Leakage-Resilient Secret Sharing Against Colluding Parties*

Ashutosh Kumar, Raghu Meka and Amit Sahai
Department of Computer Science
University of California, Los Angeles
Los Angeles, USA
Email: a@ashutoshk.com, raghum@cs.ucla.edu, sahai@cs.ucla.edu

***Abstract*—In this work, we consider the natural goal of designing secret sharing schemes that ensure security against an 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 a computationally unbounded adversary learns a bounded amount of *leakage*, where each bit of leakage *adaptively* and *jointly* depends on the shares of an *adaptively* chosen subset of p parties. Existing multi-party secret sharing schemes (Dziembowski and Pietrzak FOCS 07), (Goyal and Kumar STOC 18) and (Benhamouda, Degwekar, Ishai and Rabin CRYPTO 18) have focused on handling *non-adaptive* and *individual* leakage for (limited special cases of) *threshold* secret sharing schemes.**

- **We give an unconditional compiler that transforms any secret sharing scheme on n parties into a p-party leakage-resilient one for p upto $O(\log n)$. This yields the first multi-party secret sharing schemes that are secure against *adaptive* or *joint* leakage.**
- **As a natural extension, we initiate the study of *leakage-resilient non-malleable secret sharing*. We empower the adversary to adaptively leak from each of the shares and then use the leakage to tamper with all of them arbitrarily and independently. Leveraging our p-party leakage-resilient schemes, we compile any secret sharing scheme into a non-malleable one ensuring that 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 18) 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 many applications in cryptography.**
- **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 exponential dependence, even for *simultaneous, non-adaptive leakage*, will lead to progress on longstanding open problems in complexity theory.**

I. Introduction

Blakley [2] and Shamir [3] 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 structures (see Section II for formal definitions).

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 [4], robustness [5], functionality [6] or non-malleability [7]. In this work, our focus is on a stronger secrecy goal—*leakage-resilience*.

Leakage-resilience has a long history in cryptography. Intuitively, the goal of leakage-resilience is to make cryptosystems robust to adversaries who learn additional "leaked" information (via say passive side-channel attacks). 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 [8] initiated the study of private circuits. Micali and Reyzin [9] put forward a very general model for such side-channel attacks. Subsequently, a lot of primitives in cryptography were made leakage-resilient [7], [10]–[19]. More detailed history can be found in the recent survey of Kalai and Reyzin [20].

Leakage-Resilient Secret Sharing.: Focusing now on leakage resilience in the context of secret sharing, one of the first works to study a question in this vein was Dziembowski and Pietrzak [10] who developed n-out-of-n *intrusion-resilient* secret sharing schemes that ensure that the adversary does not learn anything about the secret even after getting access to bounded information from each of the shares and supported limited adaptivity [1]; in addition, they also observed that their results implied certain round-complexity separations in communication complexity. Davì, Dziembowski and Venturi [14] used two-source extractors to construct the first 2-out-of-2 secret sharing scheme that statistically hides the secret even after an *adaptive* adversary executes an arbitrary leakage protocol on the two shares with bounded communication.

*A prior version appeared as "Leakage-Resilient Secret Sharing" [1] containing the same set of results.

[1]The leakage could be in k rounds but had to avoid certain patterns as dictated by their reconstruction function which required $k+1$ rounds of alternating extraction.

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00045

A natural first question is whether existing secret sharing schemes such as Shamir's secret sharing scheme could itself be leakage-resilient. Surprisingly, it follows from the work of Guruswami and Wootters [21] (motivated by coding theoretic questions) that one can learn the secret under Shamir's scheme for most thresholds even with one-bit leakage from each share *individually* (i.e., the leaked bit for each party only depends on that party's share).

Recently, Goyal and Kumar [7], [22] defined and constructed an efficient 2-out-of-n secret sharing scheme that hides the secret even when a non-adaptive adversary learns some bounded amount of information from each of the n shares individually. Concurrently, Benhamouda, Degwekar, Ishai and Rabin [19] 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 independently learns some bounded amount of information from each share individually.

Our Work: Adaptive and Joint Leakage.: Therefore the focus of recent literature on leakage-resilient secret sharing is on handling *individual* and *non-adaptive* leakage for (limited special cases of) *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 where an *adaptive* adversary can learn information depending on multiple shares at once. For example, in our model, we would allow the adversary to adaptively specify an arbitrary leakage function f outputting a bounded number of bits, 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 secret sharing schemes that support *general access structures*.

As an important special case consider, for any *constants* $t \leq n$, constructing a t-out-of-n secret sharing scheme, which should remain secure even when a non-adaptive adversary can obtain joint leakages $\{f_{i_1,i_2,\ldots,i_{t-1}}(share_{i_1}, \ldots, share_{i_{t-1}})\}_{i_1,\ldots,i_{t-1}\in[n]}$. Before our work, this problem was open even for $t = n = 3$. We resolve this problem for all constants t, and more! We now elaborate.

Modeling adaptive joint leakage.: In particular, we model this by viewing leakage as (an adversary) running a communication protocol, where in each round, any adaptively chosen 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 [23], [24] from communication complexity which in its original form studies the following problem. There are n parties with each party seeing all inputs but their own (*number written on forehead*) and they can communicate by writing on a public blackboard. Their goal is to compute a function of their inputs while minimizing the total amount of communication. This is a very versatile model with many beautiful connections to circuit complexity among others (cf. book of [25]).

Note that p-party collusion protocols for $p = n - 1$ correspond exactly to NOF protocols. In a similar vein 1-party collusion protocols correspond to the well-studied *number-in-hand* (NIH) model [26]–[28] (a more straightforward extension of two-party communication to multi-party communication).

Bounded collusion protocols also seem particularly well suited to secret sharing as 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-resilient secret sharing schemes that we study (see Section II for a more formal definition):

Definition 1. *(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. 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 $\mathsf{Leak}(\mathbf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathbf{Share}(b))$.*[2]

We remark that even 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.

A. Leakage-Resilient Non-Malleable Secret Sharing

Just as leakage resilience was introduced to combat passive side-channel attacks, an important development in cryptography has been *non-malleability* introduced in the seminal work of Dolev, Dwork and Naor [29]. Intuitively, the goal here is to protect cryptosystems from adversaries who may

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

even tamper with the data (e.g., physical tampering). These were recently introduced in the context of non-malleable codes and extractors [16], [30]–[38], but have since found a vast array of applications (e.g., [39]) and many cryptographic primitives — including secret sharing as in the recent work of Goyal and Kumar [7] — have been made non-malleable (in various precise senses)).

Notice, however, that a leakage attack might be easier to perform than a tampering one in practice. This raises the challenge of making cryptosystems both leakage resilient and non-malleable[3] and we address this question for secret sharing schemes.

Following the definitions of [16], [17] (who study leakage-resilient and non-malleable 2-out-of-2 secret sharing schemes) and [7], [22] (who study non-malleability for more general access structures), we empower the adversary to adaptively leak some bounded amount of information from all the shares and in addition use this leakage to arbitrarily tamper with each of the shares. We define a secret sharing scheme to be leakage-resilient non-malleable (LR NMSS) if the secret reconstructed from the tampered shares is either the original one or a completely "unrelated" one.

The well-studied non-malleable codes in two split-state model [16], [33]–[38] correspond to the special case of 2-out-of-2 NMSSs (see [17] for a proof). [22] 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. [7] construct t-out-of-n schemes against a stronger adversary who chooses any t shares, partitions them into two subsets of different cardinality and jointly tampers all the shares within each subset independently. However, these constructions do not consider a leaking adversary, and cannot be made to handle leakage from more than t shares.

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 [16] defined and constructed such codes against computationally bounded adversaries. Aggarwal, Dziembowski, Kazana, and Obremski [17] obtained the first information-theoretic construction of 2-out-of-2 LR NMSS. Non-malleable extractors based 2-out-of-2 LR NMSS were given by Chattopadhyay and Li [40] and [7]. Goyal et al. [41] and Ostrovsky et al. [42] have used such leakage-resilient codes to obtain constructions of 'concurrent' non-malleable commitments and 'continuous' non-malleable codes respectively.

B. Previous 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. As mentioned already, classical secret sharing schemes such as Shamir's secret sharing scheme are not leakage-resilient (even against simultaneous individual leakage). While [19] overcome this using fields of large characteristic for $t = n - o(\log n)$, it is not clear how to extend their results to handle joint or adaptive leakage.

Can we use extractors to get 2-party leakage-resilient schemes?: One of the main challenges we address is in handling *joint* leakage. Indeed, most existing $(1, 2, 2)$-LRSSs are based on two source extractors [7], [10], [14], [17]. These constructions rely on the following simple but powerful observation: if two shares are independent, then conditioning on the entire transcript of a 1-party collusion protocol preserves the conditional independence between them, and therefore independent source extractors can be invoked for proving leakage-resilience. Unfortunately this idea has severe problems for 2-party collusion protocols as among other things, conditioning on a 2-party collusion protocol will break the independence of the shares.

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 [7] 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 scheme for $t = n = 3$.

Can we extend the LRSS of Goyal and Kumar?: [7] constructed 2-out-of-n schemes that are resilient against non-adaptive individual leakage. With a little work, their methods can be extended to yield efficient c-out-of-n schemes that are similarly resilient against non-adaptive individual leakage for any constant c. However, apart from not being able to handle super-constant thresholds, they are based on extractors and thus cannot be shown to be 2-party leakage-resilient.

C. Our Results

Leakage-resilient secret sharing.: As our main result, we give a generic compiler that transforms any secret sharing scheme into a p-party leakage-resilient one.

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*

[3]Note that neither one implies the other and there are systems that satisfy one but not the other.

leakage resilient against (p, n, μ)-BCP. The resulting scheme shares secrets of k bits into n shares each of length $\ell + k(\log n)(\mu + \log(1/\epsilon))2^{O(p)}$. [4]

While we do not focus on *rate* (the ratio of the length of old shares to that of new shares) in this work, our rate is essentially $\Omega(1/\log n)$ for any constant p.

We remark that efficient[5] constructions of LRSSs with asymptotically better dependence on the collusion bound p will lead to breakthroughs in communication complexity ([24], [43]–[45]). In fact, even constructing efficient resilient schemes where $p = \omega(\log n)$, where the leakage is simultaneous and non-adaptive[6] would lead to breakthroughs in circuit complexity.

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

In particular, if a $(n-1, n, n)$-LRSS is efficient, then we obtain an unconditional separation between $\mathbf{ACC^0}$ and $\mathbf{P}$. This would be a major breakthrough as existing breakthrough results [46] (resp. [47]) only separate $\mathbf{ACC^0}$ and $\mathbf{NEXP}$ (resp. $\mathbf{NQP}$). We give more precise statements of such implications in Section VI.

We next mention some interesting corollaries of our main result. Using Shamir's t-out-of-n secret sharing scheme [3], 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 individual leakage ($p = 1$):

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

Instantiating with the secret sharing scheme of Karchmer and Wigderson [48], we get:

Corollary 3 (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.*

[4]Our techniques can be easily extended to ensure leakage-resilience even after the adversary completely learns any unauthorized set of shares along with the transcript of the p-party protocol. Please see the full version.

[5]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.

[6]That is, the leakage happens in a single round where the adversary learns some leakage from each coalition of size at most p at the same time.

Using the computational scheme of Yao (mentioned in [49]), we get:

Corollary 4 (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 [50], we arrive at the following:

Corollary 5 (Informal)**.** *If one-way functions and witness-encryption for $\mathbf{NP}$ exist, then for every $\mathbf{monotone\ NP}$ 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.*

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 [7], [16], [17], [40], [41] and t-out-of-n NMSS that can be extended to handle leakage and tampering from at most t shares [7].

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 6 (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 [48]–[50], we can also get further corollaries similar to Corollaries 3, 4, 5.

D. Discussion of Model and Further Questions

The leakage model we consider is motivated by protocols in communication complexity and is admittedly quite strong

in that we allow joint as well as adaptive leakage. While this limits our quantitative bounds (we need $p = O(\log n)$ for efficiency), working in this generality gives us several advantages:

- Use powerful lower bounds in communication complexity to get leakage resilience. Further, as mentioned earlier in the introduction, constructing efficient schemes even for non-adaptive but joint leakage for $p = \omega(\log n)$ hits longstanding bottlenecks.
- For $t = O(\log n)$, in particular, for any constant t, our scheme is *fully leakage-resilient* in the sense that $p = t - 1$ would be the limit as per the secrecy guarantee.
- Being able to handle overlapping coalitions could be useful even when interested in weaker models of leakage. For example, consider a natural variant where we want to construct t-out-of-n LRSS where the adversary statically partitions the n parties into *disjoint* groups of size at most p each and non-adaptively leaks from each subset. Constructing efficient schemes for such leakage directly seems challenging even for $p = \omega(1)$. The point is that even if we only want resilience against such protocols, being secure against arbitrary coalitions *allows us* to use the ideas of [51], [52] and scatter the shares of parallel instantiations among different parties using perfect hash families. In a similar vein, while we design LR NMSS schemes secure against individual leakage and tampering, the constructions and analysis rely on resilience against adaptive joint leakage.
- Along with the results we prove, our work raises several seemingly new natural and independently interesting questions (discussed below) in communication complexity and pseudo-randomness which could be viewed as further conceptual contributions of this work.

1) Open Problems:

Beating $\log n$ *for restricted models.:* A natural question is whether the limitation of $p = O(\log n)$ can be overcome for some restricted models of joint leakage. A very natural and interesting model to study would be the case where the adversary statically partitions the n parties into *disjoint* groups of size at most p each and non-adaptively leaks from each subset. There are no bottlenecks from circuit complexity for $p = \omega(\log n)$ in this setting and in fact we have strong communication lower bounds against such protocols. However, even when $p = \omega(1)$, we don't know a way to get efficient schemes in this setup without using NOF lower bounds (owing to the use of *scattering* based on perfect hash families in our constructions).

Lower bounds for bounded-collusion protocols.: Bounded collusion protocols interpolate nicely between the NIH model, where we have strong communication lower bounds, and the NOF model where proving lower bounds for super-logarithmic (in the input length) number of parties is a fundamental challenge in communication complexity. Allowing a large number of parties n compared to the collusion bound p, could make proving lower bounds against BCPs easier than against NOF protocols; besides being interesting by itself this could shed further light on the NOF model as well. One concrete question is the following: Find an explicit function $f : (\{0,1\}^m)^n \to \{0,1\}$ that requires $m^{\Omega(1)}$ communication under p-party collusion protocols for $p = \omega(\log n)$ and $m = n^{O(1)}$. While this does not hit any bottlenecks in circuit complexity, the problem still seems challenging even for one-round simultaneous protocols.

Answering the above would be a (necessary) step toward potentially getting (p, t, n)-LRSS for $p = \omega(\log n)$, $t \gg p$ (e.g., $t = n^{\Omega(1)}$).

Extractors for cylinder intersections.: Trying to use extant techniques of deriving leakage-resilience from extractors [7], [10], [14], [15], [17] 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 [24], [25].

Definition 2. ***(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* 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 $\text{Ext} : (\{0,1\}^m)^n \to \{0,1\}$ *an extractor*[7] *for cylinder intersections as above with error ϵ if*

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

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 communication lower bounds against p-party collusion protocols. Indeed, the results of [24] do imply explicit extractors for cylinder-intersection sources when the min-entropy $k \geq (1 - c_p)m$ 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.[8]

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

[8] Indeed, constructing independent-source extractors was easier when the number of sources n is large and this could be the case here too [53], [54].

E. Overview of Constructions

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 [23]. Indeed, the connection to NOF model allows us to leverage fundamental results of Babai, Nisan, and Szegedy [24] (also Chung [43], Raz [44], Sherstov [45]) on constructing explicit hard functions against NOF protocols to get the first (and simple) secret sharing schemes that are secure against adaptive and 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 $\mathsf{GIP}(x_1,\ldots,x_n) = \sum_{i=1}^{m}\prod_{j=1}^{n} x_{ij} \mod 2$ be the *generalized inner-product* function. Babai, Nisan, and Szegedy [24] showed that the randomized communication complexity of GIP in the NOF model is $\Omega(m/4^n)$. This was further tightened by Chung [43] to the optimal bound $\Omega(m/2^n)$. 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 $\mathsf{GIP}(sh_1,\ldots,sh_{n-1},x) = s$.

Analysis. It is not hard to see that any subset of $(n-1)$-shares statistically hides the underlying secret. Further, the lower bound of [24], [43] 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 III 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 [24] can indeed be extended to show that GIP requires $\Omega(m/2^p)$ communication to compute under p-party collusion protocols.

However, there is another additional challenge: the correlation lower bounds of [24] (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 [45] 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 [51], [52], [55]. 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 [51], [52]. 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 h is injective on I. These families were introduced in the seminal work of Fredman, Komlós, and Szemerédi [56]. Alon, Yuster and Zwick [57] and Naor, Schulman and Srinivasan [58] 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 $H = \{h_1,\ldots,h_N\}$.

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 $(sh_1^i, \ldots, sh_{p+1}^i)$ for $i \in [N]$. For each $j \in [n]$ set the share of the j'th party to be

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

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.

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 an $i \in [N]$ 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 $(sh_1^i, sh_2^i, \ldots, sh_{p+1}^i)$ 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 $l_1, \ldots, l_n$ and $r_1, \ldots, r_n$ 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.

2) Leakage-Resilient Non-Malleable Secret Sharing Schemes.: Obtaining LR NMSS 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 such as additive combinatorics based analysis of inner-product function [34], flip-flop alternating extractor [36], a correlation breaker with advice generator [38], [40], [59]. Such techniques have been also employed towards 2-out-of-2 LR NMSS resulting in technical constructions [7], [17], [40], [41].

Our starting point is the compiler of Goyal and Kumar [22] 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 [7] for $t \geq 3$. To share a secret m, let $\ell, r \leftarrow$ 2-out-of-2-NMSS(m), $(\ell_1, \ldots, \ell_n) \leftarrow$ t-out-of-n-ShamirShare(ℓ) and $(r_1, \ldots, r_n) \leftarrow$ 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 [21]) disallowing us from relying on 2-out-of-2-NMSS. To fix this, we may be tempted to rely on 2-out-of-2 LR NMSS (or leakage-resilient non-malleable codes [7], [17], [40], [41]). 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 will need to sample all the n shares using ℓ and r independently 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.: We use our leakage-resilient secret sharing schemes to share both ℓ and r. Our idea is to rely on leakage-resilience to obtain a 'fake' leakage-transcript by executing the leakage-protocol on n 'fake' shares encoding arbitrary ℓ and r in our reduction and upon availability of real values of ℓ and r, adjust the 'fake' shares and use the 'fake' leakage-transcript to give explicit functions that independently tamper with ℓ and r violating the non-malleability of 2-out-of-2 NMSS. We need to be careful as there is a subtle issue: leakage-resilience of neither the scheme sharing ℓ or r may be violated and yet non-malleability may be completely lost; the leakage transcript may have information about the secret m and still be independent of each of ℓ and r (by the secrecy property of 2-out-of-2 NMSS). As we have to independently tamper with ℓ and r in our reduction, a straightforward hybrid argument cannot be applied.

Using joint leakage and adaptivity.: We partially fix the issue by strengthening the scheme sharing ℓ to be secure against adaptive and joint-leakage. While we continue to rely on the idea of [7] of treating the tampered shares of ℓ as leakage from shares of r, in this work, we also consider leakage in the other direction and 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 [22], we execute two schemes in 'parallel': one designed for authorized pairs and other catering to larger authorized sets. [22] avoided correlations in between the two schemes by ensuring that every minimal authorized set of one scheme does not have any authorized set of the other scheme. Unfortunately, we face new difficulties during the composition as leakage may correlate both the schemes.

Our key idea: 'leakage-leveraging'.: We use our idea of *leakage-leveraging* to fix many of these issues. Specifically, we think of leakage transcript as leakage from the shares of ℓ. Next, we think of the leakage transcript and tampered shares of ℓ as *adaptive* leakage from the shares of r. Finally, we think of the leakage transcript and tampered shares of both ℓ and r as *adaptive* leakage from the 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'.

Open problem: Handle joint-leakage for NMSS.: 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.

F. Concurrent and Independent Work

The following independent and concurrent works also addressed the question of leakage resilience in secret sharing schemes:

- Aggarwal et al. [60] give a compiler that transforms any statistical secret sharing scheme into one that is leakage-resilient against non-adaptive individual leakage. They also construct a NMSS scheme that is secure against a concurrent adversary who may independently and non-adaptively tamper with each of the shares multiple times. They show an application to threshold signature schemes. Our schemes do not handle multiple-tampering as they can.
- Badrinarayanan and Srinivasan [61] focus on obtaining an efficient NMSS scheme that has positive rate. They give a compiler that converts any 4-monotone [9] statistical secret sharing scheme into one that is non-malleable against independent tampering of shares. Towards this, they construct $O(1)$-out-of-n scheme that is leakage-resilient against non-adaptive individual leakage. They also handle multiple-tamperings (see [60] for detailed comparison).
- Srinivasan and Vasudevan [62] focus on the rate of LRSS and construct a rate-preserving compiler that transforms any statistical secret sharing scheme into one that is leakage-resilient against non-adaptive individual leakage. This improves the rate of NMSS scheme of [61] as well. For a result towards leakage-resilient multi-party computation, they give a rate $\Omega(1/n)$ t-out-of-n LRSS against an adversary who learns any set of $t-2$ shares and then uses these fixed $t-2$ shares to *independently* learn non-adaptive information from each of the other $n-t+2$ shares.

Our focus in this work is largely orthogonal to these works: we focus on being resilient to *adaptive and joint* leakage. The LRSS schemes in these works do not allow an adversary to adaptively leak from each of the shares ($p=1$ in our notation) or to perform joint leakage (even for $p=2$).

II. 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 3. ***(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 :*

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

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 $\mathbf{D_1} \approx_\epsilon \mathbf{D_2}$. We say $\mathbf{D_1} \not\approx_\epsilon \mathbf{D_2}$ when $|\mathbf{D_1}-\mathbf{D_2}| > \epsilon$.

A. Secret Sharing Schemes

The following definition is inspired from the survey [49].

Definition 4. ***(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*

[9] each authorized set has size at least 4

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 i. 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_i to the party i. 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 5. ***(Secret sharing scheme** [49]). 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 privacy***.

Taking inspiration from the secret sharing literature, we have chosen not to include efficiency requirement (poly time sharing and reconstruction function) as a part of definition of secret sharing. Removing the efficiency requirement allows our results to generalize to access structures for which efficient secret sharing schemes are not yet known.

B. Threshold Access Structure $\mathcal{A}^t_n$

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 $\mathcal{A}^t_n = \{B \subseteq [n] : |B| \geq t\}$. We use the notation of (t, n, ϵ)-secret sharing scheme for denoting (n, ϵ)-secret sharing scheme realizing access structure $\mathcal{A}^t_n$.

C. Leakage-Resilient Secret Sharing Schemes

Goyal and Kumar [22] 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 6. ***(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 $\mathbf{Share}(b)$.

D. 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. Note that $\mathbf{f}$ may possibly depend on τ. 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.

We remark that the n parties are assumed not to store any additional information apart from prior leakage about the shares of other parties (possibly used for prior leakage). This assumption is necessary, as otherwise, a single party jointly leaking with every other party, can completely learn the secret by maintaining a private state that stores the shares of all the other parties.

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

Definition 7 (**(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 parameters 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.

III. 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 [23]. While the above is a little repetitive, we include the usual definition of NOF communication for clarity (see [25] for more details and references).

Definition 8. ***(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,*

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

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 and $\Pi(\mathbf{f}^{-1}(b))$ denotes the distribution of the transcript written on the black-board when the inputs to the n parties is a uniformly chosen pre-image of b under the predicate $\mathbf{f}$.

Note that the predicate $\mathbf{f}$ need not be efficiently invertible to satisfy the above definition.

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 $\mathbf{CC}^{NOF}_{\mathbf{n}}(f) \geq \mu$, 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 [24] gives us the following:

Corollary 7. *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: [24] showed that the generalized-inner-product function $GIP : (\{0,1\}^b)^n \to \{0,1\}$ defined as $GIP(x_1, \ldots, x_n) = \oplus_{i=1}^{b} \prod_{j=1}^{n} x_{ij}$ satisfies $\mathbf{CC}^{\text{NOF}}_{\mathbf{n}}(GIP) \geq cb/2^n$ 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 1. ∎

Note that the share length in the above construction is exponential in the number of parties. However, as we observe in Section VI, 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.

A. 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. *([63]) 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 $sh_1 \leftarrow \mathbf{XORRec_2}(m, sh_2)$.*

B. $(n-1, n, n)$-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 bit 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 $s \leftarrow \{0,1\}$. 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. ∎

IV. $(p, p+1, n)$-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 $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{p+1}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{p+1}})$ 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, $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{n}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{n}})$, 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 7 immediately gives the following:

Corollary 8. *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, $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{n}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{n}})$, shares secrets of a bits into n shares each of length $a(\mu + \log(1/\epsilon))2^{O(p)}$.*

A. 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. [51], [52]).

Definition 9. ***[Perfect hash families** [56]] 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, 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 $\mathrm{poly}(n, d)$.

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 $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{p+1}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{p+1}})$ 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, $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{n}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{n}})$, 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 $(\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{n}}, \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{n}})$ given below :

- (**Sharing function $\mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{n}}$**).
  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 $m^j_1, \ldots, m^j_{p+1} \leftarrow \mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{p+1}}(m)$. Using functions from the above perfect hash family, for each $i \in [n]$, construct $share_i$ as $\left(m^1_{f_1(i)}, \ldots, m^d_{f_d(i)}\right)$.

- **Reconstruction function** ($\mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{n}}$).
  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 $\left(m^1_{f_1(i)}, \ldots, m^d_{f_d(i)}\right)$. Find $j \in [d]$ such that f_j is injective on T. Use the reconstruction procedure of underlying leakage resilient scheme to compute $m \leftarrow \mathbf{LRRec}^{\mathbf{p+1}}_{\mathbf{p+1}}(m^j_1, \ldots, m^j_{p+1})$. 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 definition 9). 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 $i \in [j-1]$, share m_1 using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_{p+1} \leftarrow \mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{p+1}}(m_1)$. For each $i \in [d] \setminus [j]$, share m_2 using the sharing procedure of the underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_{p+1} \leftarrow \mathbf{LRShare}^{\mathbf{p+1}}_{\mathbf{p+1}}(m_2)$. 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 $S \subset [n]$ 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 the underlying set $T \leftarrow \{f_j(i) : i \in S\}$ corresponding to f_j. Next, we construct leakage function $\mathbf{g_1}$ that takes $\otimes_{i\in T} m_i$ as input, for each $i \in S$, sets $m_i^j \leftarrow m_{f_j(i)}$. 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 $T, \mathbf{g_1}$.

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.

■

V. 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 $\epsilon_1 \geq 0$, 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 $(\mathbf{LRShare}_{\mathbf{n}}^{\mathbf{p+1}}, \mathbf{LRRec}_{\mathbf{n}}^{\mathbf{p+1}})$ 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 bit-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 (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 $r_1, \ldots, r_n \leftarrow \mathbf{LRShare}_{\mathbf{n}}^{\mathbf{p+1}}(r)$. 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 $r \leftarrow \mathbf{LRRec}_{\mathbf{n}}^{\mathbf{p+1}}(\otimes_{i\in T} r_i)$. 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 $\mathbf{g}_1$ that takes $\otimes_{i\in S} r_i$ as input, and outputs $\mathbf{g}\big(\otimes_{i\in S} (l_i \circ r_i)\big)$. Output $S, \mathbf{g}_1$.

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.

■

A. 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** $\mathbf{MBShare}$) :
  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 $m_1^j, \ldots, m_n^j \leftarrow \mathbf{SBShare}(m^j)$. For each $i \in [n]$, construct $share_i$ as $(m_i^1 \circ \ldots \circ m_i^b)$.

- (**Reconstruction function** $\mathbf{MBRec}$) :
  On input the shares of an authorized set T, for each $i \in T$, parse $share_i$ as $(m_i^1 \circ \ldots \circ m_i^b)$. For each $j \in [b]$, use the reconstruction procedure of underlying leakage resilient scheme to compute $m^j \leftarrow \mathbf{SBRec}(\otimes_{i \in T} m_i^j)$. 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 $m_1^j, \ldots, m_n^j \leftarrow \mathbf{SBShare}(c^j)$. Similarly, for each $j \in \{k+1, \ldots, b\}$, share d^j to obtain $m_1^j, \ldots, m_n^j \leftarrow \mathbf{SBShare}(d^j)$. 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 $i \in S$, computes $share_i$ as $(m_i^1, \ldots, m_i^b)$ and outputs $\mathbf{g}(\otimes_{i \in R} share_i)$. Output R, $\mathbf{g}_1$.

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^k and d^k with advantage greater than ϵ. This violates the leakage-resilience of the single bit scheme, and therefore completes the proof. ∎

B. Instantiations

Corollary 9. *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. [24]. 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. [58]. 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 10. *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 [3] in Corollary 9. ∎

It is straightforward to use secret sharing schemes of [48]–[50] to obtain corresponding corollaries mentioned in the introduction, and consequently we omit these details.

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

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 VII.

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. ∎

VI. 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 (realizing any access structure) 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, for the special case of n-out-of-n, the above observation has the following corollary:

Corollary 11. *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. Formally, $\mathbf{CC}^{NOF}_{\mathbf{n}}(\mathbf{Rec}) \geq \mu$.*

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 $\Omega(k/2^n)$ [24], [43]–[45]. Even handling *non-adaptive* adversaries is a challenge. This can be seen from the classical results of Yao [64] and Hastad and Goldmann [65] 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 12. *For single bit secrets, for any number of parties $n \geq 2$, suppose there is $(p, p+1, n)$-LRSS leakage-resilient w.r.t a computationally unbounded adversary who learns p^2 bits of leakage such that each bit of the leakage non-adaptively depends on at most p shares. If the size of each of the shares of this scheme is $2^{p^{o(1)}}$, then this implies that the reconstruction procedure of this scheme does not belong to $\mathbf{ACC^0}$.*

Proof: Follows by combining the argument of Hastad and Goldmann [65] and Yao [64]. ∎

In particular, if a $(n-1, n, n, \mu)$-LRSS is efficient (efficiency implies that the shares are of size $\mathrm{poly}(n, \mu)$ bits), then we obtain an unconditional separation between $\mathbf{P}$ and $\mathbf{ACC^0}$. Contrast this with the celebrated result of Williams [46] which unconditionally separates $\mathbf{NEXP}$ from $\mathbf{ACC^0}$ (using different techniques) and the recent result of Murray and Williams [47] separating non-deterministic quasi-poly $\mathbf{NQP}$ from $\mathbf{ACC^0}$.

The above discussion suggests that improving Corollary 10 to obtain efficient (p, t, n)-LRSS for $p = \omega(\log n)$ could be considerably harder.

VII. 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 begin by recalling the definition of non-malleable secret sharing from [7], [22].

Definition 10. *(**Non-Malleable Secret Sharing Schemes** [7], [22]) 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*

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

*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 $\mathbf{SD}^{\mathbf{f},\mathbf{T}}$ (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 $\mathbf{STamper}^{\mathbf{f},\mathbf{T}}_{\mathbf{m}}$ and*

$$\mathbf{SSim}^{\mathbf{f},\mathbf{T}}_{\mathbf{m}} = \left\{ \begin{array}{c} \widetilde{m} \leftarrow \mathbf{SD}^{\mathbf{f},\mathbf{T}} \\ Output : m \text{ if } \widetilde{m} = same^*, \text{or } \widetilde{m}, otherwise \end{array} \right\}$$

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

A. Tampering Family

We first recall the split-state (individual) tampering family from [7] and generalize this family to encompass leakage.

Individual Tampering Family $\mathcal{F}^{split}_n$.: Let $\mathbf{Share}$ 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 $\mathbf{f_i} : \mathcal{S} \rightarrow \mathcal{S}$ 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 $\mathcal{F}^{split}_n$ denote the family containing all such tampering functions, namely $(\mathbf{f_1}, \ldots, \mathbf{f_n})$.

Individual Leakage Tampering Family $\mathcal{F}^{ind-leak}_{n,\mu}$.: Let the n shares be $share_1, \ldots, share_n$ be as in the definition $\mathcal{F}^{split}_n$. 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 $\mathbf{f_i} : \mathcal{S} \times \{0,1\}^\tau \rightarrow \mathcal{S}$ be an arbitrary tampering function, that takes as input $share_i$ and τ (leakage transcript) to output $\widetilde{share_i}$. Let $\mathcal{F}^{ind-leak}_{n,\mu}$ 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 definitions from [22].

Definition 11. *(**Minimal basis access structure** [22]) 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 12. *(**Paired access structures** [22]) 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 13. *(**Authorized paired access structures** [22]) 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 $\mathcal{A}^{min}_{pairs}$ will be equal to the set of all the authorized sets of size two in $\mathcal{A}$.

Efficient membership queries.: To achieve the general result, we need 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 construction.

Theorem 4. *For any number of parties n, and any access structure $\mathcal{A}$ that does not contain singletons, 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 [10] that is ϵ_1-non-malleable w.r.t. $\mathcal{F}^{split}_2$, which encodes an element of the set $\mathbb{F}_0$ into two elements of $\mathbb{F}_1$.*
2) *Let μ be any leakage bound. For any $\epsilon_2 \geq 0$, $\epsilon_3 > 0$, let $(\mathbf{LShare}, \mathbf{LRec})$ be any (n, ϵ_2)-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ (with authorized pairs pruned [11]) 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*

[10] Alternatively, we could have used a coding scheme that is ϵ_1-non-malleable w.r.t $\mathcal{F}^{split}_2$ (see [31]). Aggarwal et al. [17] show that such a code is a $(2, 2, 2\epsilon_1)$-secret sharing scheme that is ϵ_1-non-malleable w.r.t. $\mathcal{F}^{split}_2$. Using SS allows us to encompass 2-out-of-2 NMSS that may be designed in future supporting perfect secrecy ($\epsilon_0 = 0$).

[11] Our main compiler, given in theorem 3, already prunes all authorized pairs of any access structure for $p = 2$.

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. $\mathcal{F}_{n,\mu_2}^{ind-leak}$, which shares an element of the set $\mathbb{F}_0$ into n elements of $\mathbb{F}_4$.*

then there exists $(n, \epsilon_0 + \epsilon_2 + \epsilon_6)$-secret sharing scheme (resp. computational) realizing access structure $\mathcal{A}$ that is $(\epsilon_0 + \epsilon_1 + \epsilon_3 + \epsilon_5 + \epsilon_7)$-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 [22] $\mathbf{FindMinSet} : \mathcal{A} \rightarrow \mathcal{A}^{min}$, which takes an authorized set and outputs a minimal authorized set contained in that set. Our construction follows:

- **Sharing function** $\mathbf{NMShare}$: Encode the secret $m \in \mathbb{F}_1$ using $\mathbf{NMEnc}$ to obtain $l, r \leftarrow \mathbf{NMEnc}(m)$. Share l using a $\mathbf{LShare}$ to obtain $l_1, \ldots, l_n \leftarrow \mathbf{LShare}(l)$. Share r using $\mathbf{RShare}$ to obtain $r_1, \ldots, r_n \leftarrow \mathbf{RShare}(r)$. Share m using $\mathbf{PShare}$ 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** $\mathbf{NMRec}$: 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 $\mathbf{FindMinSet}$ 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 $\mathbf{PRec}$ to obtain the hidden secret $m \leftarrow \mathbf{PRec}(p_{i_1}, p_{i_2})$. Otherwise, run the reconstruction procedure $\mathbf{LRec}$ on t shares of l, to obtain $l \leftarrow \mathbf{LRec}(\otimes_{i \in T} l_i)$. Run the reconstruction procedure $\mathbf{RRec}$ on the first 2 shares of r, to obtain $r \leftarrow \mathbf{RRec}(r_{i_1}, r_{i_2})$. Decode l and r using $\mathbf{NMDec}$ 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 [22].

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 $\mathbf{FindMinSet}$ 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 $share_i$. We can also assume without loss of generality that all these 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 adversary specified leakage and tampering functions to create explicit functions violating the non-malleability of the underlying non-malleable secret-sharing schemes. Like [22], 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 an arbitrary $m_\$$ and let $(l_\$, r_\$) \leftarrow \mathbf{NMEnc}(m_\$)$. Run the sharing function $\mathbf{LShare}$ with input $l_\$$ to obtain $(tl_1, \ldots, tl_n) \leftarrow \mathbf{LShare}(l_\$)$. Run the sharing function $\mathbf{RShare}$ with input $r_\$$ to obtain $(tr_1, \ldots, tr_n) \leftarrow \mathbf{RShare}(r_\$)$. Fix all these sampled shares.
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 Next to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, \mathbf{g} \leftarrow \mathsf{Next}(\tau)$. 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 $i, \mathbf{g}_1$. 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 $\tau \leftarrow \mathsf{Leak}_1(p_1, \ldots, p_n)$. Fix τ.
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 $(\mathbf{PShare}, \mathbf{PRec})$ 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, share $m_\$$ to obtain $l_\$, r_\$$, then generate and fix shares of $l_\$$ and $r_\$$. 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 shares of real 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 shares of real 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 $m, m_\$ \in \mathbb{F}_0$, the statistical distance in between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ is at most ϵ_0.

Proof: Assume towards contradiction that there exists m, $m_\$ \in \mathbb{F}_0$, and a distinguisher $\mathcal{D}$ that is successful in distinguishing $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ with probability greater than ϵ_0. We use the reduction and such a distinguisher $\mathcal{D}$ to construct a distinguisher $\mathcal{D}_1$ that violates the statistical secrecy of $(2, 2, \epsilon_0)$ secret sharing scheme $(\mathbf{NMEnc}, \mathbf{NMDec})$ for secrets m and $m_\$$. 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** $\mathcal{D}_1$: On input $\hat{r} \in \mathbb{F}_1$, sample $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(\hat{r})$. Using the fixed values, proceed as in the reduction by running the leakage protocol $\mathsf{Leak}_1(p_1, \ldots, p_n)$ to obtain leakage transcript τ. For each $i \in \{i_1, i_2\}$, compute $\widetilde{p_i} \leftarrow \mathbf{F}_i(p_i, \tau)$. Invoke the distinguisher $\mathcal{D}$ with the tampered secret $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$ and output its output.

Notice, if $\hat{r}$ hides the secret $m_\$$ under the 2-out-of-2 scheme, $(\mathbf{NMEnc}, \mathbf{NMDec})$, then $\mathcal{D}$ 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 $\mathcal{D}_1$ will be equal to the advantage of $\mathcal{D}$ in distinguishing these two hybrids, which is greater than ϵ_0 by assumption, violating the statistical secrecy of the 2-out-of-2 scheme. ■

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 $\mathsf{Leak}_2 \in (2, n, \mu + 1) - BCP$ and another distinguisher $\mathcal{D}_1$ 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 $(\mathbf{LShare}, \mathbf{LRec})$ 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 Next to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, \mathbf{g} \leftarrow \mathsf{Next}(\tau)$. 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 $i, \mathbf{g}_2$. 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 up to two shares of l can be fully leaked, the above leakage protocol belongs to the class $(2, n, \mu + 1) - BCP$.
3) (**Distinguisher** $\mathcal{D}_1$): On input leakage transcript $\tau \circ l_{i_1} \circ l_{i_2}$, for each $i \in \{i_1, i_2\}$, compute $\widetilde{l_i} \circ \widetilde{tr_i} \circ \widetilde{p_i} \leftarrow \mathbf{f}_i(l_i \circ tr_i \circ p_i, \tau)$. Invoke the distinguisher $\mathcal{D}$ with the tampered secret $\mathbf{PRec}(\widetilde{p_{i_1}}, \widetilde{p_{i_2}})$ and output its output.

Notice, in 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 $\mathcal{D}_1$ 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 10) are statistically indistinguishable, specifically,

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \approx_{\epsilon_0+\epsilon_3} \mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak_1},\mathbf{F},\mathbf{T}}$$

By the ϵ_7-non malleability of the scheme

$(\mathbf{PShare}, \mathbf{PRec})$, there exists a simulator $\mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak}_1,\mathbf{F},\mathbf{T}}$ such that

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak}_1,\mathbf{F},\mathbf{T}} \approx_{\epsilon_7} \mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak}_1,\mathbf{F},\mathbf{T}}$$

We use the underlying simulator as our simulator, and let

$$\mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \equiv \mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak}_1,\mathbf{F},\mathbf{T}}$$

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 protocol 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 $(\mathbf{NMEnc}, \mathbf{NMDec})$.

1) **(Initial setup)**: Fix an arbitrary $m_\$$ and let $l_\$, r_\$ \leftarrow \mathbf{NMEnc}(m_\$)$. Run the sharing function $\mathbf{LShare}$ with input $l_\$$ to obtain $(tl_1, \ldots, tl_n) \leftarrow \mathbf{LShare}(l_\$)$. Run the sharing function $\mathbf{RShare}$ with input $r_\$$ to obtain $(tr_1, \ldots, tr_n) \leftarrow \mathbf{RShare}(r_\$)$. Run the sharing function $\mathbf{PShare}$ with input $m_\$$ to obtain $(tp_1, \ldots, tp_n) \leftarrow \mathbf{PShare}(m_\$)$. 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 $\tau \leftarrow \mathsf{Leak}(tShare_1, \ldots, tShare_n)$. For each $i \in T$, run $\mathbf{f}_i$ with input $tShare_i$ and transcript τ to obtain $\widetilde{tShare_i} \leftarrow \mathbf{f}_i(tShare_i, \tau)$. Parse $\widetilde{tshare_i}$ as $(\widetilde{tl_i}, \widetilde{tr_i}, \widetilde{tp_i})$. For each $i \in \{i_1, i_2\}$, *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* $p_i \leftarrow tp_i$. *Fix* the transcript τ.
2) The **tampering function $\mathbf{F}$** is defined as follows: On input $l \in \mathbb{F}_1$, sample the value of $l_{i_3}, \ldots, l_{i_t}$ satisfying the following properties (via brute force over all possibilities):-
   - The shares $\{l_i : i \in T\}$ hide the secret l under $(\mathbf{LShare}, \mathbf{LRec})$. Moreover, the distribution of these shares is identical to the distribution produced on running $\mathbf{LShare}$ with input l conditioned on the first two shares being l_{i_1} and l_{i_2}.
   - For each $i \in T \setminus \{i_1, i_2\}$, let $share_i$ be (l_i, r_i, p_i) using the fixed values of r_i and p_i. The leakage transcript obtained by using Leak on these sampled shares is identical to the fixed τ. Note that the leakage protocol Leak on n shares can be locally simulated using $\{share_i : i \in T \setminus \{i_1, i_2\}\}$ as leakage from other shares is fixed and can be read from the transcript τ.

   In case such a sampling is not possible, then `abort`. Otherwise, for each $i \in T \setminus \{i_1, i_2\}$, run the tampering function f_i with inputs $share_i$ and transcript τ to obtain tampered $\widetilde{share_i} \leftarrow \mathbf{f}_i(share_i, \tau)$. Parse $\widetilde{share_i}$ as $(\widetilde{l_i}, \widetilde{r_i}, \widetilde{p_i})$. Using the fixed values of $\widetilde{l_{i_1}}$ and $\widetilde{l_{i_2}}$, run the reconstruction function $\mathbf{LRec}$ 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 $\mathbf{G}$** is defined as follows: On input $r \in \mathbb{F}_1$, 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 $(\mathbf{RShare}, \mathbf{RRec})$. 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) using the fixed values of l_i and p_i. The leakage transcript obtained by using Leak on these sampled shares is identical to the fixed τ. Note that the leakage protocol Leak on n shares can be locally simulated using $\{share_i : i \in \{i_1, i_2\}\}$ as leakage from other shares is fixed and can be read from the transcript τ.
   - For each $i \in \{i_1, i_2\}$, run $\widetilde{\mathbf{f}_i}$ with input $share_i$ and transcript τ to obtain $\widetilde{share_i} \leftarrow \mathbf{f}_i(share_i, \tau)$. 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 of our LR NMSS scheme. This is not immediate, because the distribution of the n shares in real tampering experiment is statistically quite far from the n shares sampled in the above reduction. Moreover, in real experiment, tampering is preceded by leakage from all the n shares. Nevertheless, we achieve this using the following hybrid argument. We begin by fixing any $l_\$, r_\$$ encoding any arbitrary $m_\$$ 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 $m_\$$

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 $tl_1, \ldots, tl_n$ 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 $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.
3) $\mathbf{Hybrid_3}$: We only make one change in the preceding hybrid. In the initial setup, the tampered $\widetilde{l}$ is computed in the initial setup using $\widetilde{tl_1}, \ldots, \widetilde{tl_n}$, that is $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{tl_{i_1}}, \ldots, \widetilde{tl_{i_t}})$. Consequently, there is no need to invoke tampering function $\mathbf{F}(l)$. 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 $tr_1, \ldots, tr_n$ 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 $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$.
5) $\mathbf{Hybrid_5}$: We only make one change in the preceding hybrid. In the initial setup, the tampered $\widetilde{r}$ is computed in the initial setup using $\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}}$, that is, $\widetilde{r} \leftarrow \mathbf{RRec}(\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}})$. Consequently, there is no need to invoke tampering function $\mathbf{G}(r)$. 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 $l, l_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ 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 $\mathbf{Hybrid_1}$ and $\mathbf{Hybrid_2}$ with probability greater than ϵ_3. We use the reduction and such a distinguisher to construct a leak protocol $\mathsf{Leak}_2 \in (2, n, \mu + 1) - BCP$ and another distinguisher $\mathcal{D}_1$ 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**): 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 $(\mathbf{LShare}, \mathbf{LRec})$ 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 $i, \mathbf{g} \leftarrow \mathsf{Next}(\tau)$. 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 $\mathbf{g}(l_i, tr_i, tp_i)$ using fixed values tr_i and tp_i. The Next_2 function outputs $i, \mathbf{g}_1$. 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} ending 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 up to two shares of l can be fully leaked at the cost of one extra bit, the above leakage protocol belongs to the class $(2, n, \mu + 1) - BCP$.
3) (**Distinguisher** $\mathcal{D}_1$): 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 $\widetilde{tShare_i} \leftarrow \mathbf{f}_i(tShare_i, \tau)$, and parse $\widetilde{tShare_i}$ as $(\widetilde{l_i} \circ \widetilde{tr_i} \circ \widetilde{tp_i})$. For each $i \in \{i_1, i_2\}$, fix l_i and $\widetilde{l_i}$. Fix transcript τ. This completes the initial setup of two hybrids in consideration. Compute $\widetilde{l} \leftarrow \mathbf{F}(l)$ and $\widetilde{r} \leftarrow \mathbf{G}(r)$. 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 $\mathbf{Hybrid_1}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_2}$. Therefore the success probability of $\mathcal{D}_1$ 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: $\mathbf{Hybrid_2}$ is identical to $\mathbf{Hybrid_3}$.

Proof: The two hybrids differ in how $\widetilde{l}$ is computed. In $\mathbf{Hybrid_2}$, the function $\mathbf{F}$ samples shares of l, namely $l_{i_3}, \ldots, l_{i_t}$, such that $l_{i_1}, \ldots, l_{i_t}$ satisfy certain constraints. At a high level, observe that in $\mathbf{Hybrid_3}$, the values of $tl_{i_3}, \ldots, tl_{i_t}$ sampled in the initial setup of the reduction already satisfy all these constraints. Consequently there is no need for sampling these shares and $\widetilde{l}$ can be directly computed using these values.

In more detail, consider any fixing of $\tau, \{(l_i, \widetilde{l_i}) : i \in \{i_1, i_2\}\}$ and $\{(r_i, p_i) : i \in T \setminus \{i_1, i_2\}\}$. Conditioned on this fixing, the distribution of $\{tl_i : i \in T \setminus \{i_1, i_2\}\}$ sampled in the initial setup of the reduction is exactly identical to the distribution of $\{l_i : i \in T \setminus \{i_1, i_2\}\}$ sampled by $\mathbf{F}(l)$ (by design of $\mathbf{F}$). As the same tampering functions are used, the distribution of tampered shares

$\{\widetilde{tl}_i : i \in T \setminus \{i_1, i_2\}\}$ in the initial setup will be identical to distribution of $\{\widetilde{l}_i : i \in T \setminus \{i_1, i_2\}\}$ in $\mathbf{F}(l)$. As $\widetilde{l_{i_1}}$ and $\widetilde{l_{i_2}}$ are already fixed, the tampered $\widetilde{tl}$ that can be reconstructed from $\{\widetilde{tl}_i : i \in T\}$ in the initial setup will be identically distributed to the the tampered $\widetilde{l}$ that can be reconstructed from $\{\widetilde{l}_i : i \in T\}$ in $\mathbf{F}(l)$. As the two hybrids do not alter $\mathbf{G}$, the distribution of $\widetilde{r} \leftarrow \mathbf{G}(r)$ remains identical in both these cases. Consequently, the distribution of $\mathbf{NMDec}(\widetilde{tl}, \widetilde{r})$ will be identical to the distribution of $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$, completing the proof. ■

The above two claims also show that $\mathbf{F}(l)$ does not `abort` with probability at least ϵ_3.

Claim: For any $r, r_\$ \in \mathbb{F}_1$, the statistical distance in between $\mathbf{Hybrid_3}$ and $\mathbf{Hybrid_4}$ is at most ϵ_5.

Proof: These two hybrids differ in the initial setup phase. In $\mathbf{Hybrid_3}$ shares of $r_\$$ are fixed, while in $\mathbf{Hybrid_4}$ 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 $(\mathbf{RShare}, \mathbf{RRec})$. The reduction is described below:

1) (**Initial setup**): Fix $tl_1, \ldots, tl_n \leftarrow \mathbf{LShare}(l)$ 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 $(\mathbf{RShare}, \mathbf{RRec})$ 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 Next to obtain an index $i \in [n]$ and leakage function $\mathbf{g}$, namely $i, \mathbf{g} \leftarrow \mathsf{Next}(\tau)$. 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 $\mathbf{g}(tl_i, r_i, tp_i)$ using the fixed values of tl_i and tp_i. The Next_2 function outputs $i, \mathbf{g}_1$. Let τ denote the transcript, when the underlying leakage protocol Leak finishes (formalized by Next outputting $\perp$). At this point, we continue leaking and each party $i \in T$ iteratively computes $(\widetilde{tl_i} \circ \widetilde{r_i} \circ \widetilde{tp_i}) \leftarrow \mathbf{f}_i(tl_i \circ r_i \circ tp_i, \tau)$ and outputs $\widetilde{tl_i}$ as leakage. As a result, the final transcript of our leakage protocol Leak_2 will be $\tau \circ \widetilde{tl_{i_1}} \circ \ldots \circ \widetilde{tl_{i_t}}$. As t (tampered) 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** $\mathcal{D}_1$): On input leakage $\tau \circ \widetilde{tl_{i_1}} \circ \ldots \circ \widetilde{tl_{i_t}}$, compute $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{tl_{i_1}}, \ldots, \widetilde{tl_{i_t}})$ 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 $(\mathbf{RShare}, \mathbf{RRec})$ is $r_\$$, $\mathcal{D}$ will be invoked with input distributed according to $\mathbf{Hybrid_3}$. Otherwise, $\mathcal{D}$ will be invoked with distribution similar to $\mathbf{Hybrid_4}$. Therefore the success probability of $\mathcal{D}_1$ 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 $(\mathbf{RShare}, \mathbf{RRec})$. ■

Claim: $\mathbf{Hybrid_4}$ is identical to $\mathbf{Hybrid_5}$.

Proof: These two hybrids differ in how $\widetilde{r}$ is computed. In $\mathbf{Hybrid_4}$, the function $\mathbf{G}$ samples two shares of r, such that r_{i_1} and r_{i_2} satisfy certain constraints. At a very high level, observe that in $\mathbf{Hybrid_5}$, the values of tr_{i_1} and tr_{i_2} sampled in the initial setup of the reduction already satisfy all these constraints. Consequently there is no need for sampling shares in $\mathbf{G}(r)$ and $\widetilde{r}$ can be directly computed using the sampled values in the initial setup.

In more detail, consider any fixing of τ and $\{(l_i, \widetilde{l}_i, p_i) : i \in \{i_1, i_2\}\}$. Conditioned on this fixing, the distribution of $\{tr_i : i \in \{i_1, i_2\}\}$ sampled in the initial setup of the reduction is exactly identical to the distribution of $\{r_i : i \in \{i_1, i_2\}\}$ sampled by $\mathbf{G}(r)$ (by design of $\mathbf{G}$). As the same tampering functions, namely $\mathbf{f}_{i_1}$ and $\mathbf{f}_{i_2}$, are used, the distribution of tampered shares $\{\widetilde{tr_i} : i \in \{i_1, i_2\}\}$ in the initial setup will be identical to distribution of $\{\widetilde{nr_i} : i \in \{i_1, i_2\}\}$ in $\mathbf{G}(r)$. Consequently, the tampered $\widetilde{tr}$ that can be reconstructed from $\{\widetilde{tr_i} : i \in \{i_1, i_2\}\}$ in the initial setup will be identically distributed to the the tampered $\widetilde{r}$ that can be reconstructed from $\{\widetilde{nr_i} : i \in \{i_1, i_2\}\}$ in $\mathbf{G}(r)$. Moreover, the sampling of r_{i_1} and r_{i_2} in $\mathbf{G}(r)$ ensures that the fixing of $\widetilde{l_{i_1}}$ and $\widetilde{l_{i_2}}$ is satisfied. This ensures that the distribution of $\widetilde{l}$ remains identical in both these cases. Consequently, the distribution of $\mathbf{NMDec}(\widetilde{l}, \widetilde{tr})$ will be identical to the distribution of $\mathbf{NMDec}(\widetilde{l}, \widetilde{r})$, completing the proof. ■

The above two claims also show that $\mathbf{G}(r)$ does not `abort` with probability at least ϵ_5.

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_7. We use the reduction and such a distinguisher to construct a leak protocol $\mathsf{Leak}_2 \in (1, n, \mu_2) - BCP$ and another distinguisher $\mathcal{D}_1$ 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 $tl_1, \ldots, tl_n \leftarrow \mathbf{LShare}(l)$ and $tr_1, \ldots, tr_n \leftarrow \mathbf{RShare}(r)$.
2) (**Leak function** Leak_2): We now design a n party leakage protocol Leak_2 for $(\mathbf{PShare}, \mathbf{PRec})$ us-

ing 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 $i, \mathbf{g} \leftarrow \mathsf{Next}(\tau)$. 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 $\mathbf{g}(tl_i, tr_i, p_i)$ using fixed values tl_i and tr_i. The Next_2 function outputs $i, \mathbf{g}_2$. 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 $(\widetilde{tl_i} \circ \widetilde{tr_i} \circ \widetilde{p_i}) \leftarrow \mathbf{f}_i(tl_i \circ tr_i \circ p_i, \tau)$ and outputs as leakage $\widetilde{tl_i}$. Finally, party i_1 and i_2 iteratively output $\widetilde{tr_{i_1}} \circ \widetilde{tr_{i_2}}$ before terminating the leakage protocol. As a result, the transcript of our leakage protocol Leak_2 will be $\tau \circ \widetilde{tl_{i_1}} \circ \ldots \circ \widetilde{tl_{i_t}} \circ \widetilde{tr_{i_1}} \circ \widetilde{tr_{i_2}}$.
As two (tampered) shares of r require $2\log|\mathbb{F}_3|$ bits and t (tampered) 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 $(1, n, \mu_2) - BCP$.

3) (**Distinguisher** $\mathcal{D}_1$): On input $\tau \circ \widetilde{tl_{i_1}} \circ \ldots \circ \widetilde{tl_{i_t}} \circ \widetilde{tr_{i_1}} \circ \widetilde{tr_{i_2}}$, compute $\widetilde{l} \leftarrow \mathbf{LRec}(\widetilde{tl_{i_1}}, \ldots, \widetilde{tl_{i_t}})$ and $\widetilde{r} \leftarrow \mathbf{RRec}(\widetilde{tr_{i_1}}, \widetilde{tr_{i_2}})$. 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 $\mathcal{D}_1$ 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 $\tilde{l}$ 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 10) are statistically indistinguishable, specifically,

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \approx_{\epsilon_3+\epsilon_5+\epsilon_7} \mathbf{Tamper}_{\mathbf{m}}^{\mathbf{F},\mathbf{G}}$$

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 $(|T| \geq 3)$.

$$\mathbf{STamper}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}} \approx_{\epsilon_1+\epsilon_3+\epsilon_5+\epsilon_7} \mathbf{SSim}_{\mathbf{m}}^{\mathsf{Leak},\mathbf{f},\mathbf{T}}$$

As the the statistical distances between real and simulated experiments in the two cases are $(\epsilon_0 + \epsilon_3 + \epsilon_7)$ and $(\epsilon_1 + \epsilon_3 + \epsilon_5 + \epsilon_7)$, we use the relaxed bound $(\epsilon_0 + \epsilon_1 + \epsilon_3 + \epsilon_5 + \epsilon_7)$ as the worst case statistical error of our scheme $(\mathbf{NMShare}, \mathbf{NMRec})$. ■

Leakage-resilient NMSS scheme for authorized pairs.: Goyal and Kumar [22] 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 [17] to obtain a leakage-resilient NMSS scheme for authorized pairs. We sketch the proof for non-malleability: without loss of generality assume that we will use the first two shares for reconstruction. Suppose the adversary adaptively leaks from all the n shares and then uses this leakage to tamper with all the n shares independently. In our reduction to 2-out-of-2-LR-NMSS, we generate n 'fake' shares encoding a 'fake' secret 0. The two real shares corresponding to the 2-out-of-2 LR-NMSS scheme can be used to simulate the leakage on n shares by replacing the specific components of the first two 'fake' shares with the given '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 completing the reduction. Using a hybrid argument we can now swap every pair of 'fake' shares with their real shares, without statistically affecting the output of the tampering experiment. After all these 'pairs' are replaced we end up with the real tampering experiment, completing the proof.

ACKNOWLEDGEMENTS

Ashutosh Kumar thanks Vipul Goyal, Venkatesan Guruswami and Jonathan Lee for useful discussions. We also thank the reviewers for their valuable feedback.

Ashutosh Kumar is supported by Dean's Special Fellowship, NSF grant 1619348, NSF frontier award 1413955, US-Israel BSF grants 2012366, 2012378, and by 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 CAREER Award 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

[1] A. Kumar, R. Meka, and A. Sahai, "Leakage-resilient secret sharing," Electronic Colloquium on Computational Complexity (ECCC), Report 2018/200, 2018.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *AFIPS National Computer Conference (NCC '79)*. Los Alamitos, CA, USA: IEEE Computer Society, 1979, pp. 313–317.

[3] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[4] 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*, ser. STOC '89. New York, NY, USA: ACM, 1989, pp. 73–85.

[5] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs, "Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors," in *EUROCRYPT*, 2008, pp. 471–488.

[6] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 337–367.

[7] V. Goyal and A. Kumar, "Non-malleable secret sharing," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2018, pp. 685–698.

[8] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Annual International Cryptology Conference*. Springer, 2003, pp. 463–481.

[9] S. Micali and L. Reyzin, "Physically observable cryptography," in *Theory of Cryptography Conference*. Springer, 2004, pp. 278–296.

[10] S. Dziembowski and K. Pietrzak, "Intrusion-resilient secret sharing," in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*. IEEE, 2007, pp. 227–237.

[11] ——, "Leakage-resilient cryptography," in *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*. IEEE, 2008, pp. 293–302.

[12] J. Alwen, Y. Dodis, and D. Wichs, "Survey: Leakage resilience and the bounded retrieval model," in *International Conference on Information Theoretic Security*. Springer, 2009, pp. 1–18.

[13] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs, "Public-key encryption in the bounded-retrieval model," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 113–134.

[14] F. Davì, S. Dziembowski, and D. Venturi, "Leakage-resilient storage," in *International Conference on Security and Cryptography for Networks*. Springer, 2010, pp. 121–137.

[15] S. Goldwasser and G. N. Rothblum, "How to compute in the presence of leakage," *SIAM Journal on Computing*, vol. 44, no. 5, pp. 1480–1549, 2015.

[16] F.-H. Liu and A. Lysyanskaya, "Tamper and leakage resilience in the split-state model," in *CRYPTO*, 2012, pp. 517–532.

[17] D. Aggarwal, S. Dziembowski, T. Kazana, and M. Obremski, "Leakage-resilient non-malleable codes," in *Twelfth IACR Theory of Cryptography Conference (TCC 2015)*, 2015.

[18] V. Goyal, Y. Ishai, H. K. Maji, A. Sahai, and A. A. Sherstov, "Bounded-communication leakage resilience via parity-resilient circuits," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*. IEEE, 2016, pp. 1–10.

[19] F. Benhamouda, A. Degwekar, Y. Ishai, and T. Rabin, "On the local leakage resilience of linear secret sharing schemes," in *CRYPTO*. Springer, 2018, pp. 531–561.

[20] Y. T. Kalai and L. Reyzin, "A survey of leakage-resilient cryptography," *IACR Cryptology ePrint Archive*, vol. 2019, p. 302, 2019.

[21] V. Guruswami and M. Wootters, "Repairing reed-solomon codes," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2016, pp. 216–226.

[22] V. Goyal and A. Kumar, "Non-malleable secret sharing for general access structures," in *CRYPTO*. Springer, 2018, pp. 501–530.

[23] A. K. Chandra, M. L. Furst, and R. J. Lipton, "Multiparty protocols," in *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. ACM, 1983, pp. 94–99.

[24] L. Babai, N. Nisan, and M. Szegedy, "Multiparty protocols, pseudorandom generators for logspace, and time-space tradeoffs," *Journal of Computer and System Sciences*, vol. 45, no. 2, pp. 204–232, 1992.

[25] E. Kushilevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 2006.

[26] J. M. Phillips, E. Verbin, and Q. 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*. SIAM, 2012, pp. 486–501.

[27] M. Braverman, F. Ellen, R. Oshman, T. Pitassi, and V. Vaikuntanathan, "A tight bound for set disjointness in the message-passing model," in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013, pp. 668–677.

[28] M. Braverman and R. Oshman, "On information complexity in the broadcast model," in *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. ACM, 2015, pp. 355–364.

[29] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography (extended abstract)," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, 1991, pp. 542–552.

[30] Y. Dodis and D. Wichs, "Non-malleable extractors and symmetric key cryptography from weak secrets," in *STOC*, 2009, pp. 601–610.

[31] S. Dziembowski, K. Pietrzak, and D. Wichs, "Non-malleable codes," in *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, 2010, pp. 434–452.

[32] Y. Dodis, X. Li, T. D. Wooley, and D. Zuckerman, "Privacy amplification and nonmalleable extractors via character sums," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 800–830, 2014.

[33] S. Dziembowski, T. Kazana, and M. Obremski, "Non-malleable codes from two-source extractors," in *CRYPTO (2)*, 2013, pp. 239–257.

[34] D. Aggarwal, Y. Dodis, and S. Lovett, "Non-malleable codes from additive combinatorics," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM, 2014, pp. 774–783.

[35] M. Cheraghchi and V. Guruswami, "Non-malleable coding against bit-wise and split-state tampering," in *TCC*, 2014, pp. 440–464.

[36] E. Chattopadhyay, V. Goyal, and X. 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*, 2016, pp. 285–298.

[37] X. 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*. ACM, 2017, pp. 1144–1156.

[38] ——, "Non-Malleable Extractors and Non-Malleable Codes: Partially Optimal Constructions," in *34th Computational Complexity Conference (CCC 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 137. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 28:1–28:49.

[39] E. Chattopadhyay and D. Zuckerman, "Explicit two-source extractors and resilient functions," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 670–683.

[40] E. Chattopadhyay and X. Li, "Non-malleable extractors and codes for composition of tampering, interleaved tampering and more," *ECCC 2018, Report 70*, 2018.

[41] V. Goyal, A. Kumar, S. Park, S. Richelson, and A. Srinivasan, "Non-malleable commitments from non-malleable extractors," Manuscript, 2019.

[42] R. Ostrovsky, G. Persiano, D. Venturi, and I. Visconti, "Continuously non-malleable codes in the split-state model from minimal assumptions," in *Annual International Cryptology Conference*. Springer, 2018, pp. 608–639.

[43] F. R. Chung, "Quasi-random classes of hypergraphs," *Random Structures & Algorithms*, vol. 1, no. 4, pp. 363–382, 1990.

[44] R. Raz, "The bns-chung criterion for multi-party communication complexity," *Computational Complexity*, vol. 9, no. 2, pp. 113–122, 2000.

[45] A. A. Sherstov, "Communication lower bounds using directional derivatives," *J. ACM*, vol. 61, no. 6, pp. 34:1–34:71, Dec. 2014.

[46] R. Williams, "Nonuniform acc circuit lower bounds," *J. ACM*, vol. 61, no. 1, pp. 2:1–2:32, Jan. 2014.

[47] C. Murray and R. 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*. ACM, 2018, pp. 890–901.

[48] M. Karchmer and A. Wigderson, "On span programs," in *Structure in Complexity Theory Conference, 1993., Proceedings of the Eighth Annual*. IEEE, 1993, pp. 102–111.

[49] A. Beimel, "Secret-sharing schemes: a survey," in *International Conference on Coding and Cryptology*. Springer Berlin Heidelberg, 2011, pp. 11–46.

[50] I. Komargodski, M. Naor, and E. Yogev, "Secret-sharing for np," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2014, pp. 254–273.

[51] S. Blackburn, "Combinatorics and threshold cryptography," *Research Notes in Mathematics*, vol. 403, pp. 44–70, 1999.

[52] Y. Desmedt, "Some recent research aspects of threshold cryptography," in *Information Security*, E. Okamoto, G. Davida, and M. Mambo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 158–173.

[53] A. Rao, "Extractors for a constant number of polynomially small min-entropy independent sources," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 168–194, 2009.

[54] X. Li, "Three-source extractors for polylogarithmic min-entropy," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015, pp. 863–882.

[55] S. R. Blackburn, M. Burmester, Y. Desmedt, and P. R. Wild, "Efficient multiplicative sharing schemes," in *Advances in Cryptology — EUROCRYPT '96*, U. Maurer, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 107–118.

[56] M. L. Fredman, J. Komlós, and E. Szemerédi, "Storing a sparse table with 0 (1) worst case access time," *Journal of the ACM (JACM)*, vol. 31, no. 3, pp. 538–544, 1984.

[57] N. Alon, R. Yuster, and U. Zwick, "Color-coding," *Journal of the ACM (JACM)*, vol. 42, no. 4, pp. 844–856, 1995.

[58] M. Naor, L. Schulman, and A. Srinivasan, "Splitters and near-optimal derandomization," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*. IEEE, 1995, pp. 182–191.

[59] E. Chattopadhyay and X. Li, "Explicit non-malleable extractors, multi-source extractors and almost optimal privacy amplification protocols," *FOCS*, 2016.

[60] D. Aggarwal, I. Damgard, J. B. Nielsen, M. Obremski, E. Purwanto, J. Ribeiro, and M. Simkin, "Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures," in *Annual International Cryptology Conference*. Springer, 2019, pp. 510–539.

[61] S. Badrinarayanan and A. Srinivasan, "Revisiting non-malleable secret sharing," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 593–622.

[62] A. Srinivasan and P. N. Vasudevan, "Leakage resilient secret sharing and applications," in *Annual International Cryptology Conference*. Springer, 2019, pp. 480–509.

[63] E. Karnin, J. Greene, and M. Hellman, "On secret sharing systems," *IEEE Transactions on Information Theory*, vol. 29, no. 1, pp. 35–41, 1983.

[64] A.-C. Yao, "On acc and threshold circuits," in *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*. IEEE, 1990, pp. 619–627.

[65] J. Hastad and M. Goldmann, "On the power of small-depth threshold circuits," *Computational Complexity*, vol. 1, no. 2, pp. 113–129, 1991.