ bits. Let μ be any bound on allowed leakage and $1 \leq p < t$ be any collusion bound. We say that (*$\mathbf{Share}, \mathbf{Rec}$*) is* $(\mathbf{p}, \mathbf{t}, \mathbf{n})$*-**leakage-resilient secret sharing scheme** (or (p, t, n)-LRSS in short) if for any* leakage *protocol* Leak *in (p, n, μ)-BCP, and for every pair of secrets $a, b \in \{0,1\}^k$, we have* $\mathsf{Leak}(\mathbf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathbf{Share}(b))$.[1]

We remark that all known LRSSs that go beyond 2-out-of-2 sharing ([GK18a, GK18b, BDIR18]) only study resilience against non-adaptive adversaries[2]. In particular, 3-out-of-3 secret sharing schemes that are resilient against 2-party collusion protocols protocols were not known before our work. For a more detailed comparison with existing works, see the related work section below.

[1] Here $A \approx_\epsilon B$ means A, B are ϵ-close in statistical distance; see Section 2 for more details.

[2] In the language of being resilient to protocols, these works can be seen as constructing schemes that are resilient to *simultaneous one-round number-in-hand protocols.*

1.1 Leakage-Resilient Non-Malleable Secret Sharing

Goyal and Kumar [GK18a] recently introduced non-malleable secret sharing (NMSS) where they consider an adversary who can tamper with all the shares. Such an adversary can completely 'destroy' the secret by overwriting the shares. The adversary can also leave the secret intact by not tampering anything. Motivated by the fascinating line of work on non-malleable codes and extractors [DW09, DPW10, LL12, DLWZ14, ADL14, CG14, CGL16, Li17], Goyal and Kumar [GK18a] defined a secret sharing scheme to be non-malleable when the aforementioned two unavoidable actions are the only ones that a computationally unbounded adversary can perform irrespective of the way it tampers with the shares. In other words, any tampering either leaves the secret intact, or completely 'destroys' the secret leading to the reconstruction of a completely "unrelated" one. Note that non-malleable codes in two split-state model [LL12, DLWZ14, ADL14, CG14, CGL16, Li17] are a special case of 2-out-of-2 NMSSs. [GK18b] give an efficient compiler that converts any standard secret sharing scheme into one that ensures non-malleability against an adversary who tampers with each of the shares arbitrarily and independently. [GK18a] constructs t-out-of-n schemes against a stronger adversary who chooses any t shares, partitions it into two non-empty subsets and jointly tampers with each subset independently.

The motivation for non-malleability stems from the natural desire to protect cryptosystems from physical tampering. Notice, however, that a leakage attack might be easier to perform than a tampering one. This is because, unlike a tampering attack, leakage attack does not necessarily require the adversary to alter the state of a (classical) cryptosystem. Therefore, it is natural to enable the tampering adversary to also perform leakage attacks. Consequently for secret sharing schemes, we empower the adversary to adaptively and arbitrarily leak some bounded amount of information from all the shares and in addition use this leakage to arbitrarily tamper with each of the shares. Seeking inspiration from the definition of [LL12, ADKO15, GK18a], we define a secret sharing scheme to be leakage-resilient non-malleable (LR NMSS) if the secret reconstructed from these tampered shares is either the original secret or a completely "unrelated" one.

The special case of 2-out-of-2 LR NMSS has already received considerable attention in the literature under the name of *two split-state leakage-resilient non-malleable codes*. Liu and Lysyanskaya [LL12] defined and constructed such codes against computationally bounded adversaries. Aggarwal, Dziembowski, Kazana, and Obremski [ADKO15] obtained the first information-theoretic construction of 2-out-of-2 LR NMSS. [GKP$^+$18] crucially used the 2-source non-malleable extractor based codes of Li [Li17] as a 2-out-of-2 LR NMSS to design three round 'concurrent' commitment protocols that additionally feature almost-optimal communication complexity. Chattopadhyay and Li [CL18] also gave constructions of 2-out-of-2 LR NMSS (based on [CGL16]). Goyal and Kumar [GK18a] constructed an 'assymetric' 2-out-of-2 LR NMSS (based on [CGL16]) and used it to design t-out-of-n secret sharing schemes that are non-malleable even against an adversary who chooses a fixed subset of t shares, partitions into two non-empty sets of different cardinality and jointly tampers the shares in each subset independently. The resulting construction, with some work, can be generalized to obtain t-out-of-n LR NMSS when the adversary is restricted to obtain leakage from a total of at most t shares. Unfortunately, this cannot be generalized to cases where the adversary can leak individually from more than t shares. In our work, we crucially leverage our new p-party LRSS schemes for $p > 1$ to build LR NMSS for general access structures (in particular t-out-of-n secret sharing) that can handle individual tampering after adaptive individual leakage from all n shares.

1.2 Related Work

Before we describe our results and techniques in detail, let us first consider previous related work, and discuss some limitations of current techniques towards achieving the goals we seek.

Can we derive leakage-resilience from linear secret sharing schemes? Most existing secret sharing schemes are linear [Bei] (Chapter 4). Unfortunately, there are known dangers when trying to derive leakage-resilience for such schemes. For instance, Guruswami and Wootters [GW16] (see also [TYB17,GR17,MBW18]) surprisingly showed that one can learn the secret under Shamir's scheme even with one-bit leakage from each share. [BDIR18] overcome this using fields of large characterstic, and proved that Shamir's t-out-of-n scheme can ensure security against non-adaptive individual leakage when $t = n - o(\log n)$. Unfortunately, it is not clear how to extend this result beyond this parameter regime or to handle adaptive adversaries.

Can we use extractors to get 2-party leakage-resilient schemes? Most existing $(1, 2, 2)$-LRSSs are based on two source extractors [DP07, DDV10, ADKO15, GK18a]. These constructions rely on the following powerful observation: if the two shares are independent, then conditioning on the entire transcript of a bounded communication protocol preserves the conditional independence between them, and therefore independent source extractors can be invoked for proving leakage-resilience. Unfortunately this idea does not generalize to 2-party collusion protocols even for 3-out-of-3 schemes. Consider 3 bits of leakage corresponding to the 3 subsets of size 2. As we fix the three leaked bits, conditional independence between pairs is lost (unlike the 2-out-of-2 case), and we cannot rely on independent source extractors. We face further challenges when considering 3-out-of-5 schemes. Even without leakage, the five shares cannot be directly modeled as independent sources, as any 3-out-of-5 shares have to encode the same secret. Moreover, leaking even a single bit from any one of the shares may reveal some joint information about other shares, and it is not clear how to rely on extractors.

t-out-of-n schemes with leakage from $\leq t$ shares. Any t-out-of-n scheme is by definition leakage-resilient against complete leakage of any $t-1$ shares. While the t-out-of-n NMSS scheme of Goyal and Kumar [GK18a] does not consider leakage, with some work their proof can be generalized to allow leakage from at most t shares (that is the adversary gets no information about at least $n-t$ shares). Unfortunately, their proof cannot be generalized to either achieve non-adaptive 1-party leakage-resilient scheme for $n > t$ or achieve non-adaptive 2-party leakage-resilient for $t = n = 3$. We stress that in contrast, our model allows for leakage to be obtained by the adversary on *all* of the n shares, not just at most t of them.

Can we extend the LRSS of Goyal and Kumar [GK18a]? Goyal and Kumar [GK18a] constructed 2-out-of-n schemes that handles non-adaptive individual leakage. They give each pair of parties an independent 2-out-of-2 sharing of the secret, and therefore the share size has an $O(n)$ term. Generalizing this, for any constant c, one can give every set of c parties an independent c-out-of-c encoding of the secret, and obtain a c-out-of-n scheme that is resilient against non-adaptive individual leakage, with share size $O(n^{c-1})$. Unfortunately, apart from not being able to handle super-constant thresholds t, this construction relies on independent source extractors, and we cannot prove it to be resilient against even non-adaptive 2-party collusion protocols.

1.3 Our Results

Leakage-resilient secret sharing. We recall that we wish to construct secret sharing schemes that are p-party leakage-resilient. As our main result, we give a generic compiler that transforms any secret sharing scheme into a p-party leakage-resilient one. We note that prior to our work, no 3-out-of-3 schemes that is 2-party leakage-resilient was known.

Theorem 1 (Informal)**.** *For any collusion bound $p \geq 1$, any access structure $\mathcal{A}$ supported on $n \geq 1$ parties such that each authorized set has more than p parties, suppose there is a perfect (resp. statistical, computational) secret sharing scheme realizing access structure $\mathcal{A}$ that shares k bit secrets into n shares each of length ℓ bits. Then for any leakage-bound μ, any error $\epsilon > 0$, there is a perfect (resp. statistical, computational) secret sharing scheme realizing $\mathcal{A}$ that is leakage resilient against (p, n, μ)-BCP. The resulting scheme has shares secrets of k bits into n shares each of length $\ell + k(\log n)(\mu + \log(1/\epsilon)) \cdot 2^{O(p)}$.*

We remark that efficient constructions of LRSSs with asymptotically better dependence on the collusion bound p will lead to breakthroughs in communication complexity. For further details about such implications see Section 6. We mention some interesting corollaries of our main result. Using Shamir's t-out-of-n secret sharing scheme [Sha79], we get the first t-out-of-n secret sharing schemes that are p-party leakage-resilient. Note that no such schemes were known even for $p = 1$ (for unrestricted values of t and n):

Corollary 1 (Informal)**.** *For any number of parties $n \geq 2$, any collusion bound $p = O(\log n)$, any threshold $t > p$, there is an efficient [3] t-out-of-n perfect secret sharing scheme that is p-party leakage-resilient.*

Instantiating our compiler with the perfect secret sharing scheme from Karchmer and Wigderson [KW93], we get:

Corollary 2 (Informal)**.** *For any access structure that can be described by a polynomial-size monotone* span program *for which authorized sets have size greater than $p = O(\log n)$, there exists an efficient perfect secret sharing scheme that is p-party leakage-resilient.*

Using the computational scheme of Yao (mentioned in [Bei11]), we get:

Corollary 3 (Informal)**.** *If one-way functions exist, then for any access structure that is computable by monotone boolean circuits of polynomial size for which authorized sets have cardinality greater than $p = O(\log(n))$, there exists an efficient computational secret sharing scheme that realizes this access structure and is p-party leakage-resilient.*

Note that the resulting secret sharing scheme features *statistical* leakage-resilience even though the secrecy is computational to begin with. Furthermore, using the secret sharing scheme from Komargodski, Naor, Yogev [KNY14], we arrive at the following:

Corollary 4 (Informal)**.** *If one-way functions and witness-encryption for* **NP** *exist, then for every* **monotone NP** *access structure for which authorized sets have cardinality greater than $p = O(\log(n))$, there exists an efficient computational secret sharing scheme that realizes this access structure and is p-party leakage-resilient.*

[3] A leakage-resilient secret sharing scheme is efficient if the sharing and reconstruction functions run in $\mathrm{poly}\big(n, k, \mu, \log(1/\epsilon)\big)$ time where n is the number of parties, k is the length of the secret, μ is the leakage-bound and $\epsilon > 0$ is the leakage error.

Leakage-resilient non-malleable secret sharing (LR NMSS). We also define and construct LR NMSS for general access structures, significantly improving the state-of-art that only deals with the special case of 2-out-of-2 LR NMSS [LL12, ADKO15, ADL14, GKP+18, GK18a, CL18] and t-out-of-n NMSS that can handle leakage and tampering when restricted to at most t shares [GK18a].

Theorem 2 (Informal)**.** *For any access structure $\mathcal{A}$ that does not contain singletons, if there exists an efficient statistical (resp. computational) secret sharing scheme realizing access structure $\mathcal{A}$, then there exists an efficient statistical (resp. computational) secret sharing scheme realizing $\mathcal{A}$ that is statistically non-malleable against an adversary who obtains a bounded amount of information by adaptively leaking from each of the shares, and then uses this leakage to tamper each of the shares arbitrarily and independently.*

We note that even for $t = 2, n = 3$ no t-out-of-n scheme satisfying our guarantee was known before. Instantiating our compiler with Shamir's secret sharing scheme we get

Corollary 5 (Informal)**.** *For any threshold $t \geq 2$, any number of parties $n \geq t$, there is an efficient statistical t-out-of-n secret sharing scheme that is statistically non-malleable against the adversary specified in Theorem 2.*

Instantiating our compiler with various secret sharing schemes [KW93, Bei11, KNY14], we can also get further corollaries similar to Corollaries 2, 3, 4.

1.4 Overview of Constructions

1.4.1 Leakage-Resilient Secret Sharing Schemes.

Along with providing a clean way to model leakage-resilience, modeling leakage in the form of communication protocols allows us to exploit tools from communication complexity of multi-party protocols initiated by the seminal work of Chandra, Furst and Lipton [CFL83]. Indeed, the connection to NOF model allows us to leverage fundamental results of Babai, Nisan, and Szegedy [BNS92] (also Chung [Chu90], Raz [Raz00], Sherstov [She14]) on constructing explicit hard functions against NOF protocols to get the first (and simple) secret sharing schemes that are secure against adaptive, joint leakage. We next describe the main ideas by focusing on the threshold access structure.

A simple $(n-1, n, n)$-LRSS. Given inputs $x_1, \ldots, x_n \in \mathbb{F}_2^m$, let $GIP(x_1, \ldots, x_n) = \sum_{i=1}^{m} \prod_{j=1}^{n} x_{ij}$ mod 2 be the *generalized inner-product* function. Babai, Nisan, and Szegedy [BNS92] showed that the randomized communication complexity of GIP in the NOF model is $\Omega(m/4^n)$. This was further tightened by Chung [Chu90] to $\Omega(m/2^n)$. We can use their lower bound to construct a $(n-1, n, n)$-LRSS as follows.

Given secret $s \in \{0, 1\}$, sample $sh_1, \ldots, sh_{n-1}$ uniformly at random from $\mathbb{F}_2^m$ and choose sh_n to be uniformly random among all x such that $GIP(sh_1, \ldots, sh_{n-1}, x) = s$.

Analysis. It is not hard to see that any subset of $(n-1)$-shares statistically hides the underlying secret. Further, the lower bound of [BNS92, Chu90] implies that for uniformly random inputs $x_1, \ldots, x_n \in \mathbb{F}_2^m$, the output of GIP is almost unbiased even conditioned on the transcript of a NOF protocol with communication at most $cm/2^n$ for some constant $c > 0$. It is not hard to argue that this *correlation lower bound* implies that the scheme above is a $(n-1, n, n)$-LRSS when the communication is bounded by $cm/2^n$.

While the above already suffices as a building block in our subsequent constructions, we do not need to work with GIP specifically and give a similarly simple argument to build a $(n-1, n, n)$-LRSS from any n-party function with large NOF complexity in a black-box manner. The latter also has the additional advantage of getting perfect secrecy while the GIP-based construction above achieves statistical secrecy. See section 3 for details.

Building $(p, p+1, n)$-LRSSs from $(p, p+1, p+1)$-LRSSs. There are two hurdles in generalizing the above approach to construct more general (p, t, n)-LRSSs.

Owing to the existing lower bounds for communication complexity in the NOF model, the $(n-1, n, n)$-LRSSs construction above incurs a 2^n blow-up in the share-length. Indeed, as described earlier, such a blow-up is unavoidable without further breakthroughs in communication complexity. In our context, we could hope to avoid the exponential dependence on the number of parties by exploiting the fact that the collusion bound p could be small. For example, can we construct efficient $(2, 3, n)$-LRSSs?

A natural approach for this perhaps is to start with functions that are hard against p-party collusion protocols for $n \gg p$. While it does not immediately follow from stated results (to the best of our knowledge), the techniques of [BNS92] can indeed be extended to show that GIP requires $\Omega(n/2^p)$ communication to compute under p-party collusion protocols.

However, there is another additional challenge: the correlation lower bounds of [BNS92] (and for other functions in the NOF model) work when the joint distribution on the inputs is uniform (or have very specific structure such as for set disjointness (see Sherstov [She14] and references there in). On the other hand, as we want the shares to constitute a t-out-of-n scheme, we necessarily need to have large dependencies (in particular, any t shares have dependencies). As a result, it is not clear how to extend the discrepancy based correlation lower bounds in the multi-party setting to obtain lower bounds against distributions that might come out of t-out-of-n secret sharing schemes.

We will overcome these hurdles in a direct way and show how to construct $(p, p+1, n)$-LRSSs from any $(p, p+1, p+1)$-LRSSs in a black-box manner inspired by the idea of *reusing* shares as studied for example in [BBDW96, Bla99, Des98]. For the moment, let us forget about leakage-resilience and only focus on constructing $(p+1)$-out-of-n SS given a $(p+1)$-out-of-$(p+1)$ SS.

Brute-force approach. A natural idea is to consider different instantiations of the $(p+1)$-out-of-$(p+1)$ scheme for the secret, one for each possible subset of $[n]$ of size $p+1$ and give the involved parties a share from this scheme. The reconstruction property of the new scheme follows immediately from that of the original scheme and it is also not difficult to argue that the new scheme essentially inherits the secrecy properties of the $(p+1)$-out-of-$(p+1)$ scheme. However, a drawback of this approach is that the share size would incur a $O(n^p)$-factor blowup owing to creating separate instantiations for each possible subset. Given this, one could ask if we can do better than this naive approach.

Reusing shares via perfect hash families. It turns out that one can do much better than the naive approach by using a special class of hash functions. The following elegant idea is attributed to Kurosawa and Stinson in the surveys [Bla99, Des98]. A family of hash functions $H = \{h : [n] \to [p+1]\}$ is called a *perfect hash family* if for every subset $I \subseteq [n]$ of cardinality $p+1$, there is a function $h \in H$ such that I is injective on I. These families were introduced in the seminal work of Fredman, Komlós, and Szemerédi [FKS84]. Alon, Yuster and Zwick [AYZ95] and Naor, Schulman and Srinivasan [NSS95] have given almost-optimal efficient deterministic constructions of such families containing at most $2^{O(p)} \log(n)$ hash functions. Let N denote the number of hash functions in the given family H, and let $H = \{h_1, \dots, h_N\}$.

Construction of $(p, p+1, n)$*-LRSS.* To share a secret m, we first construct N independent instantiations of the $(p, p+1, p+1)$-LRSS to obtain shares $(sh_1^i, \dots, sh_{p+1}^i)$ for $i \in [N]$. For each $j \in [n]$ set the share of the j'th party to be

$$share_j = (sh^1_{h_1(j)}, sh^2_{h_2(j)}, \dots, sh^N_{h_N(j)}).$$

Note that the share length of the new scheme is a factor of $N = 2^{O(p)} \log n$ more than that of the underlying scheme. While this factor blow-up may not be the best possible in such a black-box construction, as our underlying $(p, p+1, p+1)$-LRSS also has shares of length $2^{O(p)}$, the additional factor is not too important for us.

Reconstruction: We claim that any subset of $p+1$ parties can reconstruct the secret under the above scheme. Let $J \subseteq [n]$ with $|J| = p+1$. Note that as H is a perfect hash family, there must exist a $i \in [N]$ such that h_i is injective on J so that $h_i(J) = [p+1]$. The parties in J can reconstruct the secret as follows: Find $i \in [N]$ such that $h_i(J) = [p+1]$; Apply the reconstruction procedure of the underlying $(p, p+1, p+1)$-LRSS on the shares $(sh_1^i, sh_2^i, \dots, sh_{p+1}^i)$ which they have access to because $h_i(J) = [p+1]$.

Leakage-resilience: Just as before, it is easy to show secrecy of the new scheme. However, additional care is required to argue leakage-resilience against colluding protocols. In particular, it may be possible, that the additional information given to each party (via multiple encodings of the same secret) somehow helps an adversary to design "better" leakage-protocols. We prove that even the composed scheme has p-party leakage resilience by using a hybrid arguement, where we use any leakage protocol on the constructed $(p, p+1, n)$-scheme to give a leakage protocol on one of the instantiations of the underlying $(p, p+1, p+1)$-LRSS.

Building (p, t, n)-LRSSs from $(p, p+1, n)$-LRSSs. Now we construct (p, t, n)-LRSSs for arbitrary $t > p$. Given the $(p, p+1, n)$-LRSS construction from above, the sharing function of the final scheme is quite simple. Given a secret m, share using a 2-out-of-2 scheme to obtain $l, r \leftarrow$ 2-out-of-2-Share(m). Share l using any standard t-out-of-n scheme and r using our $(p, p+1, n)$-LRSS to get $l_1, \dots, l_n$ and $r_1, \dots, r_n$ respectively. Final shares have the form $share_i \leftarrow l_i, r_i$ for each $i \in [n]$. The reconstruction procedure is straightforward given the sharing function.

Any $t-1$ shares will perfectly hide the secret because even though $t-1$ shares may reveal r, l will be hidden by the perfect secrecy of the t-out-of-n scheme. Moreover, leakage-resilience follows from the intuition that even though the leaking adversary may learn l, r will be hidden by the leakage-resilience of $(p, p+1, n)$-LRSS.

Handling general access structures. While we focused on the case of threshold access structure in the above discussion, the arguments in fact extend relatively straightforwardly to give p-party leakage-resilience for general access structures as long as every authorized set has size more than p. Note that the latter is a necessary condition for p-party leakage-resilience.

Sharing multiple bits. The techniques also extend to allow to share multiple bits at the same time and we defer the details to the actual constructions.

1.4.2 Leakage-Resilient Non-Malleable Secret Sharing Schemes.

Obtaining LR NMSS for general access structures as in Theorem 2 turns out to be considerably more challenging and is significantly more technical. Existing works on 2-out-of-2 NMSS have required various sophisticated techniques: [ADL14] required additive combinatorics based analysis to show some 'non-malleability' properties of the inner-product function [ADL14], [CGL16] required new tools such as flip-flop alternating extractor, and [Li17] made use of a correlation breaker with advice generator [CL16]. To ensure leakage-resilience in non-malleable schemes one needs to prove that even with leakage, the adversary cannot violate non-malleability. Consequently, the proof of leakage-resilience in existing constructions [ADKO15, GKP$^+$18, GK18a, CL18] is very closely tied to the corresponding proof of non-malleability.

Our starting point is the compiler of the Goyal and Kumar [GK18b] that converts any secret sharing scheme into one that ensures non-malleability against an adversary who independently tampers with all the shares (but does *not allow* leakage). To convey our most important ideas, let us first recall the t-out-of-n construction of [GK18a] for $t \geq 3$. To share a secret m, let $\ell, r \leftarrow$ 2-out-of-2-NMSS(m), $(\ell_1, \dots, \ell_n) \leftarrow$ t-out-of-n-ShamirShare(ℓ) and $(r_1, \dots, r_n) \leftarrow$ 2-out-of-n-LRShare(r). Let $share_i \leftarrow \ell_i, r_i$ for each $i \in [n]$.

While this scheme is non-malleable against individual tampering, it does not satisfy our stronger notion as the leakage may reveal some (or all) bits of l (say by using leakage-functions of [GW16]), which can then be used to tamper with the shares of r and consequently the tampering of r will no longer be independent of l (as needed to use the guarantees of the 2-out-of-2-NMSS). One natural idea to fix this would be to rely on leakage-resilience of existing 2-out-of-2 LR NMSS (or leakage-resilient non-malleable codes [ADKO15, GKP$^+$18, GK18a, CL18]). Unfortunately, this approach does not allow us to handle leakage protocols that touch more than t shares. In our case, we have to deal with leakage from all the n shares, and therefore need to sample all the n shares independently using l and r in our security reduction to the 2-out-of-2 NMSS. Existing reductions of Goyal and Kumar did not have to sample more than t shares as their reconstruction functions were carefully designed to only use the first t shares (given any number of shares as input). We highlight our main ideas to fix this. Even though we are constructing LR NMSS that are secure against individual leakage and tampering, our constructions will actually rely on our LRSSs that are resilient against bounded collusion protocols.

Use our LRSS schemes Instead of relying on leakage-resilience of the 2-out-of-2 LR NMSS, we derive leakage-resilience from our LRSSs to share ℓ. We also use our new adaptive 2-out-of-n LRSS to share r. Our hope would be to rely on the leakage-resilience of these schemes to generate n 'fake' shares encoding arbitrary l and r in our reduction, to obtain a 'fake' leakage-transcript by executing the leakage-protocol on n 'fake' shares, and upon availability of real values of l and r, somehow adjust these 'fake' shares and use the 'fake' leakage-transcript to give explicit functions that independently tamper with l and r. One may be tempted to think that this will allow us to rely on the non-malleability of 2-out-of-2 NMSS. However, there is a subtle but fundamental issue, *the leakage transcript may be independent of each of l and r, but may have some information about the secret m encoded by l and r*, and the consequent tampering of shares may depend on the secret (trivially violating non-malleability). To see this, observe that even when the leakage-transcript has full information about m, leakage-resilience of neither the scheme sharing l nor the scheme sharing r is violated since the secret m is independent of each of l and r (by the secrecy property of 2-out-of-2 NMSS).

Using joint leakage and adaptivity. At a very high level we overcome this using our secret sharing schemes that are secure against adaptive, joint leakage. In particular, we strengthen our t-out-of-n scheme to be secure against joint-leakage. While we continue to rely on the idea of [GK18a] of treating the tampered shares of l as leakage from shares of r, in this work, we also consider leakage in the other direction. In particular, we rely on the adaptive joint-leakage from two shares of l to compute tampered r in our security reduction to 2-out-of-2 NMSS.

Separately build LR NMSS for authorized pairs. Similar to [GK18b], we also execute two schemes in 'parallel': one catering to authorized subsets of size at least three and other designed for authorized pairs. Using 2-out-of-2 LR NMSS of [ADKO15], we construct LR NMSS for authorized pairs. Unfortunately, we face new difficulties while composing these two schemes in 'parallel'. [GK18b] avoided dependencies between the two schemes by ensuring that every minimal authorized set of one scheme did not have any authorized set of another scheme. In our case, adaptive leakage may correlate the shares of all authorized sets.

Use 'leakage-leveraging'. We fix many issues related to composition with our idea of *leakage-leveraging* (terminology inspired from the widely used *complexity-leveraging* - see [KS17] and references therein). Specifically we think of leakage transcript from the leakage-protocol as leakage from shares of l. Next, we think of the leakage transcript and tampered shares of l as *adaptive* leakage from shares of r. Note that we crucially use the adaptivity supported by our leakage-resilient schemes since even the independently tampered shares may depend on the leakage-transcript which may have information about all the shares. We apply this idea one more time and think of the leakage transcript and tampered shares of both l and r as *adaptive* leakage from shares of the LR NMSS designed for authorized pairs. At a very high level this enforces one direction of independence necessary for proving non-malleability. We remark that this idea may have other applications in cryptography to enforce some form of 'synchronicity'.

1.5 Open Problems

Along with natural open questions about further strengthening our results for secret sharing schemes, our work suggests several natural and independently interesting questions in communication complexity and pseudorandomness. We describe a few of them next.

Lower bounds for bounded-collusion protocols. Proving lower bounds for explicit functions in NOF-model when the number of parties is $\omega(\log m)$ where m is the input length is a fundamental challenge in communication complexity. Allowing a large number of parties n compared to the collusion bound p, could make the task of designing explicit hard functions easier and we can ask if we can design explicit functions that are harder for p-party collusion protocols with a large number of parties compared to the lower bounds we currently know against NOF protocols.

Extractors for cylinder intersections. Trying to use extant techniques of deriving leakage-resilience from extractors [DDV10, GR15, ADKO15, GK18a] to handle BCPs raises the following question that seems interesting on its own. We start by describing a natural weakening of independent sources that we call *cylinder-intersection* sources taking inspiration from the communication complexity literature [BNS92, KN06].

Definition 3. ***(Cylinder intersection sources and Extractors)*** *Let $X_1, \ldots, X_n$ be n independent sources with min-entropy k supported on $\{0,1\}^m$. Let π be a (possibly randomized) (p,n,μ)-BCP. Let $\pi(X_1, \ldots, X_n)$ denote the transcript of the communication. We define a (m,k,p,n,μ)-*cylinder-intersection source *to be the conditional distribution of $X_1, \ldots, X_n$ obtained after fixing a* typical *transcript $\pi(X_1, \ldots, X_n)$.*

Call a deterministic function Ext : $(\{0,1\}^m)^n \to \{0,1\}$ *an extractor*[4] *for (m,k,p,n,μ)-cylinder intersections as above with error ϵ if*

$$(\mathrm{Ext}(X_1, \ldots, X_n), \pi(X_1, \ldots, X_n)) \approx_\epsilon (U_1, \pi(X_1, \ldots, X_n)).$$

Note that $p = 1$ corresponds to independent source extractors as conditioning on the transcript of a 1-party collusion protocol preserves independence (while losing some min-entropy). We also remark that such extractors will trivially imply lower bounds against p-party collusion protocols. Indeed, the results of [BNS92] do imply explicit $(m,k,p,p+1,\mu)$-extractors as above for $k \geq (1-c_p)m$ and $\mu = c'_p m$ for $c'_p \ll c_p = \Omega(1/2^p)$. In particular, for $p = 2$, they imply extractors for cylinder-intersection sources when min-entropy $k \geq cm$ for a fixed constant $c > 0$. Given the rich body of work on independent source extractors it is natural to ask if one could get extractors for min-entropy $k = \delta m$ for small constants δ when the collusion bound p is say even 2.[5]

Handle joint-leakage for non-malleable secret sharing schemes. Handling joint-leakage in NMSS schemes appears to be quite challenging. Concretely, can we construct a 3-out-of-3 SS that is non-malleable against an adversary who performs *joint-leakage* from each of the three subsets of size two, and uses this leakage to tamper with each share arbitrarily and independently? To understand the challenge, observe that joint leakage leads to loss of independence among all the shares, and therefore the subsequent 'independent' tampering is not independent in reality. Independence appears to be far more crucial for deriving non-malleability.

2 Definitions

We use capital letters to denote distributions and their support, and corresponding small letters to denote a sample from the distribution. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For any set $B \subseteq [n]$, let $\otimes_{i \in B} S_i$ denote the Cartesian product $S_{i_1} \times S_{i_2} \times \ldots \times S_{i_{|B|}}$, where $i_1, i_2 \ldots i_{|B|}$ are ordered elements of B, such that $i_j < i_{j+1}$.

Definition 4. ***(Statistical distance)*** *Let $\mathbf{D_1}$ and $\mathbf{D_2}$ be two distributions on a set S. The statistical distance between $\mathbf{D_1}$ and $\mathbf{D_2}$ is defined to be :*

$$|\mathbf{D_1} - \mathbf{D_2}| = \max_{T \subseteq S} |\mathbf{D_1}(T) - \mathbf{D_2}(T)| = \frac{1}{2} \sum_{s \in S} |Pr_{X \sim \mathbf{D_1}}[X = s] - Pr_{X \sim \mathbf{D_2}}[X = s]|$$

We say $\mathbf{D_1}$ is ϵ-close to $\mathbf{D_2}$ if $|\mathbf{D_1} - \mathbf{D_2}| \leq \epsilon$. Sometimes we represent the same using $\mathbf{D_1} \approx_\epsilon \mathbf{D_2}$.

[4] Strictly speaking, what we are defining is a *strong extractor* under standard terminology.

[5] Indeed, constructing independent-source extractors was easier when the number of sources n is large and this could be the case here too.

2.1 Secret Sharing Schemes

The following definition is inspired from the survey [Bei11].

Definition 5. *(**Access structures and sharing function**) A collection $\mathcal{A}$ is called monotone if $B \in \mathcal{A}$ and $B \subseteq C$, then $C \in \mathcal{A}$. Let $[n] = \{1, 2, \ldots, n\}$ be a set of identities of n parties. An **access structure** is a monotone collection $\mathcal{A} \subseteq 2^{\{1,\ldots,n\}}$ of non-empty subsets of $[n]$. Sets in $\mathcal{A}$ are called **authorized**, and sets not in $\mathcal{A}$ are called **unauthorized**.*

*Let $\mathcal{M}$ be the domain of secrets. A **sharing function** $\mathbf{Share}$ is a randomized mapping from $\mathcal{M}$ to $S_1 \times \ldots \times S_n$, where S_i is called the domain of shares of party with identity j. A dealer distributes a secret $m \in \mathcal{M}$ by computing the vector $\mathbf{Share}(m) = (s_1, \ldots, s_n)$, and privately communicating each share s_j to the party j. For a set $S \subseteq \{p_1, \ldots, p_n\}$, we denote $\mathbf{Share}(m)_S$ to be a restriction of $\mathbf{Share}(m)$ to its S entries.*

Definition 6. *(**Secret sharing scheme** [Bei11]). Let $\mathcal{M}$ be a finite set of secrets, where $|\mathcal{M}| \geq 2$. A sharing function $\mathbf{Share}$ with domain of secrets $\mathcal{M}$ is a (n, ϵ)-**Secret Sharing Scheme** realizing an access structure $\mathcal{A}$ if the following two properties hold :*

1. ***Correctness**. The secret can be reconstructed by any authorized set of parties. That is, for any set $T \in \mathcal{A}$, where $T = \{i_1, \ldots, i_{|T|}\}$, there exists a deterministic reconstruction function $\mathbf{Rec} : \otimes_{i \in T} S_i \rightarrow \mathcal{M}$ such that for every $m \in \mathcal{M}$,*

$$Pr[\mathbf{Rec}(\mathbf{Share}(m)_T) = m] = 1$$

   (over the randomness of the Sharing function)

2. ***Statistical privacy**. Collusion of unauthorized parties should reveal "almost" no information about the underlying secret. More formally, for any unauthorized set $T \notin \mathcal{A}$, and for every pair of secrets $a, b \in \mathcal{M}$, the following holds :*

$$\mathbf{Share}(a)_T \approx_\epsilon \mathbf{Share}(b)_T$$

   *The special case of $\epsilon = 0$, is known as **perfect privacy**. If the two distributions are computationally indistinguishable to any polynomial time adversary, we call it **computational indistinguishability**.*

2.2 Threshold Access Structure $\mathcal{A}_n^t$

Perhaps the most well-studied secret sharing scheme is the threshold secret sharing scheme or t-out-of-n secret sharing which was originally studied by Shamir and Blakley. The threshold access structure can be formally represented as $\mathcal{A}_n^t = \{B \subseteq [n] : |B| \geq t\}$. We use the notation of (t, n, ϵ)-secret sharing scheme for denoting (n, ϵ)-secret sharing scheme realizing access structure $\mathcal{A}_n^t$.

2.3 Leakage-Resilient Secret Sharing Schemes

Goyal and Kumar [GK18b] defined 2-out-of-n leakage-resilient secret sharing schemes for non-adaptive adversaries. We introduce a substantial generalization that not only encompasses general access structures, but more importantly also empowers the adversary to be adaptive. As described in the introduction, we will do so by modeling *leakage* as an adversary running a communication protocol among the n parties and trying to guess the secret based on the transcript.

Definition 7. *(**Leakage-resilient secret sharing schemes**) Let $\mathcal{M}$ be any message space and $\mathcal{A}$ be any access structure on n parties. Let $\mathcal{L}$ be a family of (possibly randomized) multi-party protocols that output some transcript. We say that a secret sharing scheme ($\mathbf{Share}, \mathbf{Rec}$) realizing access structure $\mathcal{A}$ is **ϵ-leakage-resilient** w.r.t. $\mathcal{L}$ if for every leakage-protocol $\mathsf{Leak} \in \mathcal{L}$, and for every pair of secrets $a, b \in \mathcal{M}$, the following holds :*

$$\mathsf{Leak}(\mathbf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathbf{Share}(b)).$$

That is, the distribution of the transcript of the protocol Leak when input is $\mathbf{Share}(a)$ is statistically close to the distribution of the transcript of the protocol when input is $\mathbf{Share}(b)$.

2.4 Bounded Collusion Protocols $(p, n, \mu) - BCP$

Let n denote the total number of parties and $p \leq n$. Let us call p as *collusion bound*, since it indicates an upper bound on the number of parties who can collude in any round. Let μ denote *leakage bound*, as it indicates an upper bound on the total number of bits of leakage across all rounds. At a very high level, the leakage family $(p, n, \mu) - BCP$ contains all possible multi-round leakage-protocols among n parties such that the total leakage is at most μ bits and the leakage in each round arbitrarily depends on the shares of at most p parties (along with all the leakages obtained in the preceding rounds). We formally model this in the following way :

- Let $share_1, \ldots, share_n$ be the n shares corresponding to n parties. We use τ to denote the transcript of the leakage-protocol. At the beginning of the leakage-protocol τ is empty. The transcript τ is appended with the leakage, at the end of each round of the leakage-protocol. At the end, τ can be at most μ bits long.
- In each round, the Next function is used to determine which parties will collude to jointly leak information about their shares. Formally, Next function takes the current transcript τ as input, and outputs a subset $S \subset [n]$ of cardinality at most p and a description of an arbitrary leakage function $\mathbf{f}$ that takes $\otimes_{i \in S} share_i$ as input. At the end of each round, the leaked information is appended to the current transcript.
$$\tau \leftarrow \tau \circ \mathbf{f}(\otimes_{i \in S} share_i)$$
- The previous step is repeated until the Next function outputs $\perp$. Output final transcript τ as leakage.

In our constructions, p-party leakage resilient schemes for threshold schemes play an important role and we state their definition next for clarity.

Definition 8 (**(p, t, n)-LRSS**)**.** *A (t, n, ϵ)-secret sharing scheme is a (p, t, n, μ, ϵ)-leakage resilient secret sharing scheme if the scheme is ϵ-leakage-resilient against $(p, n, \mu) - BCP$. When the paramters ϵ, μ will be clear from context, we use (p, t, n)-LRSS to refer to such schemes.*

Borrowing terminology from communication complexity literature, the special case in which each party individually leaks some information ($p = 1$) will be called number-in-hand (NIH) leakage. Similarly, for n-out-of-n secret sharing schemes in which leakage in each round depends on at most $n - 1$ parties will be called number-on-forehead (NOF) leakage.

3 LRSS for Number-on-Forehead (NOF) Leakage

Our first building block will be an efficient LRSS that is resilient against NOF leakage, i.e., the construction of an efficient $(n-1, n, n)$-LRSS. Our results here will rely on classical results from communication complexity that prove lower bound for the amount of communication required to compute a function in the number-on-forehead (NOF) model of Chandra, Furst, Lipton [CFL83]. While the above is a little repetitive, we include the usual definition of NOF communication for clarity (see [KN06] for more details and references).

Definition 9. *(**NOF communication complexity**) Suppose there are n parties, and an element of $\mathcal{D}$ is written on the forehead of each party. Each party can see the number on the forehead of all other parties, and has no idea of the number written on its own forehead. Suppose these parties wish to compute any arbitrary n party predicate (boolean-valued function) $\mathbf{f} : \mathcal{D}^n \to \{0,1\}$. They are allowed to communicate among themselves using a black-board. At the beginning, the black-board is empty, and each party is only allowed to append information to it (no erasing). Their goal is to compute $\mathbf{f}$ while minimizing the number of bits that needs to be written on the black-board. The **NOF communication complexity** refers to the minimum number of bits of communication required to gain ϵ advantage in computing f using any such protocol. More formally,*

$$\mathbf{CC}_{\mathbf{n}}^{\mathit{NOF}}(f) = \min_{\Pi} \max_{x \in \mathcal{D}^n} |\Pi(x)|$$

where Π ranges over all protocols of the above form satisfying

$$\Pi(\mathbf{f}^{-1}(0)) \not\approx_{\epsilon} \Pi(\mathbf{f}^{-1}(1))$$

where $|\Pi(x)|$ denotes the number of bits of communication required by protocol Π on input x.

The main result of this section is a simple construction to build $(n-1, n, n)$-LRSS starting from any function that has high NOF communication complexity.

Lemma 1. *For any $n \geq 1$, any leakage bound $\mu \geq 0$, any $\epsilon > 0$, if there is an efficient n party function $\mathbf{f} : (\{0,1\}^b)^n \to \{0,1\}$ with $\mathbf{CC}_{\mathbf{n}}^{\mathit{NOF}}(f) \geq \mu$, then there is an efficient $(n, n, 0)$-secret sharing scheme that is ϵ-leakage-resilient w.r.t. $(n-1, n, \mu)-BCP$. The resulting scheme, $(\mathbf{Share}_{\mathbf{n}}^{\mathbf{n}}, \mathbf{Rec}_{\mathbf{n}}^{\mathbf{n}})$, shares single bit secrets into n shares, each of bit-length $1 + b$.*

Combining the above result with known lower bounds on the number-on-forehead complexity of functions such as those in [BNS92] gives us the following:

Corollary 6. *For any $n \geq 1$ and any leakage bound $\mu \geq 0$ and $\epsilon > 0$, there exists an efficient $(n, n, 0)$-secret sharing scheme that is ϵ-leakage resilient against $(n-1, n, \mu) - BCP$ where the scheme shares single bit secrets into n shares with each of length $1 + O(2^n(\mu + \log(1/\epsilon)))$.*

Proof. [BNS92] showed that the generalized-inner-product function $GIP : (\{0,1\}^b)^n \to \{0,1\}$ defiend as $GIP(x_1, \ldots, x_n) = \oplus_{i=1}^{b} \prod_{j=1}^{n} x_{ij}$ satisfies $\mathbf{CC}_{\mathbf{n}}^{\mathsf{NOF}}(GIP) \geq cb/2^n$ for $\epsilon \geq c\exp(-b/2^n)$ for a universal constant $c > 0$. The corollary follows from using this lower bound in the above lemma. □

Note that the share length in the above construction is exponential in the number of parties. However, as we observe in Section 6, the construction above is somewhat *tight* in the sense that designing schemes with better share-length for NOF leakage as above would lead to breakthroughs in communication complexity.

The construction above relies on additive secret sharing schemes that we describe next.

3.1 XOR based Additive Secret Sharing

We recall the n-out-of-n additive secret sharing based on $\oplus$ (XOR) operation. For any $a \geq 1$, let the secrets be a bits long.

- (**Sharing function $\mathbf{XORShare_n}$**) : Let $\mathbf{XORShare_n} : \{0,1\}^a \rightarrow \otimes_{i \in [n]} \{0,1\}^a$ be a randomized sharing function. On input a secret $s \in \{0,1\}^a$, uniformly sample the first $n-1$ shares, namely $s_1, \ldots, s_{n-1}$, such that each $s_i \in \{0,1\}^a$. Compute the last share using the secret s and the sampled shares as

$$s_n \leftarrow s \oplus s_1 \oplus \ldots \oplus s_{n-1}$$

  Output $s_1, \ldots, s_n$ as the n shares.

- (**Reconstruction function $\mathbf{XORRec_n}$**) : Let $\mathbf{XORRec_n} : \otimes_{i \in [n]} \{0,1\}^a \rightarrow \{0,1\}^a$ be a deterministic function for reconstruction. On input n shares, namely $s_1, \ldots, s_n$, compute $s \leftarrow s_1 \oplus \ldots \oplus s_n$ and output the result s.

Lemma 2. *([KGH83]) For secret space of $a \geq 1$ bits, $(\mathbf{XORShare_n}, \mathbf{XORRec_n})$ (described above) is an $(n, n, 0)$-secret sharing scheme.*

Additionally this scheme has a useful property that given the secret and all but one shares, the leftover share can be efficiently computed. Formally,

Lemma 3. *Let $(\mathbf{XORShare_2}, \mathbf{XORRec_2})$ be an $(2, 2, 0)$-secret sharing scheme for single bit secrets. For any $m, sh_1, sh_2 \in \{0,1\}$, if $m \leftarrow \mathbf{XORRec_2}(sh_1, sh_2)$, then $sh_1 \leftarrow \mathbf{XORRec_2}(m, sh_2)$.*

3.2 $(n-1, n, n)$-LRSS

We are now in a position to give our first construction.

Proof of Lemma 1. Let $(\mathbf{XORShare_n}, \mathbf{XORRec_n})$ be the $(n, n, 0)$ additive secret sharing scheme for single bit secrets (as in Lemma 2). Similarly, let $(\mathbf{XORShare_2}, \mathbf{XORRec_2})$ be the $(2, 2, 0)$ additive secret sharing scheme for single bit secrets. The leakage-resilient scheme is defined as :

1. (**Sharing function $\mathbf{Share_n^n}$**):
   On input a secret m, for each $i \in [n]$, uniformly and independently sample $r_i \in \{0,1\}^b$. Execute function $\mathbf{f}$ on $r_1, \ldots, r_n$ to compute the bit $r \leftarrow \mathbf{f}(r_1, \ldots, r_n)$. Compute $s \leftarrow \mathbf{XORRec_2}(m, r)$. Secret share s using $\mathbf{XORShare_n}$ to obtain $s_1, \ldots, s_n \leftarrow \mathbf{XORShare_n}(s)$. For each $i \in [n]$, let $share_i \leftarrow (r_i, s_i)$.

2. (**Reconstruction function $\mathbf{Rec_n^n}$**) :
   On input n shares, namely $share_1, \ldots, share_n$, for each $i \in [n]$, parse $share_i$ as (r_i, s_i). Compute $\mathbf{f}$ on $r_1, \ldots, r_n$ to obtain the bit $r \leftarrow \mathbf{f}(r_1, \ldots, r_n)$. Apply the reconstruction procedure $\mathbf{XORRec_n}$ on $s_1, \ldots, s_n$ to obtain $s \leftarrow \mathbf{XORRec_n}(s_1, \ldots, s_n)$. Compute $m \leftarrow \mathbf{XORRec_2}(r, s)$. Output m.

Correctness and efficiency : Follows from the efficiency of $\mathbf{f}$. Notice that we only make black-box use of $\mathbf{f}$ and do not need to invert $\mathbf{f}$. Correctness follows from the fact that if $s \leftarrow \mathbf{XORRec_2}(m, s)$ then $m \leftarrow \mathbf{XORRec_2}(r, s)$ (by lemma 3).

Perfect secrecy : Follows from combining the facts that $r_1, \ldots, r_n$ is chosen uniformly and any $n-1$ shares of the XOR based n-out-of-n scheme are uniformly random.

Statistical leakage-resilience : Suppose the adversary specifies a leakage-protocol $\mathsf{Leak} \in (n-1, n, \mu) - BCP$ that violates the leakage-resilience of our scheme using at most μ bits of leakage. We use such an adversary to give a NOF protocol computing $\mathbf{f}$ with communication cost at most μ.

- **Initial setup** : Randomly fix $s \leftarrow \{0,1\}^a$. Compute $s_1, \ldots, s_n \leftarrow \mathbf{XORShare_n}(s)$ and fix $s_1, \ldots, s_n$.
- **Protocol** : For each $i \in [n]$, party i holds $r_i \in \{0,1\}^b$ as input. We use the Next function specified by the adversary for the secret sharing scheme, and the values of s_i fixed above to give a communication protocol for $\mathbf{f}$.

  1. Initialize an empty black-board (transcript) τ.
  2. Run the Next function with τ as input to obtain a subset $S \subset [n]$ and a leakage function $\mathbf{g}$ that takes $\otimes_{i \in S} share_i$ as input. In our communication protocol, corresponding to S, we fix a party, say $j \in [n]$, who can see the forehead of all the parties in S. Party j, uses the fixed value of s_i to create $share_i \leftarrow r_i, s_i$ for each $i \in S$, computes and writes $\mathbf{g}(\otimes_{i \in S} share_i)$ on the black-board.
  $$\tau \leftarrow \tau \circ \mathbf{g}(\otimes_{i \in S} share_i)$$
  3. Repeat the above step until $\mathsf{Next}(\tau)$ outputs $\perp$.

Observe that if the adversary of the leakage-resilient secret sharing scheme achieves some advantage in distinguishing shares of 0 and 1, then the communication protocol created in the above reduction achieves the same advantage in computing the value of $\mathbf{f}$. Also observe that the number of bits of leakage is equal to the communication required by the protocol given in the reduction. This completes the proof, as the communication complexity of $\mathbf{f}$ is at least μ bits. □

4 $(p, p+1, n)$-LRSS

In the previous section we saw how to construct secret sharing schemes that are resilient against NOF leakage. Here we handle more general threshold access structures and build $(p, p+1, n)$-LRSS. In doing so, we will also improve the share length significantly by removing the exponential dependence on number of parties but instead only have such a dependence on the collusion bound. As remarked in the introduction, efficient schemes like this were not known even for the case of $p = 1$.

The construction will use a $(p, p+1, p+1)$-LRSS in a black-box manner leading to the following:

Lemma 4. *For $1 \leq p < n$, suppose we have the following primitive: For any leakage bound μ, any error bound $\epsilon > 0$, an efficient $(p+1, p+1, 0)$-secret sharing scheme $(\mathbf{LRShare_{p+1}^{p+1}}, \mathbf{LRRec_{p+1}^{p+1}})$ that is ϵ-leakage-resilient w.r.t. $(p, p+1, \mu) - BCP$ and shares secrets of length a into $p+1$ shares, each of bit-length b.*

Then, there is an efficient $(p+1, n, 0)$-secret sharing scheme that is $2^{O(p)}\epsilon$-leakage-resilient against $(p, n, \mu) - BCP$. The resulting scheme, $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$, shares secrets of length a into n shares each of length $b \cdot 2^{O(p)}$.

By combining the above with the construction from Corollary 6 immediately gives the following:

Corollary 7. *For $1 \leq p < n$ and any leakage bound μ, error ϵ, there exists an efficient $(p+1, n, 0)$-secret sharing scheme that is ϵ-leakage-resilient against $(p, n, \mu) - BCP$. The resulting scheme, $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$, shares secrets of a bits into n shares each of length $a(\mu + \log(1/\epsilon))2^{O(p)}$.*

4.1 Proof of Lemma 4

We wish to construct $(p, p+1, n)$-LRSS from $(p, p+1, p+1)$-LRSS. As described in the introduction, we will do so by exploiting the idea of reusing shares via perfect hash families (cf. [Bla99, Des98]).

Definition 10. *[**Perfect hash families** [FKS84]] A family consisting of d functions of the form $\{f : [n] \to [p]\}$ is called a (p, n)-perfect hash function family of size d, if for all subsets $T \subseteq [n]$ of cardinality $p+1$, there exists a function f in the family such that f is injective on T.*

Such a family of functions is called efficient, if we can generate d efficient functions for this hash family, namely $(f_1, \ldots, f_d) \leftarrow \mathbf{PHF}(p, n)$, in time $\text{poly}(n, d)$.

Lemma 5. *For any collusion bound $p \geq 1$, any number of parties $n > p$, any message size $a > 0$, suppose we have the following primitives :*

1. *for any leakage bound μ, any error bound $\epsilon > 0$, an efficient $(p+1, p+1, 0)$-secret sharing scheme $(\mathbf{LRShare_{p+1}^{p+1}}, \mathbf{LRRec_{p+1}^{p+1}})$ that is ϵ-leakage-resilient w.r.t. $(p, p+1, \mu) - BCP$ and shares a secret of bit-length a into $p+1$ shares, each of bit-length c.*

2. *an efficient $(p+1, n)$-perfect hash family $\mathbf{PHF}$ of size d.*

Then there is an efficient $(p+1, n, 0)$-secret sharing scheme that is $d\epsilon$-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. The resulting scheme, $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$, shares secrets of length a into n shares, each of length cd.

Proof. Generate the d hash functions of the perfect hash family. Let $(f_1, \ldots, f_d \leftarrow \mathbf{PHF}(p+1, n)$. We use these functions in our construction of $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$ given below :

- **(Sharing function $\mathbf{LRShare_n^{p+1}}$).**
  On input a secret m, for each $j \in [d]$, share m using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m_1^j, \ldots, m_{p+1}^j \leftarrow \mathbf{LRShare_{p+1}^{p+1}}(m)$. Using functions from the above perfect hash family, for each $i \in [n]$, construct $share_i$ as $\left(m_{f_1(i)}^1, \ldots, m_{f_d(i)}^d\right)$.

- **Reconstruction function ($\mathbf{LRRec_n^{p+1}}$).**
  On input a set of shares corresponding to an authorized set T of cardinality $p+1$, for each $i \in T$, parse $share_i$ as $\big(m^1_{f_1(i)}, \ldots, m^d_{f_d(i)}\big)$. Find $j \in [d]$ such that f_j is injective on T. Use the reconstruction procedure of underlying leakage resilient scheme to compute $m \leftarrow \mathbf{LRRec^{p+1}_{p+1}}(m^j_1, \ldots, m^j_{p+1})$. Output m.

Perfect correctness: For any authorized set $T \subseteq [n]$ of $p+1$ parties, by the properties of the perfect hash family, there will be a function f_j in the family ($j \in [d]$), such that f_j is injective on T (see definitions 10). Therefore, all the $p+1$ shares of j^{th} encoding of m will be available, and correctness follows from the correctness of the underlying (p+1)-out-of-(p+1) scheme.

Perfect secrecy and efficiency : By construction, less than $p+1$ shares of our (p+1)-out-of-n scheme can only have less than $p+1$ shares of each of the d underlying (p+1)-out-of-(p+1) scheme. Efficiency follows from the efficiency of the perfect hash family and the underlying leakage-resilient scheme.

Statistical leakage-resilience: The adversary specifies a $\mathsf{Next} \in (p, n, \mu) - BCP$ that allows it to distinguish in between shares of m_1 and m_2 under the (p+1)-out-of-n scheme. We use such an adversary to construct $\mathsf{Next}_1 \in (p, p+1, \mu) - BCP$ that violates the leakage-resilience of the underlying (p+1)-out-of-(p+1) scheme.

- **Initial setup** : Randomly fix $j \in [d]$. For each $i \in \{1, \ldots, j-1\}$, share m_1 using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_{p+1} \leftarrow \mathbf{LRShare^{p+1}_{p+1}}(m_1)$. For each $i \in \{j+1, \ldots, d\}$, share m_2 using the sharing procedure of the underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_{p+1} \leftarrow \mathbf{LRShare^{p+1}_{p+1}}(m_2)$. Fix all these sampled shares.
- **Reduction** Next_1 : Using the adversarily specified Next and above fixings we give the description of Next_1.

  On input a transcript τ, execute the Next function with τ as input to obtain a subset $S \subset [p]$ and a leakage function $\mathbf{g}$ that takes $\otimes_{i \in S} share_i$ as input. If the output of Next is $\perp$, then output $\perp$. Otherwise, we construct leakage function $\mathbf{g_1}$ that takes $\otimes_{i \in S} m_i$ as input, treats it as $\otimes_{i \in S} m^j_i$. Then, for each $i \in S$, computes $share_i$ as $\big(m^1_{f_1(i)}, \ldots, m^d_{f_d(i)}\big)$ using the fixed values and outputs $\mathbf{g}\big(\otimes_{i \in S} share_i\big)$. Output $S, \mathbf{g_1}$.

Observe that if the adversary for the (p+1)-out-of-n secret sharing scheme can distinguish in between shares of m_1 and m_2 with advantage greater than $d\epsilon$, then the above reduction can distinguish in between the shares corresponding to m_1 and m_2 with advantage greater than ϵ. This violates the leakage-resilience of the underlying (p+1)-out-of-(p+1) scheme, completing the proof. □

5 LRSS for general access structures

In this section, we use any (p+1)-out-of-n secret sharing scheme that is leakage-resilient w.r.t. $(p, n, \mu) - BCP$ and any secret sharing scheme comprising of authorized sets of size at least $p+1$

to construct another secret sharing scheme, such that the resulting scheme not only supports the same access structure, but is also leakage-resilient w.r.t. $(p, n, \mu) - BCP$.

Theorem 3. *For any collusion bound $p \geq 1$, any access structure $\mathcal{A}$ supported on n parties such that each authorized set has cardinality greater than p, any message size $a > 0$, any leakage bound μ, suppose we have the following primitives :*

1. *For any error $\epsilon_1 > 0$, let $(\mathbf{AShare}, \mathbf{ARec})$ be a (n, ϵ_1)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that shares secrets of length a bits into n shares, each of length b bits.*

2. *For any error $\epsilon_2 > 0$, let $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$ be any $(p+1, n, 0)$-secret sharing scheme that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu) - BCP$ and shares secrets of length a bits into n shares each of length c bits.*

Then there is a (n, ϵ_1)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. The resulting scheme, $(\mathbf{LRShare}, \mathbf{LRRec})$, shares secrets of length a into n shares, each of length $b + c$ bits.

Proof. Let $(\mathbf{XORShare_2}, \mathbf{XORRec_2})$ be the $(2, 2, 0)$ additive secret sharing scheme for single bit secrets (as in Lemma 2). The construction of $(\mathbf{LRShare}, \mathbf{LRRec})$ is given below :

- **Sharing function LRShare**:
  Encode the secret input m using the 2-out-of-2 sharing scheme. Let $l, r \leftarrow \mathbf{XORShare_2}(m)$. Share l using the given secret sharing scheme for access structure $\mathcal{A}$ to obtain $l_1, \ldots, l_n \leftarrow \mathbf{AShare}(l)$. Share r using the $(p + 1)$-out-of-n leakage-resilient secret sharing scheme to obtain $r_1, \ldots, r_n \leftarrow \mathbf{LRShare_n^{p+1}}(r)$. Then for each $i \in [n]$, construct $share_i$ as (l_i, r_i).

- **Reconstruction function LRRec**:
  On input the shares $\otimes_{i \in T} share_i$ corresponding to an authorized set T, for each $i \in T$, parse $share_i$ as (l_i, r_i). Run the reconstruction procedure $\mathbf{ARec}$ on the shares of l, to obtain $l \leftarrow \mathbf{ARec}(\otimes_{i \in T} l_i)$. Run the reconstruction procedure of the leakage-resilient scheme on the shares of r, to obtain $r \leftarrow \mathbf{LRRec_n^{p+1}}(\otimes_{i \in T} r_i)$. Run the reconstruction procedure of the 2-out-of-2 sharing scheme to obtain : $m \leftarrow \mathbf{XORRec_2}(l, r)$. Output m.

Correctness and efficiency : Follows easily from the construction.

Perfect (resp. Statistical, Computational) secrecy : Any unauthorized set of shares of our scheme will only have an unauthorized set of shares of l, and therefore by the perfect (resp. statistical, computational) privacy of $(\mathbf{AShare}, \mathbf{ARec})$, l remains hidden. Therefore, the secret remains hidden by the perfect privacy of $(\mathbf{XORShare_2}, \mathbf{XORRec_2})$.

Statistical leakage-resilience: Suppose the adversary specifies a protocol $\mathsf{Leak} \in (p, n, \mu) - BCP$ that violates the leakage-resilience of our scheme using at most μ bits of leakage. We use such an adversary to give an explicit leakage protocol $\mathsf{Leak}_1 \in (p, n, \mu) - BCP$ of the underlying (p+1)-out-of-n scheme, where each party $i \in [n]$ holds a $r_i \in \{0, 1\}^b$ as input.

- **Initial setup** : Randomly fix $l \leftarrow \{0, 1\}^a$. Compute and fix $l_1, \ldots, l_n \leftarrow \mathbf{AShare}(l)$.

- **Reduction** Next_1 : Using Leak, as specified by its Next function and fixed values of shares of l we give the description of protocol Leak_1 by specifying Next_1.

  On input a transcript τ, execute the adversary specified Next function with τ as input to obtain a subset $S \subseteq [n]$ and a leakage function $\mathbf{g}$ that takes $\otimes_{i \in S} share_i$ as input. If the output of Next is $\perp$, then output $\perp$. Otherwise, we construct leakage function g_1 that takes $\otimes_{i \in S} r_i$ as input, and outputs $g\big(\otimes_{i \in S} (l_i \circ r_i)\big)$. Output S, g_1.

Observe that if the adversary for our secret sharing scheme can distinguish between shares of $m_1, m_2 \in \{0,1\}^a$ with advantage greater than ϵ_2, then the above reduction can distinguish between the shares corresponding to $(\mathbf{XORRec_2}(m_1, l)$ and $(\mathbf{XORRec_2}(m_2, l)$ with the same advantage. This violates the leakage-resilience of the underlying (p+1)-out-of-n scheme, and thus our proof is complete. □

5.1 From Single-bit Secrets to Multi-bit Secrets

Using single bit schemes, we give a construction for multi-bit secrets.

Lemma 6. *For any collusion bound $p \geq 1$, any access structure $\mathcal{A}$ supported on n parties such that each authorized set has cardinality greater than p, any leakage-bound μ, any $\epsilon_1 \geq 0$, any $\epsilon_2 > 0$, suppose* $(\mathbf{SBShare}, \mathbf{SBRec})$ *is a (n, ϵ_1)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu) - BCP$ that shares single bit secrets into n shares, each of length a. Then, for any secret space of $b > 0$ bits, there is an efficient $(n, b\epsilon_1)$-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is $b\epsilon_2$-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. The resulting scheme,* $(\mathbf{MBShare}, \mathbf{MBRec})$*, shares secrets of bit-length a into n shares, each of bit-length ab.*

Proof. The construction of $(\mathbf{MBShare}, \mathbf{MBRec})$ follows :

- (**Sharing function MBShare**) :
  On input $m \in \{0,1\}^b$, parse m as $m^1 \circ \ldots \circ m^b$. For each $j \in [b]$, share m^j using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m^j_1, \ldots, m^j_{p+1} \leftarrow \mathbf{SBShare}(m^j)$. For each $i \in [p+1]$, construct $share_i$ as $(m^1_i \circ \ldots \circ m^b_i)$.

- (**Reconstruction function MBRec**) :
  On input $p+1$ shares, for each $i \in [p+1]$, parse $share_i$ as $(m^1_i \circ \ldots \circ m^b_i)$. For each $j \in^b$, use the reconstruction procedure of underlying leakage resilient scheme to compute $m^j \leftarrow \mathbf{SBRec}(m^j_1, \ldots, m^j_{p+1})$. Output $m \leftarrow m^1 \circ \ldots \circ m^b$.

Perfect correctness, statistical privacy and efficiency : Correctness and efficiency trivially follows. It is not hard to use a hybrid argument to arrive at statistical privacy.

Statistical leakage-resilience: The adversary specifies a leakage-protocol $\mathsf{Next} \in (p, n, \mu) - BCP$ that allows it to distinguish in between shares of c and d under our multi-bit scheme. We use such an adversary to construct $\mathsf{Next}_1 \in (p, n, \mu) - BCP$ that violates the leakage-resilience of the single bit scheme.

- **Initial setup** : Randomly fix a bit location $k \in^b$, such that $c^k \neq d^k$. For each $j \in \{1, \ldots, k-1\}$, share c^j using the sharing procedure of underlying single bit scheme (using independent randomness) to obtain $m_1^j, \ldots, m_{p+1}^j \leftarrow \mathbf{SBShare}(c^j)$. Similarly, for each $j \in \{k+1, \ldots, b\}$, share d^j to obtain $m_1^j, \ldots, m_{p+1}^j \leftarrow \mathbf{SBShare}(d^j)$. Fix all these sampled shares.
- **Reduction** Next_1 : On input a transcript τ, execute the adversary specified Next function with τ as input to obtain a subset $S \subset [n]$ and a leakage function $\mathbf{g}$ that takes $\otimes_{i \in S} share_i$ as input. If the output of Next is $\perp$, then output $\perp$. Otherwise, we construct leakage function $\mathbf{g_1}$ that takes $\otimes_{i \in S} m_i$ as input, treats it as $\otimes_{i \in S} m_i^k$. Then, for each $i \in R$, computes $share_i$ as $(m_i^1, \ldots, m_i^b)$ and outputs $\mathbf{g}(\otimes_{i \in R} share_i)$. Output R, g_1.

Observe that if the adversary for the multi-bit scheme can distinguish in between shares of c and d with advantage greater than $b\epsilon$, then the above reduction can distinguish in between the shares corresponding to $c[i]$ and $d[i]$ with advantage greater than ϵ. This violates the leakage-resilience of the single bit scheme, and therefore completes the proof. □

5.2 Instantiations

Corollary 8. *For any collusion bound $p \geq 1$, any access structure $\mathcal{A}$ supported on n parties such that each authorized set has cardinality greater than p, any message size $k > 0$, any leakage bound μ, any error $\epsilon_1 \geq 0$, any error $\epsilon_2 > 0$, suppose there is a (n, ϵ_1)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that shares secrets of length k bits into n shares, each of length b bits. Then there is an (n, ϵ_1)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. The resulting scheme shares secret of length k bits into n shares, each of length $b + k\log(n)(\mu + \log(1/\epsilon_0))2^{O(p)}$ where $\epsilon_0 \leftarrow \epsilon_2/(k\log(n)2^{O(p)})$.*

Proof. We iteratively instantiate the primitives required for Theorem 3 below :

1. For Lemma 1, we let $\mathbf{f}$ be the $p+1$ party generalized inner-product functionality from Babai et al. [BNS92]. For error $\epsilon_0 > 0$, and leakage-bound μ, for single bit secrets, we get that the length of each share of $(\mathbf{SBShare}, \mathbf{SBRec})$ will be $2^{O(p)}(\mu + \log(1/\epsilon_0))$.
2. We use the previous scheme in Lemma 6, to obtain $(\mathbf{MBShare}, \mathbf{MBRec})$ that shares k bit secrets into $k2^{O(p)}(\mu + \log(1/\epsilon_0))$ bit shares. The error of the resulting scheme is $k\epsilon_0$.
3. For Lemma 5, we let $\mathbf{PHF}(p+1, n)$ be the perfect hash family of size $2^{O(p)}\log(n)$ from the work of Naor et al. [NSS95]. We use the previous multi-bit scheme along with $\mathbf{PHF}(p+1, n)$ to obtain a (p+1)-out-of-n scheme, $(\mathbf{LRShare_n^{p+1}}, \mathbf{LRRec_n^{p+1}})$, that shares secret of length k bits into shares of length $k\log(n)(\mu + \log(1/\epsilon_0))2^{O(p)}$. The error of the resulting scheme is $k\log(n)\epsilon_0 2^{O(p)}$.
4. We use the previous (p+1)-out-of-n scheme in Theorem 3 to obtain our final scheme. We want our resultant scheme to have error $\epsilon_2 \leftarrow k\log(n)\epsilon_0 2^{O(p)}$. Therefore, we set $\epsilon_0 \leftarrow \epsilon_2/(k\log(n)2^{O(p)})$.

□

Corollary 9. *For any number of parties $n \geq 2$, any collusion bound $p = O(\log n)$, any threshold $t > p$, any leakage-bound μ, any error $\epsilon > 0$, there is a efficient $(t, n, 0)$-secret sharing scheme that is ϵ-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. The resulting scheme shares k bit secrets into $\mathrm{poly}(n, k, \mu, \log(1/\epsilon))$ bits shares.*

Proof. Use $(t, n, 0)$-secret sharing scheme of Shamir [Sha79] in Corollary 8. □

It is straightforward to use secret sharing schemes of [KW93, Bei11, KNY14] to obtain corresponding corollaries mentioned in the introduction, and consequently we omit these details.

5.3 Leaking p Shares at the Cost of One Extra Bit of Leakage.

We note that that we can empower the adversary to completely leak any p shares at the end of the its leakage protocol, and still ensure leakage-resilience. This observation will prove crucial later while constructing leakage-resilient non-malleable secret sharing scheme in section 7.

Lemma 7. *Any secret sharing scheme on n parties that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu + 1) - BCP$ is also ϵ_2-leakage-resilient against an adaptive adversary who completely leaks any p shares after executing a leakage-protocol $\mathsf{Leak} \in (p, n, \mu) - BCP$.*

Proof. (Sketch) : We can prove this via contradiction. In particular, we can use the distinguisher $\mathcal{D}$ violating leakage-resilience in this new model, to adaptively compute the last bit of leakage and violate leakage-resilience of the underlying scheme. □

6 LRSS implies Complexity Lower Bounds.

While in previous sections, we relied on communication complexity lower bounds to construct leakage-resilient schemes, in this section we make the simple observation that leakage-resilient schemes also imply communication complexity lower bounds.

Lemma 8. *Suppose there is an efficient (n, ϵ_1)-secret sharing scheme for single bit secrets that is ϵ_2-leakage-resilient w.r.t. $(p, n, \mu) - BCP$. Let $\mathbf{Rec} : (\{0,1\}^b)^n \to \{0,1\}$ be the reconstruction procedure of the secret sharing scheme. Then, computing $\mathbf{Rec}$ with advantage better than ϵ_2 (over random guessing) requires communication complexity at least μ bits for any communication protocol which allows any collection of p parties to speak in any round.*

Proof. Follows immediately from the definition of leakage-resilience. □

In particular, the above observation has the following corollary:

Corollary 10. *Suppose there is an efficient construction of (n, n, ϵ_1)-secret sharing scheme that is ϵ_2-leakage-resilient w.r.t. $(n-1, n, \mu) - BCP$, then the ϵ_2-NOF communication complexity of the reconstruction procedure is at least μ ($\mathbf{CC}_{\mathbf{n}}^{\mathit{NOF}}(\mathbf{Rec}) \geq \mu$).*

Proving lower bounds on the NOF communication complexity of explicit functions where the number of parties is super-logarithmic in the input length is one of the most outstanding challenges in complexity theory with many eminent implications. In particular, if the size of each of the shares in a LRSS as above is $k \leftarrow o(\mu 2^n)$ bits, then the NOF communication complexity of the reconstruction function (a function on $(\{0,1\}^k)^n$) would be $\omega(k/2^n)$. While there have been numerous attempts to obtain a lower bound of the form $\omega(k/2^n)$, all known attempts are only able to achieve $\Omega(l/2^n)$ [BNS92,Chu90,Raz00,She14]. Even handling *non-adaptive* adversaries is a challenge. This can be seen from the classical results of Yao [Yao90] and Hastad and Goldmann [HG91] who showed that (simultaneous) NOF communication complexity lower bounds imply circuit lower bounds. In our setting if we further limit the adversary and only allow for non-adaptive leakage, then we can obtain lower bounds for depth 3 threshold circuits.

Corollary 11. *For single bit secrets, for any number of parties $n \geq 2$, suppose there is $(n-1, n, n, \mu)$-LRSS leakage-resilient w.r.t a computationally unbounded adversary who learns n^2 bits of leakage such that each bit of the leakage non-adaptively depends on at most $n-1$ shares. If the size of each of the shares of this scheme is $2^{n^{o(1)}}$, then this implies that the reconstruction procedure of this scheme does not belong to $\mathbf{ACC^0}$.*

Proof. Follows by combining the arguement of Hastad and Goldmann [HG91] and Yao [Yao90]. □

In particular, if a $(n-1, n, n, \mu)$-LRSS is efficient (efficiency implies that the shares are of size poly(n, μ) bits), then we obtain an unconditional separation between $\mathbf{P}$ and $\mathbf{ACC^0}$. Contrast this with the celebrated result of Williams [Wil14] which unconditionally separates $\mathbf{NEXP}$ from $\mathbf{ACC^0}$ (using different techniques) and the recent result of Murray and Williams [MW18] separating quasi-non-deterministic poly $\mathbf{NQP}$ from $\mathbf{ACC^0}$.

The above discussion suggests that improving Corollary 9 to obtain efficient (p, t, n)-LRSS for $p = \omega(\log n)$ could be considerably harder.

7 Leakage-Resilient Non-Malleable Secret Sharing

In this section we convert any secret sharing scheme into another one that additionally ensures non-malleability against an adversary who arbitrarily learns a bounded amount of information via a number-in-hand leakage-protocol and then uses this leakage to arbitrarily tamper each of the shares independently.

We recall the definition of non-malleable secret sharing from [GK18b].

Definition 11. *(**Non-Malleable Secret Sharing Schemes** [GK18a, GK18b]) Let $\mathcal{A}$ be some access structure. Let $\mathcal{A}^{min}$ be its corresponding minimal basis access structure. Let $(\mathbf{Share}, \mathbf{Rec})$ be any (n, ϵ)-secret sharing scheme realizing access structure $\mathcal{A}$ for message space $\mathcal{M}$. Let $\mathcal{F}$ be some family of tampering functions. For each $\mathbf{f} \in \mathcal{F}$, $m \in \mathcal{M}$ and $T \in \mathcal{A}^{min}$, define the tampering experiment*

$$\mathbf{STamper_m^{f,T}} = \left\{ \begin{array}{c} shares \leftarrow \mathbf{Share}(m) \\ \widetilde{shares} \leftarrow \mathbf{f}(shares) \\ \tilde{m} \leftarrow \mathbf{Rec}(\widetilde{shares}_T) \\ Output : \tilde{m} \end{array} \right\}$$

which is a random variable over the randomness of the sharing function $\mathbf{Share}$. *We say that the (n, ϵ)-secret sharing scheme, $(\mathbf{Share}, \mathbf{Rec})$, realizing access structure $\mathcal{A}$ is ϵ'-**non-malleable** w.r.t $\mathcal{F}$ if for each $\mathbf{f} \in \mathcal{F}$ and authorized $T \in \mathcal{A}^{min}$, there exists a distribution $\mathbf{SD^{f,T}}$ (corresponding to* *the simulator) over $\mathcal{M} \cup \{same^*, \perp\}$ such that, for all $m \in \mathcal{M}$, we have that the statistical distance between $\mathbf{STamper_m^{f,T}}$ and*

$$\mathbf{SSim_m^{f,T}} = \left\{ \begin{array}{c} \tilde{m} \leftarrow \mathbf{SD^{f,T}} \\ Output : m \; if \, \tilde{m} = same^*, or \; \tilde{m}, otherwise \end{array} \right\}$$

is at most ϵ'. Additionally, $\mathbf{SD^{f,T}}$ should be efficiently samplable given oracle access to $\mathbf{f}$(.)

We recall the split-state (individual) tampering family from [GK18a]:

7.0.1 Individual Tampering Family $\mathcal{F}_n^{split}$

Let **Share** be any sharing function that takes a secret as input and outputs n shares, namely $share_1, \ldots, share_n$. For each $i \in [n]$, let $\mathbf{f_i} : \mathcal{S} \to \mathcal{S}$ be an arbitrary tampering function, that takes as input $share_i$ (the i^{th} share) and outputs $\widetilde{share_i}$ (the tampered i^{th} share). Let $\mathcal{F}_n^{split}$ denote the family containing all such tampering functions, namely $(\mathbf{f_1}, \ldots, \mathbf{f_n})$.

We generalize this family to encompass leakage:

7.0.2 Individual Leakage Tampering Family $\mathcal{F}_{n,\mu}^{ind-leak}$

Let the n shares be $share_1, \ldots, share_n$ be as in the definition $\mathcal{F}_n^{split}$. Let $\mathsf{Leak} \in (1, n, \mu) - BCP$ be any number-in-hand leakage protocol that adaptively leaks at most μ bits of information about the n shares. Let $\tau \leftarrow \mathsf{Leak}(share_1, \ldots, share_n)$ denote the transcript of this leakage. The adversary uses this leakage to tamper each of the n shares arbitrarily and independently. More formally, for each $i \in [n]$, let $\mathbf{f_i} : \mathcal{S} \times \{0,1\}^\tau \to \mathcal{S}$ be an arbitrary tampering function, that takes as input $share_i$ and τ (leakage transcript) to output $\widetilde{share_i}$. Let $\mathcal{F}_{n,\mu}^{ind-leak}$ denote the family containing all such leakage and tampering functions, namely $(\mathsf{Leak}, \mathbf{f_1}, \ldots, \mathbf{f_n})$.

Access structures based definitions. We recall some defintions from [GK18b].

Definition 12. *(**Minimal basis access structure** [GK18b]) For any access structure $\mathcal{A}$, we define **minimal basis access structure** of $\mathcal{A}$, denoted by $\mathcal{A}^{min}$, as the the minimal subcollection of $\mathcal{A}$, such that for all authorized set $T \in \mathcal{A}$, there exists an authorized subset $B \subseteq T$ which is an element of $\mathcal{A}^{min}$.*

Definition 13. *(**Paired access structures** [GK18b]) An access structure $\mathcal{A}$ is called a **paired access structure**, if each authorized set contains an authorized subset of size two. Formally, for all $B \in \mathcal{A}$, there exists a subset $C \subseteq B$ such that C is authorized and has cardinality two.*

Notice that, if $\mathcal{A}$ is a paired access structure then its corresponding minimal basis access structure $\mathcal{A}^{min}$ will only contain authorized sets of size two.

Definition 14. *(**Authorized paired access structures** [GK18b]) For any access structure $\mathcal{A}$, we call a paired access structure $\mathcal{A}_{pairs}$ an **authorized paired access structure** corresponding to $\mathcal{A}$ if $\mathcal{A}_{pairs}$ is the maximal subcollection of $\mathcal{A}$. Formally,*

$$\mathcal{A}_{pairs} = \{B \in \mathcal{A} : \exists C \subseteq B, (C \in \mathcal{A}) \wedge (|C| = 2)\}$$

Notice that $\mathcal{A}_{pairs}^{min}$ will be equal to the set of all the authorized sets of size two in $\mathcal{A}$.

Leakage-resilient NMSS scheme for authorized pairs. Goyal and Kumar [GK18b] also constructed a NMSS scheme for authorized pairs by giving every authorized pair an encoding of the secret under a 2-out-of-2-NMSS. Analogously, we can give every authorized pair an encoding of the secret under a 2-out-of-2-LR-NMSS [ADKO15] to obtain a leakage-resilient NMSS scheme for authorized pairs. We sketch the proof for non-malleability: suppose we will use the first two shares for reconstruction. Suppose the adversary leaks from all the n shares, tampers with all the n shares independently. In our reduction, we generate n 'fake' shares encoding a 'fake' secret 0. The two real shares of the 2-out-of-2 scheme can simulate the leakage by replace the specific components of the first two 'fake' shares with the two 'real' shares and run the adversarial leakage-protocol on all the n resulting shares to obtain a leakage-transcript. This transcript is then used to tamper both the real shares independently. This completes the reduction. Using a hybrid argument we can show swap every pair of 'fake' shares with their real shares, without affecting the output of the tampering experiment. After all these 'pairs' are replaced we end up with the real tampering experiment, completing the proof.

Efficient membership queries. To achieve the general result, we need one to recall one more definition. We say that an access structure supports efficient membership queries, if we can efficiently decide whether the given set of identities of parties is authorized or not. As an example, given any access structure, we can check every pair of parties to see if the pair in hand is authorized or not, and therefore efficiently construct the corresponding paired access structure. Another way to model this is via a membership oracle.

Main result for general access structures. We are now in position to give our main result.

Theorem 4. *For any number of parties n, and any access structure $\mathcal{A}$ that does not contain singletons, any leakage bound τ, if we have the following primitives :*

1. *For any $\epsilon_0 \geq 0$, $\epsilon_1 > 0$, let $(\mathbf{NMEnc}, \mathbf{NMDec})$ be any $(2, 2, \epsilon_0)$-secret sharing scheme that is ϵ_1-non-malleable w.r.t. $\mathcal{F}_2^{split}$, which encodes an element of the set $\mathbb{F}_0$ into two elements of $\mathbb{F}_1$.*

2. *For any $\epsilon_2 \geq 0$, $\epsilon_3 \geq 0$, let $(\mathbf{LShare}, \mathbf{LRec})$ be any (n, ϵ_2)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is ϵ_3-leakage resilient w.r.t. $(2, n, \mu + 1) - BCP$, which shares an element of $\mathbb{F}_1$ into n elements of $\mathbb{F}_2$.*

3. *Let $\mu_1 \leftarrow \mu + n \log |\mathbb{F}_2|$. For any $\epsilon_4 \geq 0$, $\epsilon_5 > 0$ let $(\mathbf{RShare}, \mathbf{RRec})$, be any $(2, n, \epsilon_4)$-secret sharing scheme that is ϵ_5-leakage-resilient w.r.t. $(1, n, \mu_1) - BCP$, which shares an element of $\mathbb{F}_1$ into n elements of the $\mathbb{F}_3$.*

4. *Let $\mu_2 \leftarrow \mu + n \log |\mathbb{F}_2| + 2 \log |\mathbb{F}_3|$. For any $\epsilon_6 \geq 0$, $\epsilon_7 > 0$, let $(\mathbf{PShare}, \mathbf{PRec})$, be any (n, ϵ_6)-secret sharing scheme realizing the authorized paired access structure $\mathcal{A}_{pairs}$ that is ϵ_7-leakage-resilient non-malleable w.r.t. $\mathcal{F}_{n,\mu_1}^{ind-leak}$, which shares an element of the set $\mathbb{F}_0$ into n elements of $\mathbb{F}_4$.*

then there exists $(n, \epsilon_0 + \epsilon_2 + \epsilon_4 + \epsilon_6)$-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is $(\epsilon_1 + \epsilon_3 + \epsilon_5 + \epsilon_7)$-leakage-resilient non-malleable w.r.t $\mathcal{F}_{n,\mu}^{ind-leak}$. The resulting scheme, $(\mathbf{NMShare}, \mathbf{NMRec})$, shares an element of the set $\mathbb{F}_0$ into n shares where each *share is an element of* $(\mathbb{F}_2 \times \mathbb{F}_3 \times \mathbb{F}_4)$*. Further, if the four primitives have efficient construction (polynomial time sharing and reconstruction functions) and the access structure* $\mathcal{A}$ *supports efficient membership queries, then the constructed scheme is also efficient.*

Proof. In our constructions, we need a method to find a minimal authorized set given any authorized set. For any access structure $\mathcal{A}$ not containing singletons, recall the efficient deterministic procedure from [GK18b] **FindMinSet** : $\mathcal{A} \to \mathcal{A}^{min}$, which takes an authorized set and outputs a minimal authorized set contained in that set. Our construction follows :

- **Sharing function NMShare**: Encode the secret $m \in \mathbb{F}_1$ using **NMEnc** to obtain l, $r \leftarrow \mathbf{NMEnc}(m)$. Share l using a **LShare** to obtain $l_1, \ldots, l_n \leftarrow \mathbf{LShare}(l)$. Share r using **RShare** to obtain $r_1, \ldots, r_n \leftarrow \mathbf{RShare}(r)$. Share m using **PShare** to obtain $(p_1, \ldots, p_n) \leftarrow \mathbf{PShare}(m)$. Then for each $i \in [n]$, construct $share_i$ as l_i, r_i, p_i.

- **Reconstruction function NMRec**: On input the shares $\otimes_{i \in D} share_i$ corresponding to authorized set $\mathcal{D}$, for each $i \in D$, parse $share_i$ as (l_i, r_i, p_i). Find the minimal authorized set $T \in \mathcal{A}^{min}$ by running the procedure **FindMinSet** with input $\mathcal{D}$. Let T be a set containing t indices $\{i_1, i_2, \ldots, i_t\}$ such that $i_j < i_{j+1}$ for each $j \in [t-1]$. If $|T| = 2$, use the decoding procedure **PRec** to obtain the hidden secret $m \leftarrow \mathbf{PRec}(p_{i_1}, p_{i_2})$. Otherwise, run the reconstruction procedure **LRec** on t shares of l, to obtain $l \leftarrow \mathbf{LRec}(\otimes_{i \in T} l_i)$. Run the reconstruction procedure **RRec** on the first 2 shares of r, to obtain $r \leftarrow \mathbf{RRec}(r_{i_1}, r_{i_2})$. Decode l and r using **NMDec** to obtain : $m \leftarrow \mathbf{NMDec}(l, r)$. Output m.

Correctness and efficiency : Follows trivially.

Statistical (resp. Computational) Privacy : The proof is the same as [GK18b].

Statistical non-malleability : Without loss of generality we can assume that adversary chooses an authorized set $T \in \mathcal{A}^{min}$ to be used for reconstruction of the secret, as otherwise we can use the function **FindMinSet** to compute $T \in \mathcal{A}^{min}$ from any $D \in \mathcal{A}$. As the adversary belongs to $\mathcal{F}_{n,\mu}^{ind-leak}$, it specifies a leakage protocol Leak and a set of n tampering functions $\{\mathbf{f_i} : i \in [n]\}$. Recall that Leak produces a leakage transcript τ and each function f_i takes $share_i$ and τ as input and outputs the tampered $\widetilde{share_i}$. We can also assume without loss of generality that all these tampering functions are deterministic, as the computationally unbounded adversary can compute the optimal randomness.

To prove leakage-resilient non-malleability of our scheme, we use the adversarily specified leakage and tampering functions to create explicit functions violating the non-malleability of the underlying non-malleable secret-sharing schemes. Like [GK18a], depending on the cardinality of T we get two cases :

Case 1 ($|T| = 2$) :
Let i_1 and i_2 be the two indices of T such that $i_1 < i_2$. In this case, we use the leakage function Leak and tampering functions $\mathbf{f_1}, \mathbf{f_2}$ for the scheme $(\mathbf{NMShare}, \mathbf{NMRec})$ to create explicit leakage function Leak_1 and tampering functions $\mathbf{F}_{i_1}$ and $\mathbf{F}_{i_2}$ for the underlying scheme $(\mathbf{PShare}, \mathbf{PRec})$. The reduction is described below :

1. **(Initial setup)** : Fix any message $m_\$ \in \mathcal{M}$, and run the sharing function **NMShare** with input $m_\$$ to obtain 'fake' shares. That is, $(tShare_1, \ldots, tShare_n) \leftarrow \mathbf{NMShare}(m_\$)$. For each $i \in [n]$, parse $tShare_i$ as tl_i, tr_i, tp_i and fix tl_i, tr_i.

2. **(Leakage function)** Leak_1 : We now design a n party leakage protocol Leak_1 using the given n party leakage protocol Leak. For this, it suffices to construct the corresponding Next_1 function. Let τ denote the transcript (initially empty). On input transcript τ, the function Next_1 invokes the underlying next function to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, g \leftarrow \mathsf{Next}(\tau)$. Then it uses the leakage function $\mathbf{g}(share_i)$ to define the leakage function $\mathbf{g}_1(p_i)$ as follows : On input p_i, output $\mathbf{g}(tl_i, tr_i, p_i)$. The Next_1 function outputs i, g_1. In case Next outputs $\perp$, Next_1 also outputs $\perp$ completing the leakage protocol Leak_1. Denote the final output of the leakage protocol Leak_1 as $\tau \leftarrow \mathsf{Leak}_1(p_1, \ldots, p_n)$.

3. For each $i \in [n]$, **Tampering function** $\mathbf{F}_i$ is defined as follows : On input $p_i \in \mathbb{F}_4$ and leakage transcript $\tau \in \{0,1\}^\mu$, let $share_i \leftarrow tl_i, tr_i, p_i$. Run $\mathbf{f_i}$ on $share_i$ and transcript τ to obtain tampered $\widetilde{share_i}$. Parse $\widetilde{share_i}$ as $\widetilde{l_i}, \widetilde{r_i}, \widetilde{p_i}$. Output $\widetilde{p_i}$.

To prove non-malleability of our scheme, our hope is to rely on the simulator of (**PShare**, **PRec**) whose output distribution is statistically close to the distribution of the tampered secret produced by the above reduction. To this end, we have to show that distribution of the tampered secret produced in the reduction is statistically close to the one produced in the real tampering experiment. We achieve this using the following hybrid argument :

1. $\mathbf{Hybrid_1}$: The distribution of the tampered secret is identical to the distribution of the tampered secret produced by the above reduction. To recall, create $tShare_i$ using 'fake' shares generated using $m_\$$. Let $\tau \leftarrow \mathsf{Leak}_1(p_1, \ldots, p_n)$. For each $i \in \{i_1, i_2\}$, compute $\widetilde{p_i} \leftarrow \mathbf{F}_i(p_i, \tau)$. Output $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$.

2. $\mathbf{Hybrid_2}$: We only make one change in the previous hybrid. In the initial setup the fixed shares of $r_\$$ are replaced with real shares of r (produced while encoding m instead of $m_\$$). Output $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$.

3. $\mathbf{Hybrid_3}$: We only make one change in the previous hybrid. In the initial setup the fixed shares of $l_\$$ are replaced with real shares of l (produced while encoding m instead of $m_\$$). Output $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$. Note that this is identical to the distribution of the tampered secret in the real tampering experiment.

Claim: For any $r, r_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ is at most ϵ_0.
Proof: The two hybrids differ in the intial setup phase. In $\mathbf{Hybrid_2}$, shares of $r_\$$ are fixed, while in $\mathbf{Hybrid_3}$ shares of r are fixed. We can use any distinguisher for these two hybrds to construct a distinguisher voilating the statistical secrecy of $(2, 2, \epsilon_0)$ secret sharing scheme (**NMEnc**, **NMDec**). In more detail,

1. **Initial setup** : Let $tl_1, \ldots, tl_n \leftarrow \mathbf{LShare}(l_\$)$ and $p_1, \ldots, p_n \leftarrow \mathbf{PShare}(m)$.

2. **Distinguisher** : On input r, sample $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(r)$. Proceed, as in the reduction to obtain a transcript τ using leakage protocol Leak_1. For each $i \in \{i_1, i_2\}$, compute $\widetilde{p_i} \leftarrow \mathbf{F}_i(p_i, \tau)$. Invoke the distinguisher with $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$ and output its output.

It is immediate that the two distinguishers have the same distinguishing advantage, completing the proof of the claim. ■

Claim: For any $l, l_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_2}$ and $\mathbf{Hybrid_3}$ is at most ϵ_3.
Proof: Assume towards contradiction that there exists $l, l_\$ \in \mathbb{F}_1$, and a distinguisher $\mathcal{D}$ that is successful in distinguishing $\mathbf{Hybrid_2}$ and $\mathbf{Hybrid_3}$ with probability greater than ϵ_3. We use the reduction and such a distinguisher to construct a leak protocol $\mathsf{Leak}_2 \in (1, n, \mu) - BCP$ and another distinguisher D_1 that violates the leakage-resilience of the scheme $(\mathbf{LShare}, \mathbf{LRec})$ for the secrets $l, l_\$$. The reduction is described below :

1. (**Initial setup**) : Let $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(r)$ and $p_1, \ldots, p_n \leftarrow \mathbf{PShare}(m)$.
2. (**Leak function** Leak_2) : We now design a n party leakage protocol Leak_2 for (**LShare**, **LRec**) using the given n party leakage protocol Leak for $(\mathbf{NMShare}, \mathbf{NMRec})$. To this end, it suffices to construct the corresponding Next_2 function. Let τ denote the transcript (initially empty). On input transcript τ, the function Next_2 invokes the underlying next function to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, g \leftarrow \mathsf{Next}(\tau)$. Then it uses the leakage function $\mathbf{g}$ to define the leakage function $\mathbf{g}_2(l_i)$ as follows : On input l_i, output $\mathbf{g}(l_i, tr_i, p_i)$. The Next_2 function outputs i, g_2. Let τ denote the transcript, when the leakage protocol Leak finishes (formalized by Next outputting $\perp$). At this point, we continue our leakage protocol and party i_1 and i_2 completely leak l_{i_1} and l_{i_2} in the next two rounds. As a result, the final transcript of our leakage protocol Leak_2 will be $\tau \circ l_{i_1} \circ l_{i_2}$.

   As τ is at most μ bits and from Corollary 7 upto two shares of l can be fully leaked, the above leakage protocol belongs to the class $(2, n, \mu) - BCP$.

3. (**Distinguisher** D_1) : On input leakage transcript $\tau \circ l_{i_1} \circ l_{i_2}$, for each $i \in \{i_1, i_2\}$, compute $\widetilde{p_i} \leftarrow \mathbf{f}_i(l_i \circ tr_i \circ p_i, \tau)[3]$ ([3] denotes the third component of vector). Invoke the distinguisher with $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$ and output its output.

Notice, in the case the secret hidden by the leakage-resilient scheme $(\mathbf{LShare}, \mathbf{LRec})$ is $l_\$$, $\mathcal{D}$ will be invoked with input distributed according to $\mathbf{Hybrid_2}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_3}$. Therefore the success probability of D_1 will be equal to the advantage of $\mathcal{D}$ in distinguishing these two hybrids, which is greater than ϵ_3 by assumption. Hence, we have arrived at a contradiction to statistical leakage-resilience of the scheme $(\mathbf{LShare}, \mathbf{RRec})$. ■

As constructed, the set of tampering functions $\{\mathbf{F}_i : i \in [n]\}$ and the leakage function Leak_1 belongs to $\mathcal{F}_{n,\mu}^{ind-leak}$. Therefore, the tampering experiments of the two non-malleable secret-sharing scheme (see definition 11) are statistically indistinguishable, specifically,

$$\mathbf{STamper_m^{\mathsf{Leak},f,T}} \approx_{\epsilon_0+\epsilon_3} \mathbf{STamper_m^{\mathsf{Leak}_1,F,T}}$$

By the ϵ_7-non malleability of the scheme $(\mathbf{PShare}, \mathbf{PRec})$, there exists a simulator $\mathbf{SSim_m^{\mathsf{Leak}_1,F,T}}$ such that

$$\mathbf{STamper_m^{\mathsf{Leak}_1,F,T}} \approx_{\epsilon_7} \mathbf{SSim_m^{\mathsf{Leak}_1,F,T}}$$

We use the underlying simulator as our simulator, and let

$$\mathbf{SSim_m^{\mathsf{Leak},f,T}} \equiv \mathbf{SSim_m^{\mathsf{Leak}_1,F,T}}$$

Applying triangle inequality to the above relations we prove the statistical non malleability for this case.

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak}\,\mathbf{f},\mathbf{T}} \approx_{\epsilon_0+\epsilon_3+\epsilon_7} \mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}}$$

Case 2 ($|T| \geq 3$) :
Let $T = \{i_1, \ldots i_t\}$ be an ordered set of t indices, such that $i_j < i_{j+1}$. In this case, we use the leakage function Leak and tampering functions $\{\mathbf{f_i} : i \in T\}$ for the scheme $(\mathbf{NMShare}, \mathbf{NMRec})$ to create explicit tampering functions $\mathbf{F}$ and $\mathbf{G}$ that independently tampers the two shares of the underlying 2-out-of-2 non-malleable secret sharing scheme $(\mathbf{NMEnc}, \mathbf{NMDec})$. Note that as $\mathcal{F}_2^{split}$ allows arbitrary computation, the functions $\mathbf{F}$ and $\mathbf{G}$ are allowed to brute force over any finite set. The reduction giving explicit $(\mathbf{F}, \mathbf{G}) \in \mathcal{F}_2^{split}$ is described below.

1. **(Initial setup)** : Fix an arbitrary $m_\$$ and let $l_\$, r_\$ \leftarrow \mathbf{NMEnc}(m_\$)$. Run the sharing function **LShare** with input $l_\$$ to obtain $tl_1, \ldots, tl_n$. Run the sharing function $\mathbf{RShare}(r_\$)$ to obtain $tr_1, \ldots, tr_n$. Run the sharing function **PShare** with input $m_\$$ to obtain $tp_1, \ldots, tp_n$. For each $i \in [n]$, create $tShare_i$ as tl_i, tr_i, tp_i. Run the the leakage protocol on these 'fake' shares to obtain the leakage transcript $\tau \leftarrow \mathsf{Next}(tShare_1, \ldots, tShare_n)$. For each $i \in \{i_1, i_2\}$, run $\mathbf{f}_i$ on $tShare_i$ and transcript τ to obtain $\widetilde{tShare_i} \leftarrow \mathbf{f}_i(tShare_i, \tau)$. Parse $tshare_i$ as $\widetilde{tl_i}, \widetilde{tr_i}, \widetilde{tp_i}$. *Fix* $l_i \leftarrow tl_i$ and $\widetilde{l_i} \leftarrow \widetilde{tl_i}$. For each $i \in \{i_3, \ldots, i_t\}$, *fix* $r_i \leftarrow tr_i$. For all $i \in T$, *fix* $p_i \leftarrow tp_i$.

2. The **tampering function F** is defined as follows : On input l, sample the value of $l_{i_3}, \ldots, l_{i_t}$ such that the shares $\{l_i : i \in T\}$ hide the secret l under $(\mathbf{LShare}, \mathbf{LRec})$ and the distribution of these shares is identical to the distribution produced on running **LShare** with input l. In case such a sampling is not possible, then `abort`. Otherwise, for each $i \in T \backslash \{i_1, i_2\}$, construct $share_i$ as (l_i, r_i, p_i) using the fixed values of r_i and p_i. Run the tampering function f_i with inputs $share_i$ and transcript τ to obtain tampered $\widetilde{share_i} \leftarrow \mathbf{f}_i(share_i)$. Parse $\widetilde{share_i}$ as $\widetilde{l_i}$, $\widetilde{r_i}, \widetilde{p_i}$. Run the reconstruction function **LRec** with input $\otimes_{i \in T}\widetilde{l_i}$ to obtain $\widetilde{l} \leftarrow \mathbf{LRec}(\otimes_{i \in T}\widetilde{l_i})$. Output $\widetilde{l}$.

3. The **tampering function G** is defined as follows : On input r, sample the values of first two shares of r, namely $\{r_{i_1}, r_{i_2}\}$ satisfying the following properties (via brute force over all possibilities) :-

   - The two shares $\{r_{i_1}, r_{i_2}\}$ encode the secret r under the $(\mathbf{RShare}, \mathbf{RRec})$. Moreover, the two shares should be distributed according to the output distribution of scheme $(\mathbf{RShare}, \mathbf{RRec})$.
   - For each $i \in \{i_1, i_2\}$, let $share_i$ be (l_i, r_i, p_i), run f_i with inputs $share_i$ and transcript τ to obtain $\widetilde{share_i}$. Parse $\widetilde{share_i}$ as $(\widetilde{nl_i}, \widetilde{nr_i}, \widetilde{np_i})$. The value of $\widetilde{nl_i}$ should be equal to $\widetilde{l_i}$ (the value that was fixed in the initial step of reduction).

   In case such a sampling is not possible, then `abort`. Otherwise, run the reconstruction procedure of the leakage-resilient scheme to obtain $\widetilde{r}$, using the tampered values of first 2 shares of r. That is $\widetilde{r} \leftarrow \mathbf{LRec}(\widetilde{nr_{i_1}}, \widetilde{nr_{i_2}})$. Output $\widetilde{r}$.

To prove non-malleability of our scheme, our hope is to rely on the simulator of $(\mathbf{NMEnc}, \mathbf{NMDec})$ whose output distribution is statistically close to the distribution of the tampered secret produced in the above reduction. To this end, we have to show that distribution of the tampered secret produced by the reduction is statistically close to the one produced in the real tampering experiment. This is not immediate, because in real tampering experiment, tampering is preceded with leakage where all the n shares are involved. Nevertheless, we achieve this using the following hybrid argument. We begin by fixing any $l_\$, r_\$$ encoding $m_\$$ and any l, r encoding m under the 2-out-of-2 non-malleable scheme ($\mathbf{NMEnc}, \mathbf{NMDec}$).

1. $\mathbf{Hybrid_1}$: The distribution of the tampered secret is identical to the distribution of the tampered secret produced by the above reduction. To recall, we share $l_\$$ to obtain $tl_1, \ldots,$ tl_n. Similarly, we also share $r_\$$ and $p_\$$ using respective schemes. Next we create 'fake' shares $tShare_i \leftarrow tl_i, tr_i, tp_i$. After which, we execute the leakage protocol on these 'fake' shares to obtain the transcript $\tau \leftarrow \mathsf{Leak}(tShare_1, \ldots, tShare_n)$. Use the functions defined in the reduction to compute $\widetilde{l} \leftarrow \mathbf{F}(l)$ and $\widetilde{r} \leftarrow \mathbf{G}(r)$. Output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.

2. $\mathbf{Hybrid_2}$: We only make one change in the preceding hybrid. In the initial setup, the shares $tr_1, \ldots, tr_n$ are generated by sharing r (instead of $r_\$$), that is $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(r)$. Proceed as in preceding hybrid and output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.

3. $\mathbf{Hybrid_3}$: We only make one change in the preceding hybrid. In the initial setup, the tampered $\widetilde{r}$ is computed in the initial setup using $\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}}$ (instead of invoking the tampering function $\mathbf{G}$), that is, $\widetilde{r} \leftarrow \mathbf{RRec}(\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}})$. Proceed as in preceding hybrid and output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.

4. $\mathbf{Hybrid_4}$: We only make one change in the preceding hybrid. In the initial setup, the shares $tl_1, \ldots, tl_n$ are generated by sharing l (instead of $l_\$$), that is $tl_1, \ldots, tl_n \leftarrow \mathbf{LShare}(l)$. Proceed as in preceding hybrid and output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.

5. $\mathbf{Hybrid_5}$: We only make one change in the preceding hybrid. In the initial setup, the tampered $\widetilde{l}$ is computed in the initial setup using $\widetilde{tl_1}, \ldots, \widetilde{tl_n}$ (instead of invoking the tampering function $\mathbf{F}$), that is $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{tl_{i_1}}, \ldots, \widetilde{tl_{i_t}})$. Proceed as in preceding hybrid and output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.

6. $\mathbf{Hybrid_6}$: We only make one change in the preceding hybrid to obtain the current hybrid. In the initial setup, the shares $tp_1, \ldots, tp_n$ are generated by sharing m (instead of $m_\$$), that is $tp_1, \ldots, tp_n \leftarrow \mathbf{PShare}(m)$. Proceed as in preceding hybrid and output $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$. Note that this is identical to the distribution of the tampered secret in the real tampering experiment conditioned on the output of $\mathbf{NMEnc}$ being l, r.

Claim: For any $r, r_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ is at most ϵ_5.
Proof: These two hybrids differ in the intial setup phase. In $\mathbf{Hybrid_1}$ shares of $r_\$$ are fixed, while in $\mathbf{Hybrid_1}$ shares of r are fixed. We can use the adversary and the distinguisher for these two hybrids to construct a leakage-protocol violating the statistical leakage-resilience of the 2-out-of-n secret sharing scheme ($\mathbf{LShare}, \mathbf{LRec}$). The reduction is described below :

1. (**Initial setup**) : Fix $l_1, \ldots, l_n \leftarrow \mathbf{LShare}(l)$ and $p_1, \ldots, p_n \leftarrow \mathbf{PShare}(m_\$)$.

2. (**Leak function** Leak_2) : We now design a n party leakage protocol Leak_2 for (**RShare**, **RRec**) using the given n party leakage protocol Leak for (**NMShare**, **NMRec**). To this end, it suffices to construct the corresponding Next_2 function. Let τ denote the transcript (initially empty). On input transcript τ, the function Next_2 invokes the underlying next function to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, g \leftarrow \mathsf{Next}(\tau)$. Then it uses the leakage function $\mathbf{g}(share_i)$ to define the leakage function $\mathbf{g}_2(r_i)$ as follows : On input r_i, output $\mathbf{g}(l_i, r_i, p_i)$. The Next_2 function outputs i, g_1. Let τ denote the transcript, when the leakage protocol Leak finishes (formalized by Next outputting $\perp$). At this point, we continue and each party $i \in T$ iteratively computes $\widetilde{l_i} \circ \widetilde{r_i} \circ \widetilde{p_i} \leftarrow \mathbf{f}_i(l_i \circ r_i \circ p_i, \tau)$ and outputs as leakage $(\widetilde{l_i}$. As a result, the transcript of our leakage protocol Leak_2 will be $\tau \circ \widetilde{l_{i_1}} \circ \ldots \circ \widetilde{l_{i_t}}$.

   As t shares of l can require at most $n \log |\mathbb{F}_2|$ bits (recall $t \leq n$), the above leakage protocol belongs to the class $(1, n, \mu_1) - BCP$.

3. (**Distinguisher** D_1) : On input $\tau \circ \widetilde{l_{i_1}} \circ \ldots \circ \widetilde{l_{i_t}}$, compute $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{l_{i_1}}, \ldots, \widetilde{l_{i_t}})$ and $\widetilde{r} \leftarrow \mathbf{G}(r)$. Invoke the distinguisher $\mathcal{D}$ with $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$ and output its output.

Notice, in the case the secret hidden under the scheme (**RShare**, **RRec**) is $r_\$$, $\mathcal{D}$ will be invoked with input distributed according to $\mathbf{Hybrid_1}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_2}$. Therefore the success probability of D_1 will be equal to the advantage of $\mathcal{D}$ in distinguishing these two hybrids, which is greater than ϵ_5 by assumption. Hence, we have arrived at a contradiction to statistical leakage-resilience of the scheme (**RShare**, **RRec**). ■

Claim: $\mathbf{Hybrid_2}$ is identical to $\mathbf{Hybrid_3}$.
Proof: These two hybrids differ in how $\widetilde{r}$ is computed. In $\mathbf{Hybrid_2}$, the function $\mathbf{G}$ samples two shares of r, such that r_{i_1} and r_{i_2} satisfy certain constraints. Notice that in $\mathbf{Hybrid_3}$, the fixed values of tr_{i_1} and tr_{i_2} already satisfy all these constraints. Consequently there is no need for sampling and $\widetilde{r}$ can be directly computed using the fixed values. ■

The above two claims also show that $\mathbf{G}(r)$ does not `abort` with probability at least ϵ_5.
Claim: For any $l, l_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$ is at most ϵ_3.
Proof: These two hybrids differ in the initial stage while creating share $tl_1, \ldots, tl_n$. Assume towards contradiction that there exists $l, l_\$ \in \mathbb{F}_1$, and a distinguisher $\mathcal{D}$ that is successful in distinguishing $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$ with probability greater than ϵ_3. We use the reduction and such a distinguisher to construct a leak protocol $\mathsf{Leak}_2 \in (2, n, \mu) - BCP$ and another distinguisher D_1 that violates the leakage-resilience of the scheme (**LShare**, **LRec**) for the secrets $l, l_\$$. The reduction is described below :

1. (**Initial setup**) : Fix $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(r)$ and $tp_1, \ldots, tp_n \leftarrow \mathbf{PShare}(m_\$)$.

2. (**Leak function** Leak_2) : We now design a n party leakage protocol Leak_2 for (**LShare**, **LRec**) using the given n party leakage protocol Leak for (**NMShare**, **NMRec**). To this end, it suffices to construct the corresponding Next_2 function. Let τ denote the transcript (initially empty). On input transcript τ, the function Next_2 invokes the underlying next function to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, g \leftarrow \mathsf{Next}(\tau)$. Then it uses the leakage function $\mathbf{g}$ to define the leakage function $\mathbf{g}_2(l_i)$ as follows : On input l_i,

output $\mathbf{g}(l_i, tr_i, tp_i)$. The $next_2$ function outputs i, g_1. Let τ denote the transcript, when the leakage protocol Leak finishes (formalized by Next outputting $\perp$). At this point, we continue and completely leak l_{i_1} and l_{i_2} in the next two rounds and then end our leakage protocol. As a result, the transcript of our leakage protocol Leak_2 will be $\tau \circ l_{i_1} \circ l_{i_2}$.

As τ is at most μ bits and and from Corollary 7 upto two shares of l can be fully leaked, the above leakage protocol belongs to the class $(2, n, \mu) - BCP$.

3. (**Distinguisher** D_1) : On input $\tau \circ l_{i_1} \circ l_{i_2}$, for each $i \in \{i_1, i_2\}$, let $tShare_i \leftarrow l_i \circ tr_i \circ tp_i$, tamper using $\mathbf{f}_i$ to obtain $\widetilde{tShare_i}$ which is parsed as $\widetilde{l_i} \circ \widetilde{tr_i} \circ \widetilde{tp_i}$. Using these values compute $\widetilde{r} \leftarrow \mathbf{RRec}(\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}})$ (this completes the initial setup of two hybrids in consideration). Compute $\widetilde{l} \leftarrow \mathbf{F}(l)$. Invoke the distinguisher with $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$ and output its output.

Notice, in the case the secret hidden by the leakage-resilient scheme $(\mathbf{LShare}, \mathbf{LRec})$ is $l_\$$, $\mathcal{D}$ will be invoked with input distributed according to $\mathbf{Hybrid_4}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_5}$. Therefore the success probability of D_1 will be equal to the advantage of $\mathcal{D}$ in distinguishing these two hybrids, which is greater than ϵ_3 by assumption. Hence, we have arrived at a contradiction to statistical leakage-resilience of the scheme $(\mathbf{LShare}, \mathbf{RRec})$. ■

Claim: $\mathbf{Hybrid_4}$ is identical to $\mathbf{Hybrid_5}$.
Proof: The two hybrids differ in how $\widetilde{l}$ is computed. In $\mathbf{Hybrid_4}$, the function $\mathbf{F}$ samples shares of l, such that $l_{i_1}, \ldots, l_{i_t}$ satisfy certain constraints. Notice that in $\mathbf{Hybrid_4}$, the fixed values of tl_{i_1} and tl_{i_2} already satisfy all these constraints. Consequently there is no need for sampling these shares and $\widetilde{l}$ can be directly computed using the fixed values. ■

The above two claims also show that $\mathbf{F}(l)$ does not `abort` with probability at least ϵ_3.
Claim: For any $m, m_\$ \in \mathbb{F}_0$, the statistical distance in between $\mathbf{Hybrid_5}$ and $\mathbf{Hybrid_6}$ is at most ϵ_7.
Proof: These two hybrids differ in the initial stage while creating share $tp_1, \ldots, tp_n$. Assume towards contradiction that there exists $m, m_\$ \in \mathbb{F}_0$, and a distinguisher $\mathcal{D}$ that is successful in distinguishing $\mathbf{Hybrid_5}$ and $\mathbf{Hybrid_6}$ with probability greater than ϵ_3. We use the reduction and such a distinguisher to construct a leak protocol $\mathsf{Leak}_2 \in (1, n, \mu) - BCP$ and another distinguisher D_1 that violates the statistical leakage-resilience of the scheme $(\mathbf{PShare}, \mathbf{PRec})$ for the secrets $m, m_\$$. The reduction is described below :

1. (**Initial setup**) : Fix $r_1, \ldots, r_n \leftarrow \mathbf{RShare}(r)$ and $l_1, \ldots, l_n \leftarrow \mathbf{LShare}(l)$.
2. (**Leak function** Leak_2) : We now design a n party leakage protocol Leak_2 for $(\mathbf{PShare}$, $\mathbf{PRec})$ using the given n party leakage protocol Leak for $(\mathbf{NMShare}, \mathbf{NMRec})$. To this end, it suffices to construct the corresponding Next_2 function. Let τ denote the transcript (initially empty). On input transcript τ, the function Next_2 invokes the underlying next function to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, g \leftarrow \mathsf{Next}(\tau)$. Then it uses the leakage function $\mathbf{g}(share_i)$ to define the leakage function $\mathbf{g}_2(p_i)$ as follows : On input p_i, output $\mathbf{g}(l_i, r_i, p_i)$. The Next_2 function outputs i, g_1. Let τ denote the transcript, when the leakage protocol Leak finishes (formalized by Next outputting $\perp$). At this point, we continue and each party $i \in T$ iteratively computes $\widetilde{l_i} \circ \widetilde{r_i} \circ \widetilde{p_i} \leftarrow \mathbf{f}_i(l_i \circ r_i \circ p_i, \tau)$ and outputs as leakage

$(\widetilde{l_i}$. Finally, party i_1 and i_2 iteratively output $\widetilde{r_{i_1}} \circ \widetilde{r_{i_2}}$ before terminating the leakage protocol. As a result, the transcript of our leakage protocol Leak_2 will be $\tau \circ \widetilde{l_{i_1}} \circ \ldots \circ \widetilde{l_{i_t}} \circ \widetilde{r_{i_1}} \circ \widetilde{r_{i_2}}$.

As two shares of r require $2\log|\mathbb{F}_3|$ bits and t shares of l can require at most $n\log|\mathbb{F}_2|$ bits (recall $t \leq n$), the above leakage protocol belongs to the class $(2, n, \mu) - BCP$.

3. (**Distinguisher** D_1) : On input $\tau \circ \widetilde{l_{i_1}} \circ \ldots \circ \widetilde{l_{i_t}} \circ \widetilde{r_{i_1}} \circ \widetilde{r_{i_2}}$, compute $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{l_{i_1}}, \ldots, \widetilde{l_{i_t}})$ and $\widetilde{r} \leftarrow \mathbf{RRec}(\widetilde{r_{i_1}}, \widetilde{r_{i_2}})$. Invoke the distinguisher $\mathcal{D}$ with $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$ and output its output.

Notice, in the case the secret hidden under the scheme $(\mathbf{PShare}, \mathbf{PRec})$ is $m_\$$, $\mathcal{D}$ will be invoked with input distributed according to $\mathbf{Hybrid_5}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_6}$. Therefore the success probability of D_1 will be equal to the advantage of $\mathcal{D}$ in distinguishing these two hybrids, which is greater than ϵ_7 by assumption. Hence, we have arrived at a contradiction to statistical leakage-resilience of the scheme $(\mathbf{PShare}, \mathbf{PRec})$. ■

By repeated application of triangle inequality to the above claims, we get that the statistical distance between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_6}$ is at most $\epsilon_3 + \epsilon_5 + \epsilon_7$. From our construction of $\mathbf{F}$ and $\mathbf{G}$, it is clear that for any l and r, if the reduction is successful in creating the t shares, then the secret hidden is these t shares is the same as the message encoded by l and r (under 2-out-of-2 scheme $(\mathbf{NMEnc}, \mathbf{NMDec})$). That is,

$$\mathbf{NMRec}(\{share_i : i \in T\}) = \mathbf{NMDec}(l, r)$$

Similarly, we can say that the secret hidden is the t tampered shares is the same as the message encoded by tampered $\widetilde{l}$ and tampered $\tilde{r}$. That is,

$$\mathbf{NMRec}(\{\mathbf{f_i}(share_i) : i \in T\}) = \mathbf{NMDec}(\mathbf{F}(l), \mathbf{G}(r))$$

Therefore, the tampering experiments of the two non-malleable secret-sharing schemes (see definition 11) are statistically indistinguishable, specifically,

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \approx_{\epsilon_0+\epsilon_3+\epsilon_7} \mathbf{Tamper}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}}$$

By the ϵ_1-non malleability of the scheme $(\mathbf{NMEnc}, \mathbf{NMDec})$, there exists a simulator $\mathbf{Sim}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}}$ such that

$$\mathbf{Tamper}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}} \approx_{\epsilon_1} \mathbf{Sim}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}}$$

We use the underlying simulator as our simulator and let

$$\mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \equiv \mathbf{Sim}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}}$$

Applying triangle inequality to the above relations we prove the statistical leakage-resilient non-malleability for this case $(|T| \geq 3)$.

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \approx_{\epsilon_0+\epsilon_1+\epsilon_3+\epsilon_7} \mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}}$$

As the the statistical distances between real and simulated experiments in the two cases are $(\epsilon_0 + \epsilon_3 + \epsilon_7)$ and $(\epsilon_0 + \epsilon_1 + \epsilon_3 + \epsilon_7)$, we take $(\epsilon_0 + \epsilon_1 + \epsilon_3 + \epsilon_7)$ as the worst case statistical error of our scheme $(\mathbf{NMShare}, \mathbf{NMRec})$. □

Acknowledgements

Ashutosh Kumar thanks Vipul Goyal, Venkatesan Guruswami and Jonathan Lee for useful discussions.

Ashutosh Kumar is supported by Dean's GSR Fellowship, NSF grant 1619348, NSF frontier award 1413955, US-Israel BSF grants 2012366, 2012378, and by the Defense Advanced Research Projects Agency (DAPRA) SAFEWARE program through the ARL under Contract W911NF-15-C-0205 and through a subcontract with Galois, Inc. Amit Sahai is supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, and NSF grant 1619348, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. Raghu Meka is supported by NSF grant CCF-1553605. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government

References

[ADKO15] Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In *Twelfth IACR Theory of Cryptography Conference (TCC 2015)*, 2015.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 774–783. ACM, 2014.

[ADN$^+$10] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 113–134. Springer, 2010.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Survey: Leakage resilience and the bounded retrieval model. In *International Conference on Information Theoretic Security*, pages 1–18. Springer, 2009.

[AYZ95] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.

[BBDW96] Simon R. Blackburn, Mike Burmester, Yvo Desmedt, and Peter R. Wild. Efficient multiplicative sharing schemes. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 107–118, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[BDIR18] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In *CRYPTO*, pages 531–561. Springer, 2018.

[Bei] Amos Beimel. *Secure schemes for secret sharing and key distribution, PhD Thesis.*

[Bei11] Amos Beimel. Secret-sharing schemes: a survey. In *International Conference on Coding and Cryptology*, pages 11–46. Springer Berlin Heidelberg, 2011.

[BEO$^+$13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 668–677. IEEE, 2013.

[BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 337–367. Springer, 2015.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *AFIPS National Computer Conference (NCC '79)*, pages 313–317, Los Alamitos, CA, USA, 1979. IEEE Computer Society.

[Bla99] SR Blackburn. Combinatorics and threshold cryptography. *Research Notes in Mathematics*, 403:44–70, 1999.

[BNS92] Laszlo Babai, Noam Nisant, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[BO15] Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 355–364. ACM, 2015.

[CDF$^+$08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT*, pages 471–488, 2008.

[CFL83] Ashok K Chandra, Merrick L Furst, and Richard J Lipton. Multi-party protocols. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 94–99. ACM, 1983.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, pages 155–168, 2014.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 285–298, 2016.

[Chu90] Fan RK Chung. Quasi-random classes of hypergraphs. *Random Structures & Algorithms*, 1(4):363–382, 1990.

[CL16] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors and almost optimal privacy amplification protocols. *FOCS*, 2016.

[CL18] Eshan Chattopadhyay and Xin Li. Non-malleable extractors and codes for composition of tampering, interleaved tampering and more. *ECCC 2018, Report 70*, 2018.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *International Conference on Security and Cryptography for Networks*, pages 121–137. Springer, 2010.

[Des98] Yvo Desmedt. Some recent research aspects of threshold cryptography. In Eiji Okamoto, George Davida, and Masahiro Mambo, editors, *Information Security*, pages 158–173, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[DLWZ14] Yevgeniy Dodis, Xin Li, Trevor D Wooley, and David Zuckerman. Privacy amplification and nonmalleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 227–237. IEEE, 2007.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 293–302. IEEE, 2008.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 434–452, 2010.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, pages 601–610, 2009.

[FKS84] Michael L Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0 (1) worst case access time. *Journal of the ACM (JACM)*, 31(3):538–544, 1984.

[GIM$^+$16] Vipul Goyal, Yuval Ishai, Hemanta K Maji, Amit Sahai, and Alexander A Sherstov. Bounded-communication leakage resilience via parity-resilient circuits. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 1–10. IEEE, 2016.

[GK18a] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 685–698. ACM, 2018.

[GK18b] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In *CRYPTO*, pages 501–530. Springer, 2018.

[GKP$^+$18] Vipul Goyal, Ashutosh Kumar, Sunoo Park, Silas Richelson, and Akshayaram Srinivasan. Non-malleable commitments from non-malleable extractors. Manuscript, 2018.

[GR15] Shafi Goldwasser and Guy N Rothblum. How to compute in the presence of leakage. *SIAM Journal on Computing*, 44(5):1480–1549, 2015.

[GR17] Venkatesan Guruswami and Ankit Singh Rawat. Mds code constructions with small sub-packetization and near-optimal repair bandwidth. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2109–2122. SIAM, 2017.

[GW16] Venkatesan Guruswami and Mary Wootters. Repairing reed-solomon codes. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 216–226. ACM, 2016.

[HG91] Johan Hastad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *Annual International Cryptology Conference*, pages 463–481. Springer, 2003.

[KGH83] Ehud Karnin, Jonathan Greene, and Martin Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, 1983.

[KN06] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 2006.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for np. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 254–273. Springer, 2014.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 564–575. IEEE, 2017.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Structure in Complexity Theory Conference, 1993., Proceedings of the Eighth Annual*, pages 102–111. IEEE, 1993.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1156. ACM, 2017.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *CRYPTO*, pages 517–532, 2012.

[MBW18] Jay Mardia, Burak Bartan, and Mary Wootters. Repairing multiple failures for scalar mds codes. *IEEE Transactions on Information Theory*, 2018.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography Conference*, pages 278–296. Springer, 2004.

[MW18] Cody Murray and Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for np and nqp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 890–901. ACM, 2018.

[NSS95] M Naor, LJ Schulman, and A Srinivasan. Splitters and near-optimal derandomization. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 182–191. IEEE, 1995.

[OPVV18] Rafail Ostrovsky, Giuseppe Persiano, Daniele Venturi, and Ivan Visconti. Continuously non-malleable codes in the split-state model from minimal assumptions. In *Annual International Cryptology Conference*, pages 608–639. Springer, 2018.

[PVZ12] Jeff M Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 486–501. SIAM, 2012.

[Raz00] Ran Raz. The bns-chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[RBO89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 73–85, New York, NY, USA, 1989. ACM.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[She14] Alexander A. Sherstov. Communication lower bounds using directional derivatives. *J. ACM*, 61(6):34:1–34:71, December 2014.

[TYB17] Itzhak Tamo, Min Ye, and Alexander Barg. Optimal repair of reed-solomon codes: Achieving the cut-set bound. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 216–227. IEEE, 2017.

[Wil14] Ryan Williams. Nonuniform acc circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, January 2014.

[Yao90] AC-C Yao. On acc and threshold circuits. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 619–627. IEEE, 1990.