Constant Rate (Non-malleable) Secret Sharing Schemes Tolerating Joint Adaptive Leakage

Nishanth Chandran[*] Bhavana Kanukurthi [†]
Sai Lakshmi Bhavana Obbattu [‡] Sruthi Sekar[§]

October 9, 2020

Abstract

A Leakage Resilient Secret Sharing (LRSS) is a secure secret sharing scheme, even when the adversary obtains some (bounded) leakage on honest shares. Ideally, such schemes must be secure against adaptive and joint leakage queries - i.e., the adversary can make a sequence of adaptive leakage queries where each query can be a joint function of many of the shares. The most important parameters of interest are the *rate* ($= \frac{|secret|}{|longestshare|}$) and the *leakage rate* (ratio of the total allowable leakage from a single leakage query to the size of a share). None of the prior works tolerating such adaptive and joint leakage could attain a constant rate and constant leakage rate, even for the threshold access structure. An LRSS is *non-malleable* (LRNMSS) when an adversary cannot tamper shares in a way that the reconstructed secret is related to the original secret. Similar to LRSSs, none of the prior LRNMSS schemes in the information theoretic setting could attain a constant rate, even for the threshold access structure.

In this work, we provide the first *constant rate* LRSS (for the general access structure) and LRNMSS (for the threshold access structure) schemes that tolerate such joint and adaptive leakage in the information-theoretic setting. We show how to make use of our constructions to also provide constant rate constructions of leakage-resilient (and non-malleable) secure message transmission.

We obtain our results by introducing a novel object called *Adaptive Extractors*. Adaptive extractors can be seen as a generalization of the notion of exposure-resilient extractors (Zimand, CCC 2006). Such extractors provide security guarantees even when an adversary obtains leakage on the source of the extractor after observing the extractor output. We make a compelling case for the study of such extractors by demonstrating their critical use for obtaining adaptive leakage and believe that such an object will be of independent interest.

[*] Microsoft Research, India, Email: `nichandr@microsoft.com`.

[†] Department of Computer Science and Automation, Indian Institute of Science, Email: `bhavana@iisc.ac.in`. Research supported in part by Department of Science and Technology Inspire Faculty Award. Research grant by Microsoft Research, India.

[‡] Microsoft Research, India, Email: `oslbhavana@gmail.com`. This work was done, in part, while the author was affiliated with Department of Computer Science and Automation, Indian Institute of Science.

[§] Department of Mathematics, Indian Institute of Science, Email: `sruthi.sekar1@gmail.com`.

Contents

1 Introduction

Secret sharing schemes [Sha79, Bla79] are a fundamental cryptographic primitive and have many applications, such as in multi-party computation [BGW88, CCD88], and leakage-resilient circuit compilers [ISW03, FRR$^+$10, Rot12]. These are cryptographic primitives that allow a dealer to distribute a secret to N parties, such that only an authorized subset of parties can reconstruct the original secret and any unauthorized set of parties have no information about the underlying secret (*privacy*). For instance, in a threshold secret sharing scheme, a collection of t (for some threshold $t \leq N$) or more parties would be an authorized set and any collection of less than t parties would be unauthorized. Note that an implicit assumption is that the unauthorized set of parties has no information about secrets of the remaining shares. A rich study on leakage attacks initiated by Kocher [Koc96] tell us that this is an idealized assumption that may not hold in practice. Such leakage can be dangerous and completely break the security of the underlying primitive[1].

Leakage Resilient Secret Sharing (LRSS). Dziembowski and Pietrzak in [DP07] initiated a study of leakage resilience in secret sharing schemes and their work has received much attention (for example, [DDV10, LL12, ADKO15, GK18, BDIR18], [SV19, KMS19, ADN$^+$19, FV19, BFV19, KMZ20, CGGL20, BFO$^+$20]), wherein researchers have strived to improve various parameters such as its rate (defined as (message length)/(length of longest share)), leakage model as well as leakage rate (defined as (number of bits of leakage allowed)/(the size of a share)).

At a high level, in an LRSS, the adversary is allowed leakage on shares of the secret. This is captured by permitting the adversary to specify functions $\ell_1, \ell_2, \ldots$, and receive, in response, $\ell_i(sh_i)$ (where sh_i denotes the i^{th} share). Informally, security of an LRSS requires that privacy should hold even given this leakage. In our work, we are specifically interested in the setting where the adversary specifies which share to receive leakage from, in an adaptive manner - i.e., the adversary specifies i, ℓ_i and upon learning $\ell_i(sh_i)$, it may make the next leakage query by specifying j, ℓ_j. We explore the question of building adaptive LRSS with good rate as well as leakage rate. All previous work that considered adaptive leakage either required computational assumptions or suffered from poor rate $\mathcal{O}(\frac{1}{\text{poly}N})$ where N denotes the number of parties [KMS19, KMZ20]. Furthermore, we consider a strengthening of this model where the adversary is permitted to ask for leakage of shares jointly. In particular, we ask:

*Can we construct a **constant rate** LRSS scheme in a joint and adaptive leakage model?*

We answer this question in the affirmative by giving the first LRSS scheme for general access structures that achieves a constant rate (and a good leakage rate) while tolerating joint and adaptive leakage. Additionally, we show applications of our techniques to leakage resilient, non-malleable secret sharing schemes as well as to secure message transmission.

1.1 Our Results

Result 1: *We build the first constant-rate LRSS scheme, tolerating adaptive as well as joint leakage, for general access structures.*

We build LRSS schemes which are additionally resilient to tampering attacks. Specifically, we build LRSS schemes which are also non-malleable (LRNMSS) i,e., wherein, an adversary cannot

[1] For example, Guruswami and Wooters [GW16] show that Shamir's secret sharing scheme is completely insecure when the adversary gets some $t-1$ shares and just one-bit of leakage from other shares.

tamper shares in a way that the reconstructed secret is related to the original secret. The only LRNMSS schemes tolerating adaptive leakage in the information theoretic setting [BFV19, KMS19] suffer from poor rate ($\mathcal{O}(\frac{1}{\text{poly}(N)})$) and do not allow joint leakage.

Result 2: *We build the first constant-rate LRNMSS scheme, tolerating adaptive as well as joint leakage, for threshold access structures.*

We finally apply our techniques to the problem of secure message transmission (SMT) introduced in [DDWY93]. In this problem, there is a sender S who needs to transmit a message m to a receiver R, where S and R are connected by N independent wires. Perfect secrecy is guaranteed even in the presence of an adversary that can observe at most $t-1$ wires and perfect resiliency is guaranteed (i.e., receiver receives the correct m), even when the adversary can modify the messages sent on those t wires arbitrarily. In our work, we introduce the notion of leakage resilient SMT, in which an adversary is additionally allowed to make leakage queries from wires not under its control.

Result 3: We provide the first constructions of SMT protocols tolerating leakage. First, for the case of passive adversaries (i.e., adversaries who can view but not modify values on wires), we obtain leakage-resilient SMT protocols where the adversary can obtain leakage from messages sent on $t-1$ other wires in addition to viewing the complete contents on $t-1$ wires. Next, for the case of active adversaries, we obtain a leakage-resilient non-malleable SMT[2] protocol where the adversary can obtain leakage from messages sent on $t-4$ other wires in addition to viewing and completely modifying the contents on $t-1$ wires. Both these constructions once again enjoy constant rate and information-theoretic security.

1.2 Our Techniques

Our framework for the LRSS and LRNMSS considers the following leakage and tampering models:

- **Leakage Model (LRSS):** We allow the adversary to get a joint leakage on up to an unauthorized set of shares, adaptively. In total, we allow all the leakage queries to depend on at most an unauthorized set of shares, post which we allow the adversary to get full shares of a fresh unauthorized set as well.
- **Tampering Model (LRNMSS):** We allow our adversary to get a joint leakage on up to $t-4$ shares (where t is the threshold of the secret sharing scheme), adaptively. In total, all leakage queries are also on at most $t-4$ shares, post which the adversary can choose the tampering functions (independently acting on each share) and reconstruction set (to recover the tampered message).

Now we proceed to describe our key technical contributions. The starting point for our work is the LRSS compiler given by Srinivasan and Vasudevan in [SV19, Section 3.2.1], which transforms any secret sharing scheme into a leakage resilient one. Their compiler makes a critical use of randomness extractors [NZ96] which offer a mechanism to obtain uniform randomness from non-uniform randomness, using a short random seed. More formally, a randomness extractor Ext is a function that takes as input an n-bit entropic source W, a uniformly random d-bit string S (seed)

[2] The notion of non-malleable SMT without leakage was introduced in a recent work of [GK18]. We strengthen their adversarial model to incorporate leakage.

and outputs $\mathsf{Ext}(W;S)$ such that $\mathsf{Ext}(W;S)$ "looks uniform" to an unbounded adversary Eve even given the seed S. In the construction of [SV19], the uniformity of the output is critically used for guaranteeing leakage resilience of the scheme.

At a high level, the [SV19] compiler works as follows: it takes any secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ for a general access structure $\mathcal{A}$ and then:

- It samples shares $(sh_1, .., sh_N)$ of the message m using Share.
- It chooses an extractor seed s and splits s into $(s_1, .., s_N)$ using a 2-threshold secret sharing scheme.
- Now, for every sh_i, it chooses an extractor source w_i uniformly and computes $y_i = sh_i \oplus \mathsf{Ext}(w_i; s)$.
- Finally, it outputs the final shares $\{share_i\}$ as $\{(w_i, y_i, s_i)\}$.

The leakage resilience of the scheme is reduced to the extractor security on some source w_i. Intuitively, by picking s_i uniformly at random and independent of s, the leakage function on $\{share_i\}$, can be answered as an auxiliary leakage query on the source w_i. Furthermore, since s is revealed (in the extractor security game), the reduction can pick the other s_j values in a consistent manner. (Note that it is important that the auxiliary leakage query on w is independent of s; however, there is a dependence on s via y_i. The authors get rid of this dependence by using an additional one-time pad to mask y_i. For the purpose of this exposition, however, we will ignore the issue of dependence via y_i and focus only on s_i.)

To build our LRSS we first identify the critical bottleneck when trying to prove adaptive leakage resilience. In the standard extractor security game, once the distinguisher is given either the output of the extractor or a uniform random string, the game does not permit any further leakage on the source w. However, in the case of adaptive LRSS, the adversary may choose to ask for further leakage on the i^{th} share (where w_i is the source with respect to which the extractor security is being played). To overcome this bottleneck, we ask the following question: *Can we build extractors which allow for leakage queries which depend even on the seed as well as the output given to the distinguisher?*

This brings us to the work of Zimand [Zim06], who introduced the notion of exposure resilient extractors, which allow for some specific, restricted adaptive leakage on the source. Specifically, Zimand's extractors allow the adversary to adaptively learn few bits of the source W (up to n^{δ} bits for some $\delta < 1$); the adversary can determine which bits to query based on an arbitrary function of the extractor output. For our application to LRSS, unfortunately, this limited form of leakage (i.e., few bits of w) is insufficient. In particular, we require a randomness extractor that allows for a (bounded) leakage that is an arbitrary function of the source. Furthermore, this leakage may be dependent on an arbitrary function of the output. With this motivation, we put forth a general notion of *adaptive extractors* with respect to arbitrary adaptive leakage on the source. We then show that every randomness extractor is also an adaptive extractor with respect to a leakage family depending arbitrarily on the source and the output, with some loss in parameters. We demonstrate that, in spite of the loss in parameters that adaptivity incurs, such extractors can be extremely powerful. In particular, we use them to build our constant-rate adaptive LRSS schemes.

Leakage Resilient Non-malleable Secret Sharing. We now explain how to apply our techniques to obtain our LRNMSS scheme. We begin with the non-malleable secret sharing scheme of [GK18], which uses a leakage resilient secret sharing scheme, a threshold secret sharing scheme and a 2-split-state non-malleable code[3] as building blocks. The scheme is as follows: the message m is encoded using a 2-split-state non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ to (L, R). Now, L is secret shared using a t-out-of-n threshold secret sharing scheme to get $(\mathsf{L}_1, \cdots, \mathsf{L}_n)$ and R is secret shared using a 2-out-of-n leakage resilient secret sharing scheme to get $(\mathsf{R}_1, \cdots, \mathsf{R}_n)$. The non-malleability of this scheme can be reduced to the non-malleability of the underlying non-malleable code, as long as, we can capture the independent tampering on the shares as split-state tampering on L and R. This relies on two key features of the construction: First, the tampering of R can be captured to be independent of L, as the threshold of the LRSS used to secret share R is 2, which is lower than t. Hence, any 2 shares of L, will still hide it and this can be used to determine the tampered right share. Second, the tampering of L requires t of the tampered shares of L, which depend on the corresponding shares of R and needs to be obtained as leakage on the shares of R.

Ideally, to add leakage resilience to the above NMSS, we would like to replace the LRSS in the construction above with our joint adaptive LRSS to get an LRNMSS which is not only secure against joint and adaptive leakage and tampering but also preserves the rate asymptotically[4]. Unfortunately, this does not work as-is. The reason is that we would only be able to get leakage from a single share now, as the LRSS used to share R has threshold 2 and can only allow for an adaptive query on a single share. Suppose we took both the schemes to be a t-threshold LRSS. Then, the tampering of L depends on t shares of R and vice versa. Hence, to get the split-state tampering, we need to get the tampered left shares as leakage from shares of R and vice versa. This brings us to the first challenge. We require leakages of size $|\mathsf{L}_i|$ from the second LRSS and of size $|\mathsf{R}_i|$ from the first LRSS, which means the sizes of shares of L and R need to be simultaneously larger than each other (to accommodate the desired leakage size). Clearly, this is contradictory. Hence, it is necessary that the threshold of the two schemes differ. Our final LRNMSS scheme is the scheme described above with the following changes: use a t-out-of-n LRSS to share L and a $(t-1)$-out-of-n LRSS to share R.

For proving the leakage resilient non-malleability of this scheme, we need to simulate the tampering as split-state tampering and also simulate the leakage queries, independent of the message. The three key observations which capture the crux of our proof are: First, the joint adaptive leakage queries made in the first phase fit the leakage model of the underlying LRSS and hence can be simulated using that. Second, the tampering of R requires $t-1$ of the shares of L, which can be obtained as a full share query on the first LRSS scheme (as its threshold is t). Third, the tampering of L requires t of the shares of R, which exceeds the threshold of the second LRSS. But we can get up to $t-2$ full shares of R. We get around this by obtaining two tampered shares of L as leakage queries on the second LRSS. Note that, keeping the underlying leakage model in mind, we restrict the number of leakage queries to be on at most $t-4$ shares, so that the 2 additional leakage queries (from the second LRSS) can be obtained. This captures the structure of our proof but combining the observations to a formal security proof requires a careful setting of parameters as well as some additional subtle properties from the underlying LRSS.

Applying our LRSS and LRNMSS to the context of SMT, directly gives us leakage-resilient

[3] A 2-split-state non-malleable code (NMC) gives a guarantee that if the codeword L, R of a message m is tampered such that L and R are tampered arbitrarily but independent of each other, then the recovered m' will either be the same as m or will be independent of it.

[4] We can instantiate the scheme with a constant rate NMC. Hence, the rate is determined by the rate of the LRSS used.

SMT protocols appropriately in the passive and active setting respectively. Finally, we believe that our notion of adaptive extractors and their application to other cryptographic primitives is of independent interest.

1.3 Related Work

We first list out some of the parameters that are relevant to LRSS schemes:

- *Rate*: This is defined as $\frac{\mathsf{messagelength}}{\mathsf{sharelength}}$.
- *Global Limit*: This refers to the total number of shares on which the leakage queries can depend on.
- *Per-query Limit*: This refers to the number of shares that a specific query can depend on.
- *Per-query Leakage Rate*: This is the ratio of the total allowable leakage from a single leakage query to the size of a share.

The problems of leakage resilient and non-malleable secret sharing have seen a flurry of activity in recent times [LL12, BDIR18, GK18, BS19, SV19, ADN$^+$19, FV19, BFV19, KMS19, LCG$^+$19], [KMZ20, CGGL20, BFO$^+$20]. Here we compare our work with only the most relevant works in this area.

The only prior LRSS schemes allowing for a joint and adaptive leakage model are [KMS19, KMZ20, CGGL20]. Similar to our model, the model of [KMZ20] (for general access structures) allows the adversary to make joint and adaptive queries on disjoint sets of shares of size up to an unauthorized set. In other words, for the threshold access structure, both works require any particular query to depend on at most $t-1$ shares and the sets of the shares across queries to be disjoint. However, [KMZ20] permits making leakage queries on all N shares while we restrict it to the size of an unauthorized set. While this comes at the expense of the rate – [KMZ20]'s has rate $\mathcal{O}(1/N)$ against our constant rate–, the biggest drawback of the [KMZ20] is that their allowable leakage is very poor. In particular, while we allow a leakage of a constant fraction of size of a single share per query, [KMZ20] allows a leakage of $\mathcal{O}(\frac{1}{N})$ of a share. To put this in context, even if [KMZ20] makes independent leakage on all shares, the maximum number of bits they can leak is at most a constant fraction of the size of a single share, while we can leak close $(t-1)$ times a constant fraction of the size of a single share!

Furthermore, the works of [KMS19, KMZ20, CGGL20] consider a stronger joint leakage model, allowing leakage queries on overlapping sets of shares, but it comes at an expense of the rate and leakage rate, both of which are poor for these schemes and at the expense of achieving it for the threshold access structures in the case of [KMZ20] and for the N-out-of-N setting in the case of [CGGL20]. We refer the readers to Table 1 for the exact parameters attained by these works.

In Table 1, we offer a complete comparison between our work and the relevant prior works, with respect to specific parameters of interest (which were defined above).

Work*	Rate	Joint Leakage	Global Limit	Per Query Limit	Leakage Rate (per query)	Adaptive	Full shares
[SV19]	$1/3$	No	N	1	≈ 1	No	Unauthorized
[ADN+19]	$O(1/N)$	No	N	1	$\approx (1-c)$	No	Unauthorized
[KMS19]	$O\left(\frac{1}{poly(N)}\right)$	Yes (overlapping)	N	$\log(N)$	$\approx \Theta\left(\frac{1}{poly(N)}\right)$	Yes	$\log(N)$
[KMZ20]	$O\left(\frac{1}{N}\right)$	Yes	N	$t-1$	$\approx \Theta\left(\frac{1}{N}\right)$	Yes	Unauthorized
[KMZ20] (threshold)	$O\left(\frac{1}{poly(N)}\right)$	Yes (overlapping)	N	$O\left(\frac{t}{\log(t)}\right)$	$\approx \Theta\left(\frac{1}{poly(N)}\right)$	Yes	Unauthorized
[CGGL20] (N-out of-N)	$O\left(\frac{1}{poly(l_{msg})}\right)$	Yes (overlapping)	N	$0.99N$	$\approx \Theta\left(\frac{1}{poly(l_{msg})}\right)$	Yes	Unauthorized
Our result	$\Theta(1)$	Yes	$t-1$	$t-1$	$\approx \Theta(1)$	Yes	Unauthorized**

Table 1

- *All works mentioned here are information-theoretic.
- ** For our result, the unauthorized queries cannot overlap with the leakage queries.
- N is the total number of parties, t is the corruption threshold, c is a small constant and l_{msg} is the message length.
- All schemes (except the joint overlapping scheme of [KMZ20] (threshold) and [CGGL20] (N-out-of-N)) are for general access structure. For ease of exposition, we only compare threshold schemes in the table.
- **Full Shares**: Number of complete shares that an adversary can see (at the end of all leakage queries, in the adaptive schemes).
- **Colour coding**: Red indicates a weaker feature and green the stronger one with respect to a property.

We now discuss the most relevant works on leakage-resilient non-malleable secret sharing.

1. In the information theoretic setting, the only known LRNMSS schemes are [KMS19, BFV19], both of which achieve a rate of $O(1/\text{poly}(N))$. Their model allows the adversary to get independent and adaptive leakage before allowing a single independent tampering (each share is tampered independent of the other shares) query. In comparison, we allow the adversary to get adaptive and joint leakage on at most $t-4$ shares in total before allowing a single independent tampering query, and we achieve a constant rate. While our leakage model is incomparable to [KMS19, BFV19], we get the first *constant rate* scheme for a joint and adaptive leakage model.

2. In the computational setting, there are several works [BFV19, FV19, BFO$^+$20] which give a LRNMSS in a joint and adaptive leakage model with continuous non-malleability in a joint tampering model, of which the most recent work of [BFO$^+$20], in combination with the compiler from [FV19] gives a rate 1 scheme. There are several variants of joint leakage considered in these works (allowing overlapping queries), but all variants have a poor rate. We refer the readers to Table 2 for the exact parameters achieved by these schemes.

In Table 2 below, we present a detailed comparison of our work with the most relevant NMSS schemes[5].

[5] All the schemes mentioned here are in the compartmentalized model or the split-state model which assumes that the adversary cannot tamper all shares together. The work of [LCG$^+$19] is the only one to consider the non-compartmentalized model and give a leakage resilient non-malleable secret sharing scheme for adaptive affine leakage and affine tampering dependent on the leakage.

Work	Access Structure	IT/ Comp	Rate	Leakage	Global Limit	Per Query Limit	Adaptive	Tampering
[BS19,SV19]	General*	IT	$\Theta(1)$	No	N.A.	N.A.	N.A.	Concurrent
[ADN+19]	General*	IT	$O(1/N)$	No	N.A.	N.A.	N.A.	Concurrent**
[KMS19, BFV19]	General	IT	$O\left(\frac{1}{poly(N)}\right)$	Independent	N	1	Yes	Single
[BFO+20] (+[FV19])	Threshold	Comp	1	Joint	N	$t-1$	Yes	Continuous (and joint)
[BFO+20] (+[FV19])	General	Comp	1	Joint (overlapping)	N	$O(\sqrt{\log N})$	Yes	Continuous (and joint)
Our result	Threshold	IT	$\Theta(1)$	Joint	$t-4$	$t-4$	Yes	Single

Table 2

- *[BS19, SV19] and [ADN$^+$19] are for 4 and 3-monotone access structures.
- **[ADN$^+$19] has a stronger concurrent tampering model than [BS19].
- N represents the number of parties and t represents the threshold.
- **Colour coding**: Red indicates a weaker feature, green the stronger one with respect to a property and yellow represents an intermediate feature.

1.4 Organization of the Paper

We provide the preliminaries and definitions in Section 2. Then, we define the leakage and tampering models in Section 3. We give our constructions of the leakage resilient secret sharing and leakage resilient non-malleable secret sharing in Sections 4 and 5 respectively. Finally, we give our application to secure message transmission in Section 6.

2 Preliminaries and Definitions

2.1 Notation

We denote the security parameter by κ. For any two sets S and S', $S \backslash S'$ denotes the set of elements that are present in S, but not in S'. For any natural number n, $[n]$ denotes the set $\{1, 2, \cdots, n\}$. $s \in_R S$ denotes uniform sampling from set S. $x \leftarrow X$ denotes sampling from a probability distribution X. The notation $\Pr_X[x]$ denotes the probability assigned by X to the value x. $x||y$ represents concatenation of two binary strings x and y. $|x|$ denotes length of binary string x. U_l denotes the uniform distribution on $\{0,1\}^l$. All logarithms are base 2. Any "For loop" of the form, "For $j = a$ to b" will only be executed iff $a \leq b$. If S is a subset of $[n]$:

- If $x_1, .., x_n$ are some variables or elements, then x_S denotes the set $\{x_i \text{ such that } i \in S\}$.
- For some function f outputting n values $y_1, \cdots, y_n$ on input x, $f(x)_S$ denotes $(y_i)_{i \in S}$.
- If $T_1, .., T_n$ are sets, then T_S denotes the union $\cup_{i \in S} T_i$.

We give standard definitions of statistical distance and entropy along with some preliminary lemmata of the same in Appendix A.1.

2.2 Adaptive Extractors

Extractors(introduced by Nissan and Zuckerman [NZ96]) output an almost uniform string from a (n, t_{ext})-source, using a short uniform string, called *seed*, as a catalyst. Average-case extractors are extractors whose output remains close to uniform, even given the seed and some auxiliary information about the source (but independent of the seed), whenever the source has enough average entropy given the auxiliary information. We formally define them as below.

Definition 1. *[DORS08] Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^l$ *be a polynomial time computable function. We say that* Ext *is an efficient average-case* $(n, t_{ext}, d, l, \epsilon)$*-strong extractor if for all pairs of random variables* (W, Z) *such that* W *is an* n*-bit string satisfying* $\widetilde{\mathbf{H}}_\infty(W|Z) \geq t_{ext}$*, we have*

$$\mathsf{Ext}(W; U_d), U_d, Z \approx_\epsilon U_l, U_d, Z$$

We now formally define the notion of adaptive extractors, which guarantee that the extractor output is statistically close to uniform even given some adaptive leakage on the source w, dependent on the seed s or the extractor output(or the uniform string) y. While expecting this guarantee, one cannot leak out arbitrary functions on the source, seed and the output. For example, given w, s, y, adaptive leakage of the bit whether $\mathsf{Ext}(w, s) = y$ helps distinguish the extractor output from a uniform string with very high probability. Hence, we give the definition specific to a leakage family.

Definition 2. *We say* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^l$*, is an* $(n, t_{ext}, d, l, \epsilon, \delta)$*-**adaptive extractor** with respect to a function family* $\mathcal{F}$*, if* Ext *is an* $(n, t_{ext}, d, l, \epsilon)$*-average case strong extractor and for every function* $f \in \mathcal{F}$*,*

$$Z, U_d, f(W, \mathsf{Ext}(W; U_d), U_d), \mathsf{Ext}(W; U_d) \approx_\delta Z, U_d, f(W, U_l, U_d), U_l$$

We now show that every extractor is in fact an adaptive extractor for the family where adaptive leakage only depends on the source and the extractor output (or the uniform string) with some loss of parameters. In fact, if the adaptive leakage function does not depend on the entire extractor output, we get better adaptivity error. We now explicitly define this family which specifies the amount of leakage as well as the amount of dependence of the leakage on the extractor output.

$$\mathcal{F}_{a,m} = \{f' : \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^m | \exists\ f : \{0,1\}^{n'+l} \rightarrow \{0,1\}^a \text{ and}$$
$$g : \{0,1\}^{n+a} \rightarrow \{0,1\}^m \text{ such that } \forall w, y,\ f'(w, y) = g(w, f(z, y))\}^{6,7}$$

'm' denotes the length of the adaptive leakage. 'a' denotes the number of bits of the extractor output (or the uniform string) the adaptive leakage depends on. This is captured by requiring that every function f' has an equivalent representation in terms of some g and f such that $f'(w, y) = g(w, f(z, y))$ where f's output is only a bits long. Here w, z and y should be interpreted as the source, any auxiliary information purely dependent on source and the extractor output(or the uniform string) respectively.

Theorem 1. *Every* $(n, t_{ext}, d, l, \epsilon)$*- average case extractor* Ext *is an* $(n, t_{ext}+m, d, l, \epsilon, 2^{a+2}\epsilon)$*- adaptive average case extractor with respect to the family* $\mathcal{F}_{a,m}$*, for any* $t_{ext} + m \leq n$ *and* $a \leq l$*.*

[6] z denotes any auxiliary information about w the satisfies the appropriate average entropy requirement. n' denotes length of the auxiliary information. Readers may ignore z if needed.

[7] f', f and g can be randomized functions as well, where the randomness of the function is true independent randomness.

Proof of the theorem can be found in Appendix A.2

We now show that there exists an explicit (efficient) instantiation for the adaptive extractor in Theorem 1. Specifically, we show how to instantiate the extractor to get meaningful adaptivity error even when the leakage depends on the entire extractor output. We will use the extractor due to Guruswami et.al in this instantiation, which is as follows.

Lemma 1. *[GUV07] For every constant $\nu > 0$ all integers $n \geq t_{ext}$ and all $\epsilon \geq 0$, there is an explicit (efficient) $(n, t_{ext}, d, l, \epsilon)-$strong extractor with $l = (1-\nu)t_{ext} - \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$ and $d \leq \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$.*

With the following lemma we show that one can appropriately set the parameters of the above extractor to get negligible adaptivity error while extracting and adaptively leaking constant fraction of bits of the min-entropy using a seed of the same order. This particular instantiation will be useful for our LRSS construction.

Lemma 2. *For every n, t_{ext}, m such that $n \geq t_{ext} + m$ and $t_{ext} = \omega(\log n)$, there is an explicit (efficient) $(n, t_{ext} + m, d, l, \epsilon, \delta)-$adaptive extractor with respect to $\mathcal{F}_{l,m}$ with $\epsilon = 2^{-\Omega(t_{ext})}$, $\delta = \epsilon 2^{l+2} = 2^{-\Omega(t_{ext})}$, $l = \Theta(t_{ext})$ and $d = \Theta(t_{ext})$.*

Proof of the lemma can be found in Appendix A.3

We also require the following property of strong average case extractors, which essentially states that the same seed can be used to extract from multiple independently drawn sources.

Lemma 3. *If Ext is an $(n, t_{ext}, d, l, \epsilon)$-strong average case extractor, then for any $h \geq 1$ and any $\delta > 0$, Ext_h is an $(hn, (h-1)n + t_{ext}, d, n \cdot l, h\epsilon)$-strong average case extractor, where Ext_h is defined as follows:*

- *Parse w as $w_1 || w_2 || \cdots || w_h$ (where w_i is n-bit long, for all $i \in [h]$)*
- *Output $\mathsf{Ext}(w_1; s) || \mathsf{Ext}(w_2; s) || \cdots || \mathsf{Ext}(w_h; s)$*

The proof of Lemma 3 is given in Appendix A.4.

2.3 Secret Sharing Schemes

Secret sharing schemes provide a mechanism to distribute a secret into shares such that only an authorized subset of shares can reconstruct the secret and any unauthorized subset of shares has "almost" no information about the secret. We now define secret sharing schemes formally.

Definition 3. *Let $\mathcal{M}$ be a finite set of secrets, where $|\mathcal{M}| \geq 2$. Let $[N]$ be a set of identities (indices) of N parties. A sharing function $\mathsf{Share} : \mathcal{M} \rightarrow (\{0,1\}^l)^N$ [8] is a $(\mathcal{A}, N, \epsilon_s)$- **secret sharing scheme** with respect to a monotone access structure[9] $\mathcal{A}$ if the following two properties hold :*

[8] Can be interpreted as each party receiving a share of length l bits.

[9] $\mathcal{A}$ is a monotone access structure if for all A, B such that $A \subset B \subseteq [N]$ and $A \in \mathcal{A}$, it holds that $B \in \mathcal{A}$. Throughout this paper whenever we consider a general access structure, we mean a monotone access structure.

1. ***Correctness**: The secret can be reconstructed by any set of parties that are part of the access structure $\mathcal{A}$. That is, for any set $T \in \mathcal{A}$, there exists a deterministic reconstruction function* $\mathsf{Rec} : (\{0,1\}^l)^{|T|} \to \mathcal{M}$ *such that for every $m \in \mathcal{M}$,*

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

   where the probability is over the randomness of the Share *function and if $(sh_1, .., sh_N) \leftarrow$* $\mathsf{Share}(m)$*, then* $\mathsf{Share}(m)_T$ *denotes $\{sh_i\}_{i \in T}$. We will slightly abuse the notation and denote* Rec *as the reconstruction procedure that takes in $T \in \mathcal{A}$ and* $\mathsf{Share}(m)_T$ *as input and outputs the secret.*

2. ***Statistical Privacy**: Any collusion of parties not part of the access structure should have "almost" no information about the underlying secret. More formally, for any unauthorized set $U \notin \mathcal{A}$, and for every pair of secrets $m, m' \in \mathcal{M}$,*

   $$\Delta((\mathsf{Share}(m))_U; (\mathsf{Share}(m'))_U) \leq \epsilon_s$$

An access structure $\mathcal{A}$ is said to be t-threshold if and only if $\mathcal{A}$ contains all subsets of $[N]$ of size atleast t.
***Rate** of a secret sharing scheme is defined as $\frac{\textit{message size}}{\textit{share size}}$(which would be equal to $\frac{\log |\mathcal{M}|}{l}$).*

We now study a stronger privacy requirement, *adaptive privacy* (introduced by Bellare and Rogaway [BR07][10])

2.3.1 Adaptive Privacy

Statistical privacy captures privacy against any non-adaptively chosen unauthorised set U. *Adaptive privacy* preserves privacy even when the choice of U to be adaptive, which means the following. Let $U = \{i_1, .., i_q\}$. We say i_j is chosen adaptively, if its choice depended on $\{share_j\}_{j \in \{i_1, .., i_{j-1}\}}$. The choice of which share to query next depends on all the previously observed shares. We give the formal definition below A.6 and also prove (in the following lemma) that adaptive privacy can be guaranteed for any secret sharing scheme that satisfies statistical privacy whose proof is given in Appendix A.6.
We say a $(\mathcal{A}, N, \epsilon_s)$-secret sharing scheme satisfies adaptive privacy with error ϵ_{adp} if for any distinguisher $\mathcal{D}$, the advantage in the following game is at most ϵ_{adp}.

$\mathsf{Game}_{\mathsf{Ad-Privacy}}$: For any arbitrary distinct messages $m_0, m_1 \in \mathcal{M}$

1. $(share_1, \cdots, share_N) \leftarrow \mathsf{Share}(m_b)$ where $b \in_R \{0,1\}$
2. For $j = 1$ to q [11]
   - $\mathcal{D}$ queries on a distinct index i_j(such that $i_{[j]} \notin \mathcal{A}$) and receives $share_{i_j}$
3. $\mathcal{D}$ outputs the guess b' for b and wins if $b = b'$

[10] In [BR07], the authors refer to adaptive privacy as privacy against dynamic adversaries.

[11] q is arbitrary and chosen by $\mathcal{D}$. It need not be chosen a-priori. We only use it to denote the total number queries made by $\mathcal{D}$

Lemma 4. *Any $(\mathcal{A}, N, \epsilon_s)$- secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *for $\mathcal{M}$ satisfies adaptive privacy with error $2p \cdot \epsilon_s$, where p is the cardinality of the largest unauthorized set with respect to $\mathcal{A}$.*

In our proofs we would frequently need to re-sample shares of a messages m consistent w.r.t. some a-priori fixed shares. For the sake of completeness, We formally explain this procedure in Appendix A.5.

3 Leakage and Tampering Model

In this section, we describe the leakage model for our LRSS and the tampering model for our LRNMSS and formally define them.

3.1 Leakage Resilient Secret Sharing

Leakage-resilience of a secret sharing scheme is defined specific to a leakage model/ leakage family. We begin by formally defining leakage-resilience and then describe the leakage model.

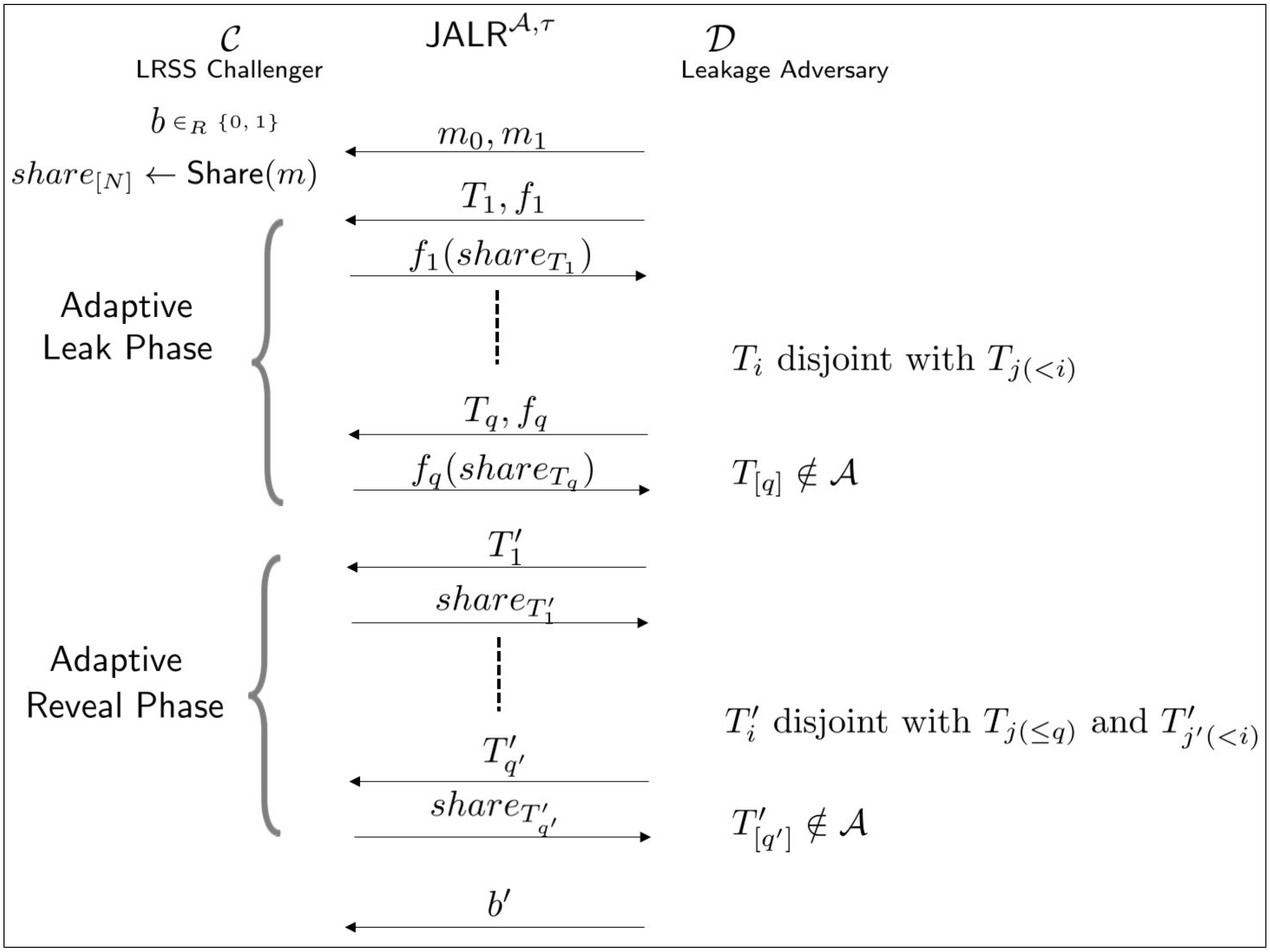
Definition 4. *An $(\mathcal{A}, N, \epsilon_s)$-secret sharing scheme is said to be an $(\mathcal{A}, N, \epsilon_s, \epsilon_l)$- **leakage resilient secret sharing scheme** against a leakage family $\mathcal{F}$ if for all functions $f \in \mathcal{F}$ and for any two messages m, m',*

$$\Delta(f(Share(m)); f(Share(m'))) \leq \epsilon_l$$

We now describe the leakage model. Informally, the adversary can adaptively make joint leakage queries on shares of an unauthorized set. We allow the leakage queries to be arbitrary and depend on up to an unauthorized set of shares (both in total for all queries and per query). While we describe the model for leakage queries on non-overlapping sets of shares (for ease of exposition), we show that the model can in fact capture the model where the adversary makes queries on overlapping sets (going back and forth) with a loss in the leakage rate. The equivalence of these models are described in detail in Appendix B. Moreover, while the model description only allows the adversary to make one query on a set of shares and then move to the next non-overlapping set, it is indeed equivalent to allowing the adversary to make adaptive queries on a set of shares before moving to the next (as we are in the information theoretic setting). The above two observations show that the model captures any adaptive overlapping leakage queries within an unauthorized set (with some loss in the amount of leakage allowed).

We formally introduce the model through a game based definition. Let $(\mathsf{LRShare}, \mathsf{LRRec})$ (where $\mathsf{LRShare} : \mathcal{M} \rightarrow (\{0,1\}^l)^N)$ be a secret sharing scheme for an access structure $\mathcal{A}$ with the total number of parties being N . Let $\mathcal{C}$ be the challenger of this secret sharing scheme and $\mathcal{D}$ be the distinguisher who asks leakage queries according to the joint adaptive model ($\mathsf{JALR}^{\mathcal{A},\tau}$) as follows.

1. $\mathcal{D}$ chooses two messages m_0, m_1 and sends it to $\mathcal{C}$, who chooses a bit b uniformly and generates $(share_1, \cdots, share_N) \leftarrow \mathsf{LRShare}(m_b)$
2. For $i = 1$ to q [12]:
   (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set T_i (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi)$[12] and a function $f_i : (\{0,1\}^l)^{|T_i|} \to \{0,1\}^\tau$.
   (b) $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{C}$.
3. For $i = 1$ to q':
   (a) $\mathcal{D}$ adaptively queries $\mathcal{C}$ on a set $T'_i \subset [N]$ where T'_i is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.
   (b) $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{C}$.
4. $\mathcal{D}$ outputs a guess b' for b.



[12] q, q' are some parameters of $\mathcal{D}$'s choice
equivalent to requiring: a) $\cup_{j \leq i} T_j$ is an unauthorized set, and b) $T_1, \cdots, T_i$ are all pair-wise disjoint.

$(\mathsf{LRShare}, \mathsf{LRRec})$ is ϵ_l- leakage resilient with respect to the family $\mathsf{JALR}^{\mathcal{A},\tau}$ if the advantage of any (possibly computationally unbounded) $\mathcal{D}$ is at most $\frac{1}{2} + \epsilon_l$.

3.2 Leakage-resilient Non-malleable Secret Sharing

We will now formally define the tampering model with adaptive leakage ($\mathcal{F}_N^{t-4,\tau}$), that we consider for our LRNMSS scheme.
Let $\mathsf{Share} : \mathcal{M} \to (\{0,1\}^l)^N$ be a sharing function which takes a secret and outputs N shares to be $share_1, \cdots, share_N$. The leakage model we consider is a special case of $\mathsf{JALR}^{\mathcal{A},\tau}$ for t-threshold access structure $\mathcal{A}$, where you do not allow full share queries[13] and only allow the leakage queries (with leakage threshold τ as in $\mathsf{JALR}^{\mathcal{A},\tau}$) on at most $t-4$ shares. We denote this family as $\mathcal{F}^{leak,\tau}$. More specifically $\mathcal{F}^{leak,\tau}$ consists of $(G, \mathcal{L})$ satisfying the following conditions:

- $\mathcal{L}$ is the set of indices of shares on which $\mathsf{JALR}^{\mathcal{A},\tau}$ leakage queries were made.
- G is a function acting on $\{share_i\}_{i \in \mathcal{L}}$ and follows the leakage model of $\mathsf{JALR}^{\mathcal{A},\tau}$ with the added restriction that $|\mathcal{L}| \leq t-4$.

Note that, while $\mathsf{JALR}^{\mathcal{A},\tau}$ allowed leakage queries at most $t-1$ shares for the threshold access structure, $\mathcal{F}^{leak,\tau}$ only allows it for $t-4$ shares. The threshold τ for leakage is exactly what $\mathsf{JALR}^{\mathcal{A},\tau}$ allows.
The leakage resilient tampering family allows the adversary to get a joint adaptive leakage on the shares as in $\mathcal{F}^{leak,\tau}$ and then specify the reconstruction set T along with independent tampering functions $f_1, \cdots, f_N$. We require a restriction that the reconstruction set T shares no index with the set of indices on which leakage queries were made. Formally, we define the leakage resilient tampering family $\mathcal{F}_N^{t-4,\tau}$ as the set of functions $(G, \mathcal{L}, f_1, \cdots, f_N, I)$[14] satisfying the following conditions:

- $(G, \mathcal{L}) \in \mathcal{F}^{leak,\tau}$.
- Let $\mathsf{Leak} := G(\{share_i\}_{i \in \mathcal{L}})$
- For each $i \in [N]$, f_i is a function taking input $share_i$ and Leak and outputs the tampered share $\widetilde{share_i}$.
- I is a function taking input Leak and outputs the reconstruction set T such that $|T| = t$ and $\mathcal{L} \cap T = \phi$.

We now define leakage resilient non-malleable secret sharing with respect to the family $\mathcal{F}_N^{t-4,\tau}$ defined above, for the threshold access structure[15].

[13] Here, we cannot consider full share queries because the tampering functions, which depend on the leakage, will no longer remain independent then.

[14] While in regular tampering family, we only consider the tampering functions acting on the shares, here we also consider the leakage function and the index function which adaptively chooses the reconstruction set dependent on the leakage.

[15] This definition can be thought of as a special adaptation of the general definition [GK18] of non-malleable secret sharing against a tampering family $\mathcal{F}$

Definition 5 (Leakage Resilient Threshold Non-Malleable Secret Sharing)**.** *Let* $(\mathsf{Share}, \mathsf{Rec})$ *be any* (t, N, ϵ_s)*-threshold secret sharing scheme for message space* $\mathcal{M}$*. Let* $\mathcal{F}_N^{t-4,\tau}$ *be the family of tampering functions described above. For each* $(G, \mathcal{L}, f_1, \cdots, f_N, I) \in \mathcal{F}_N^{t-4,\tau}$*,* $m \in \mathcal{M}$ *define the tampering experiment*

$$\mathsf{STamper}_m^{G,\mathcal{L},f_1,\cdots,f_N,I} = \left\{ \begin{array}{l} (share_1, \cdots, share_N) \leftarrow \mathsf{Share}(m) \\ \mathsf{Leak} = G(\{share_i\}_{i \in \mathcal{L}}) \\ T = I(\mathsf{Leak}) \\ \forall i \in [N], \widetilde{share_i} = f_i(share_i, \mathsf{Leak}) \\ \tilde{m} = \mathsf{Rec}(\{\widetilde{share_i}\}_{i \in T}) \\ Output : \mathsf{Leak}, \tilde{m} \end{array} \right\}$$

We say that the (t, N, ϵ_s)*-threshold secret sharing scheme,* $(\mathsf{Share}, \mathsf{Rec})$*, is* ϵ_{nm}*-leakage resilient non-malleable w.r.t to family* $\mathcal{F}_N^{t-4,\tau}$ *if for each* $(G, \mathcal{L}, f_1, \cdots, f_N, I) \in \mathcal{F}_N^{t-4,\tau}$ *there exists a distribution* $\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}$ *over* $\mathcal{M} \cup \{same^*, \perp\}$ *such that, for all* m*,* $\mathsf{STamper}_m^{G,\mathcal{L},f_1,\cdots,f_N,I} \approx_{\epsilon_{nm}} Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}, m)$*, where*

$$Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}, m) = \begin{cases} (\mathsf{Leak}, \tilde{m}) & \leftarrow \mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I} \\ Output: & (\mathsf{Leak}, m) \text{ } if \text{ } \tilde{m} = same^* \\ & (\mathsf{Leak}, \tilde{m}) \text{ } otherwise \end{cases}$$

Further, the distribution $\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}$ *should be efficiently samplable given oracle access to functions* $G, \mathcal{L}, f_1, \cdots, f_N, I$*.*

4 Our Leakage Resilient Secret Sharing Scheme

4.1 Construction

We now present our compiler which uses adaptive extractors and compiles any $(\mathcal{A}, N, \epsilon_s)$-secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ with rate R into a leakage resilient scheme $(\mathsf{LRShare}, \mathsf{LRRec})$, with respect to $\mathsf{JALR}^{\mathcal{A},\tau}$, for the same access structure, with rate $\Theta(R)$. We formally give the construction below.

- Let Ext be an $(n, t_{ext}, d, l, \epsilon)$ average case extractor.
- Let $(\mathsf{Share}, \mathsf{Rec})$ and $(\mathsf{Share}', \mathsf{Rec}')$ be $(\mathcal{A}, N, \epsilon_s)$ and $(\mathcal{A}, N, \epsilon'_s)$-secret sharing schemes for messages in $\{0,1\}^{l_{msg}}$ and $\{0,1\}^d$ respectively.

```
$\mathsf{LRShare}(m)$
  • $(sh_1, ..., sh_N) \leftarrow \mathsf{Share}(m)$
  • $s \in_R \{0,1\}^d$
  • $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s)$
  • For $i \in [N]$
      – $w_i \in_R \{0,1\}^n$
      – $y_i = sh_i \oplus \mathsf{Ext}(w_i; s)$
  • $\forall i \in [N]$,
    Output $share_i$ as $(w_i, y_i, s_i)$
```

```
$\mathsf{LRRec}(share_{\mathcal{X}})$ where $\mathcal{X} \in \mathcal{A}$
  • For $j \in \mathcal{X}$, parse $share_j$ as $(w_j, y_j, s_j)$
  • $s = \mathsf{Rec}'(s_{\mathcal{X}})$
  • For $j \in \mathcal{X}$, $sh_j = y_j \oplus \mathsf{Ext}(w_j; s)$
  • Output $\mathsf{Rec}(sh_{\mathcal{X}})$
```

Theorem 2. *For any $(\mathcal{A}, N, \epsilon_s)-$secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ with rate R for message space $\{0,1\}^{l_{msg}}$ and an $(n, t_{ext}, d, l, \epsilon)$- average case extractor Ext (with $l = l_{msg}/R$), there exists a leakage resilient secret sharing scheme $(\mathsf{LRShare}, \mathsf{LRRec})$ for $\mathcal{A}$, that is leakage resilient with respect to the family $\mathsf{JALR}^{\mathcal{A},\tau}$ with share length $n + d + l$, rate $\Theta(R)$, number of bits that can be leaked per joint query $\tau = n - t_{ext}$ and error $2p(2\epsilon_s + 2\epsilon'_s + \epsilon \cdot 2^{pl})$(where p is the size of largest unauthorized set with respect to $\mathcal{A}$).*

Furthermore, whenever $(\mathsf{Share}, \mathsf{Rec})$ is a constant rate secret sharing scheme with error $2^{-\Omega(l_{msg})}$ and Ext is as in Lemma 2 (which instantiates adaptive extractors), $(\mathsf{LRShare}, \mathsf{LRRec})$ achieves constant rate with leakage and privacy error $2^{-\Omega(l_{msg})}$ and $\tau = \Theta(l)$.

Proof. Correctness follows easily. Privacy requirement of the scheme is subsumed by the leakage-resilience requirement with respect to the family $\mathsf{JALR}^{\mathcal{A},\tau}$. Hence, it suffices to prove leakage-resilience.

Let $\mathcal{D}$ be an arbitrary distinguisher in the game in Section 3.1 and LEAK^{m_0} denote $\mathcal{D}$'s view in the game when the message chosen by $\mathcal{C}$ is m_0. We will now describe the distribution LEAK^{m_0} and also describe a sequence of statistically close hybrids ($\mathsf{Hybrid}_k^{m_0}$) to show statistical closeness of LEAK^{m_0} to LEAK^{m_1}, thus proving the leakage resilience of $(\mathsf{LRShare}, \mathsf{LRRec})$ w.r.t the family $\mathsf{JALR}^{\mathcal{A},\tau}$.

LEAK^{m_0}

1. $(sh_1, ..., sh_N) \leftarrow \mathsf{Share}(m_0)$
2. $s \in_R \{0,1\}^d$
3. $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s)$
4. For $i \in [N]$:
   - $w_i \in_R \{0,1\}^n$
   - $y_i = sh_i \oplus \mathsf{Ext}(w_i; s)$
   - set $share_i$ as (w_i, y_i, s_i)
5. For $i = 1$ to q:
   (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set T_i (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi$) and a function $f_i : (\{0,1\}^l)^{|T_i|} \to \{0,1\}^\tau$.
   (b) $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{C}$.
6. For $i = 1$ to q':
   (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set T'_i where T'_i is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.
   (b) $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{C}$.

Claim 1. *If* $(\mathsf{Share}', \mathsf{Rec}')$ *is an* $(\mathcal{A}, N, \epsilon_s)$*- secret sharing scheme, then by its adaptive privacy,* $\mathsf{LEAK}^{m_0} \approx_{2p\epsilon'_s} \mathsf{Hybrid}_0^{m_0}$.

Proof. Observe that the only difference between these distributions is the way in which $s_1, \cdots, s_N$ are generated. We will now prove the claimed statistical closeness through a reduction to the game of adaptive privacy $\mathsf{Game}_{\mathsf{Ad-Privacy}}$ for the secret sharing scheme $(\mathsf{Share}', \mathsf{Rec}')$. Let $\mathcal{E}_{priv}$ be the challenger in $\mathsf{Game}_{\mathsf{Ad-Privacy}}$ and $\mathcal{R}$ be the reduction. The reduction is as follows:

1. $\mathcal{R}$ samples $(sh_1, ..., sh_N) \leftarrow \mathsf{Share}(m)$.
2. $\mathcal{R}$ chooses $s \in_R \{0,1\}^d$ and sends $s, 0^d$ as messages to $\mathcal{E}_{priv}$.
3. $\mathcal{E}_{priv}$ chooses $b \in_R \{0,1\}$ and samples $(s^0_1, ..s^0_N) \leftarrow \mathsf{Share}'(0^d)$, $(s^1_1, ..s^1_N) \leftarrow \mathsf{Share}'(s)$.
4. For $i = 1$ to q:
   (a) $\mathcal{D}$ queries $\mathcal{R}$ on a set T_i (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi$) and a function $f_i : (\{0,1\}^l)^{|T_i|} \to \{0,1\}^\tau$.
   (b) $\mathcal{R}$ sends T_i to $\mathcal{E}_{priv}$ and receives $s^b_{T_i}$.
   (c) for each $j \in T_i$:
      - $\mathcal{R}$ chooses $w_j \in_R \{0,1\}^n$ and sets $y_j = sh_j \oplus \mathsf{Ext}(w_i; s)$ and $share_j = (w_j, y_j, s^b_j)$.
   (d) $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{R}$.
5. Let $\mathcal{L} = \cup_{i \in [q]} T_i$

6. $\mathcal{R}$ re-samples $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s|s^b_{\mathcal{L}}$ on $\mathcal{L})$.

7. For $i \in [N]/\mathcal{L}$:

   - $\mathcal{R}$ chooses $w_i \in_R \{0,1\}^n$ and sets $y_i = sh_i \oplus \mathsf{Ext}(w_i; s)$ and $share_i$ as (w_i, y_i, s_i).

8. For $i = 1$ to q':

   (a) $\mathcal{D}$ queries $\mathcal{R}$ on a set T'_i where T'_i is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.

   (b) $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{R}$.

9. $\mathcal{R}$ sends s to $\mathcal{D}$.

10. $\mathcal{R}$ receives a guess b' from $\mathcal{D}$ and forwards it to $\mathcal{E}_{priv}$.

Observe that, if $(b = 0)$, the view of $\mathcal{D}$ in this reduction is identical to $\mathcal{D}$'s view according to $\mathsf{Hybrid}^{m_0}_0$. Similarly, if $b = 1$ the view is identical to $\mathcal{D}$'s view as per LEAK^{m_0} (this follows from the observation that by Remark A.5, $(s_1, .., s_N) \leftarrow \mathsf{Share}'(s)$ is identical to $(s_1, .., s_N) \leftarrow \mathsf{Share}'(s|s'_{\mathcal{L}}$ on $\mathcal{L})$ where $(s'_1, .., s'_N) \leftarrow \mathsf{Share}'(s)$).

Since $(\mathsf{Share}', \mathsf{Rec}')$ is $2p\epsilon'_s$-adaptive private by Lemma 4, the advantage of $\mathcal{R}$ and thus $\mathcal{D}$ is atmost $2p\epsilon'_s$, which proves the claim.

□

```
$\mathsf{Hybrid}_k^{m_0}$ where $k \in \{0, 1, .., q\}$

  1. $(s_1^0, ..s_N^0) \leftarrow \mathsf{Share}'(0^d)$.
  2. For $i = 1$ to $k$:
      (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set $T_i$ (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi$) and a function $f_i : (\{0,1\}^l)^{|T_i|} \rightarrow \{0,1\}^\tau$.
      (b) for each $j \in T_i$:
          • Choose $w_j \in_R \{0,1\}^n$, $y_j \in_R \{0,1\}^l$.
          • Set $s_j = s_j^0$ and $share_j = (w_j, y_j, s_j)$.
      (c) $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{C}$.
  3. $(sh_1, ..., sh_N) \leftarrow \mathsf{Share}(m)$.
  4. $s \in_R \{0,1\}^d$.
  5. For $i = k+1$ to $q$:
      (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set $T_i$ (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi$) and a function $f_i : (\{0,1\}^l)^{|T_i|} \rightarrow \{0,1\}^\tau$.
      (b) for each $j \in T_i$:
          • Choose $w_j \in_R \{0,1\}^n$.
          • Set $s_j = s_j^0$, $y_j = sh_j \oplus \mathsf{Ext}(w_i; s)$ and $share_j = (w_j, y_j, s_j)$.
      (c) $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{C}$.
  6. Let $\mathcal{L} = \cup_{i \in [q]} T_i$.
  7. $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s|s_{\mathcal{L}}^0$ on $\mathcal{L})$.
  8. For $i \in [N]/\mathcal{L}$:
      • $w_i \in_R \{0,1\}^n$.
      • $y_i = sh_i \oplus \mathsf{Ext}(w_i; s)$.
      • Set $share_i$ as $(w_i, y_i, s_i)$.
  9. For $i = 1$ to $q'$:
      (a) $\mathcal{D}$ queries $\mathcal{C}$ on a set $T'_i$ where $T'_i$ is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.
      (b) $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{C}$.
```

Claim 2. *If* Ext *is an* $(n, t_{ext}, d, l, \epsilon)$*-extractor, then for any* $k \in [q]$*,* $\mathsf{Hybrid}_{k-1}^{m_0} \approx_{2^{|T_k|l+2}\cdot|T_k|\epsilon_{ext}} \mathsf{Hybrid}_k^{m_0}$.

Proof. Let $h = |T_k|$. If Ext is an $(n, t_{ext}, d, l, \epsilon)$-extractor, then Ext_h (given in Lemma 3) is an $(hn, (h-1)n + t_{ext}, d, hl, h\epsilon)$- average case extractor.

For a uniform source W by Theorem 1, Ext_h can adaptively leak upto $n - t_{ext}$ bits with adaptivity error being $2^{hl+2} \cdot h\epsilon$. Assume for the sake of contradiction that $\mathcal{D}$, can distinguish the hybrids with advantage greater than $2^{hl+2} \cdot h\epsilon$. We now give a reduction $\mathcal{R}$, that uses $\mathcal{D}$ to break the adaptive security of the extractor Ext_h. Let $\mathcal{E}$ be the adaptive extractor challenger. The reduction does the following:

- $\mathcal{R}$ samples $(s_1^0, ..s_N^0) \leftarrow \mathsf{Share}'(0^d)$ and $(sh_1, ..., sh_N) \leftarrow \mathsf{Share}(m)$.
- For $i = 1$ to $k - 1$:
  1. $\mathcal{D}$ queries $\mathcal{R}$ on a set T_i and a function f_i.
  2. for each $j \in T_i$, $\mathcal{R}$ chooses $w_j \in_R \{0,1\}^n$, $y_j \in_R \{0,1\}^l$ and sets $s_j = s_j^0$ and $share_j = (w_j, y_j, s_j)$.
  3. $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{R}$.
- $\mathcal{D}$ queries $\mathcal{R}$ on a set T_k and a function f_k.
- $\mathcal{R}$ sends T_k to $\mathcal{E}$.
- $\mathcal{E}$ chooses a source $w \in_R \{0,1\}^{hn}$ and a seed $s \in_R \{0,1\}^d$ and a bit b. w will be parsed as $w_{j_1}|| \cdots ||w_{j_h}$[16].
- $\mathcal{E}$ computes and sends $\mathcal{R}$ either $x = (x_{j_1}|| \cdots ||x_{j_h}) = \mathsf{Ext}_h(w; s)$(if $b = 1$) or $x \in_R \{0,1\}^{hl}$ (if $b = 0$).
- $\mathcal{R}$ computes $y_j = sh_j \oplus x_j$, $\forall j \in T_k$.
- $\mathcal{R}$ sends f_k, $s_{T_k}^0, y_{T_k}$ to $\mathcal{E}$ in order to make an adaptive query.
- $\mathcal{R}$ receives $f_k(w_{T_k}, y_{T_k}, s_{T_k}^0), s$[17] from $\mathcal{E}$ and forwards the same to $\mathcal{D}$. At this point $\mathcal{E}$ sends s to $\mathcal{R}$.
- For $i = k + 1$ to q:
  1. $\mathcal{D}$ queries $\mathcal{R}$ on a set T_i (such that $T_{[i]} \notin \mathcal{A}$ and $T_i \cap T_{[i-1]} = \phi$) and a function $f_i : (\{0,1\}^l)^{|T_i|} \rightarrow \{0,1\}^\tau$.
  2. for each $j \in T_i$, $\mathcal{R}$ chooses $w_j \in_R \{0,1\}^n$ and sets $s_j = s_j^0$, $y_j = sh_j \oplus \mathsf{Ext}(w_i; s)$ and $share_j = (w_j, y_j, s_j)$.
  3. $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{R}$.
- Let $\mathcal{L} = \cup_{i \in [q]} T_i$.
- $\mathcal{R}$ re-samples $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s|s_{\mathcal{L}}^0$ on $\mathcal{L})$.
- For $i \in [N]/\mathcal{L}$, $\mathcal{R}$ chooses $w_i \in_R \{0,1\}^n$ and sets $y_i = sh_i \oplus \mathsf{Ext}(W_i; s)$ and $share_i$ as (w_i, y_i, s_i).
- For $i = 1$ to q':

[16] $j_1, .., j_h$ is the ascending order of vertices in T_k

[17] sending s only after receiving the adaptive query to $\mathcal{E}$ is to ensure that the query itself was independent of s

1. $\mathcal{D}$ queries $\mathcal{R}$ on a set T'_i where T'_i is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.
2. $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{R}$.

- $\mathcal{R}$ receives a guess b' from $\mathcal{D}$ and forwards it to $\mathcal{E}$.

If x is the output of an extractor then view of $\mathcal{D}$ is as identically distributed to the view of $\mathcal{D}$ in $\mathsf{Hybrid}^{m_0}_{k-1}$. Similarly, if x is a uniform string then view of $\mathcal{D}$ is as identically distributed to the view of $\mathcal{D}$ in $\mathsf{Hybrid}^{m_0}_{k}$. Since Ext_h is an $2^{hl+2}h\epsilon$-adaptive extractor by Theorem 1, the advantage of $\mathcal{R}$ and thus $\mathcal{D}$ is atmost $2^{hl+2}h\epsilon$, which proves the claim. □

Claim 3. *If* $(\mathsf{Share}, \mathsf{Rec})$ *is an* $(\mathcal{A}, N, \epsilon_s)$*- secret sharing scheme, then by its adaptive privacy, for any two secrets* m_0, m_1*,* $\mathsf{Hybrid}^{m_0}_{q} \approx_{2p\epsilon_s} \mathsf{Hybrid}^{m_1}_{q}$*.*

Proof. The proof of this claim reduces to the security of $\mathsf{Game}_{\mathsf{Ad-Privacy}}$ of $(\mathsf{Share}, \mathsf{Rec})$. Let $\mathcal{R}$ be the reduction and $\mathcal{E}'$ be the challenger in $\mathsf{Game}_{\mathsf{Ad-Privacy}}$. The reduction does the following:

- $\mathcal{R}$ samples $(s^0_1, ..s^0_N) \leftarrow \mathsf{Share}'(0^d)$.
- For $i = 1$ to q:
  1. $\mathcal{D}$ queries $\mathcal{R}$ on a set T_i and a function f_i.
  2. for each $j \in T_i$, $\mathcal{R}$ chooses $w_j \in_R \{0,1\}^n$, $y_j \in_R \{0,1\}^l$ and sets $s_j = s^0_j$ and $share_j = (w_j, y_j, s_j)$.
  3. $\mathcal{D}$ receives $f_i(share_{T_i})$ from $\mathcal{R}$.
- Let $\mathcal{L} = \cup_{i \in [q]} T_i$.
- $\mathcal{R}$ chooses $s \in_R \{0,1\}^d$ and sends s to $\mathcal{D}$.
- $\mathcal{R}$ re-samples $(s_1, ..s_N) \leftarrow \mathsf{Share}'(s | s^0_{\mathcal{L}}$ on $\mathcal{L})$.
- $\mathcal{R}$ sends m_0, m_1 to $\mathcal{E}'$.
- $\mathcal{E}'$ chooses $b \in_R \{0,1\}$ and $(sh_1, .., sh_N) \leftarrow \mathsf{Share}(m_b)$.
- For $i = 1$ to q':
  1. $\mathcal{D}$ queries $\mathcal{R}$ on a set T'_i where T'_i is disjoint with $T_{[q]} \cup T'_{[i-1]}$ and $T'_{[i]} \notin \mathcal{A}$.
  2. $\mathcal{R}$ queries $\mathcal{E}'$ on T'_i and receives $sh_{T'_i}$.
  3. For $j \in T'_i$, $\mathcal{R}$ chooses $w_j \in_R \{0,1\}^n$ and sets $y_j = sh_j \oplus \mathsf{Ext}(w_j; s)$ and $share_i$ as (w_j, y_j, s_j).
  4. $\mathcal{D}$ receives $share_{T'_i}$ from $\mathcal{R}$.
- $\mathcal{R}$ receives a guess b' from $\mathcal{D}$ and forwards it to $\mathcal{E}$.

Observe that whenever $b = 0$ (resp. $b = 1$) the view of $\mathcal{D}$ in this reduction is identically distributed to $\mathsf{Hybrid}^{m_0}_{q}$(resp. $\mathsf{Hybrid}^{m_1}_{q}$). Thus, by Lemma 4, it can be shown that advantage of $\mathcal{R}$, thus advantage of $\mathcal{D}$ in distinguishing between $\mathsf{Hybrid}^{m_0}_{q}$ and $\mathsf{Hybrid}^{m_1}_{q}$ is atmost $2p.\epsilon_s$. □

Combining above claims we get $\mathsf{LEAK}^{m_0} \approx_{2^{pl+2}p\epsilon+2p.\epsilon'_s} \mathsf{Hybrid}_q^{m_0} \approx_{2p.\epsilon_s} \mathsf{Hybrid}_q^{m_1}$. Similarly, we can show $\mathsf{LEAK}^{m_1} \approx_{2^{pl+2}p\epsilon+2p.\epsilon'_s} \mathsf{Hybrid}_q^{m_1}$. Hence, the proof of the theorem follows. □

We can instantiate the above scheme with appropriate underlying primitives to obtain a constant rate leakage resilient secret sharing scheme with error $2^{-\Omega(l_{msg})}$. The details of the instantiation are given in Appendix C.

5 Our Leakage Resilient Non-Malleable Secret Sharing Scheme

5.1 Construction

To build our LRNMSS, we enhance our LRSS from Section 4 to additionally satisfy the property of "conditional independence" (see Definition 9). We then combine this with techniques from the NMSS construction of [GK18] (with some modification) to obtain our LRNMSS.

We describe the construction formally in the figure below. It uses a 2-split-state non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ (as defined in Section D.2) and our LRSS scheme $(\mathsf{LRShare}, \mathsf{LRRec})$ (Section 4.1) specifically for t and $t-1$-threshold access structures. The detailed parameters of these building blocks are given in Section D.3. Informally, to secret share a secret m, we first non-malleably encode it to a 2-split-state code (L, R). Then we secret share L using a t-out-of-N LRSS, $(\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)})$, to get the shares $(\mathsf{L}_1, \cdots, \mathsf{L}_N)$. Similarly, we secret share R using the second $(t-1)$-out-of-N LRSS, $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$, to get the shares $(\mathsf{R}_1, \cdots, \mathsf{R}_N)$. The i-th share Sh_i is then set to be $\mathsf{L}_i, \mathsf{R}_i$. The reconstruction procedure, given any t shares just uses the reconstruction algorithms $\mathsf{LRRec}^1_{(t,N)}$ to get L and $\mathsf{LRRec}^2_{(t-1,N)}$ to get R. Finally, it decodes (L, R) to get m.

$\mathsf{Share}(m)$: The N shares of the secret m are generated as follows:

1. $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$.
2. We further secret share L and R as:

$$\begin{aligned}(\mathsf{L}_1, \cdots, \mathsf{L}_N) &\leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L}) \\ (\mathsf{R}_1, \cdots, \mathsf{R}_N) &\leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R})\end{aligned}$$

3. For each $i \in [N]$, set $\mathsf{Sh}_i = (\mathsf{L}_i, \mathsf{R}_i)$.
4. Output the shares $(\mathsf{Sh}_1, \cdots, \mathsf{Sh}_N)$

$\mathsf{Rec}((\mathsf{Share}(m))_T)$: From an authorized set $T = \{i_1, \cdots, i_t\}$, to recover m do:

1. For each $j \in T$, parse Sh_j as $(\mathsf{L}_j, \mathsf{R}_j)$.
2. Recover L and R as:

$$\begin{aligned}\mathsf{L} &:= \mathsf{LRRec}^1_{(t,N)}(\mathsf{L}_{i_1}, \cdots, \mathsf{L}_{i_t}) \\ \mathsf{R} &:= \mathsf{LRRec}^2_{(t-1,N)}(\mathsf{R}_{i_1}, \cdots, \mathsf{R}_{i_{t-1}})\end{aligned}$$

3. Output $m := \mathsf{Dec}(\mathsf{L}, \mathsf{R})$

Figure 1: Construction of t-out-of-n Leakage Resilient Non-Malleable Secret Sharing Scheme

Theorem 3. *For any $N \in \mathbb{N}$ and threshold t, if $(\mathsf{Enc}, \mathsf{Dec})$ is a 2-split-state ϵ_1-non-malleable code (with secret sharing error ϵ_2), $(\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)})$ and $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$ are $(t, N, \epsilon'_3, \epsilon_3)$ and $(t-1, N, \epsilon_4, \epsilon'_4)$ LRSS schemes against JALR^{t,τ_1} and $\mathsf{JALR}^{t-1,\tau_2}$ respectively, then the construction given in Figure 1 is a $(t, N, 2\epsilon'_3 + \epsilon_2)$-secret sharing scheme, which is $(\epsilon_1 + \epsilon_3 + \epsilon_4)$-non-malleable against the leakage resilient tampering family $\mathcal{F}^{t-4,\tau}_N$.*

Overview of the proof of Theorem 3. The proof involves showing that the above scheme satisfies correctness (which is straightforward), statistical privacy and leakage resilient non-malleability against $\mathcal{F}^{t-4,\tau}_N$. Statistical privacy of the scheme relies on the statistical privacy of $\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)}$ and the secret sharing property of the non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$. The leakage resilient non-malleability uses the adaptive leakage resilience of $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$ to remove the dependence of the tampering of L on R and the adaptive leakage resilience of $(\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)})$ to remove the dependence of the tampering of R on L. This also makes the leakage queries of the adversary simulatable without the message. Finally, we use the non-malleability of the underlying NMC to get the final simulator. While this captures the high level idea of our proof, the detailed analysis requires an additional stronger property of "conditional independence" from the LRSS. Further, we see that we can instantiate our scheme with appropriate building blocks to get a constant rate scheme. The details of the proof and rate analysis are given in Appendix D.

6 Leakage Resilient and Non-malleable Secure Message Transmission

The problem of perfectly secure message transmission (SMT) was introduced in [DDWY93], where the goal is the following: the sender S needs to transmit a message m to a receiver R, where S and R are connected by some N number of wires, such that perfect secrecy is guaranteed even in the presence of an adversary which can see a bounded number of wires and perfect resiliency is guaranteed (i.e., receiver receives the correct m), even in the presence of an adversary controlling a bounded number of wires completely. The notion of non-malleable secure message transmission was introduced in [GK18], where the goal is to guarantee that the receiver either receives the original message m or m is destroyed and R gets an "unrelated" message, when an adversary is allowed to tamper with the N wires (according to a certain tampering model). Further, they build this non-malleable secure transmission using a non-malleable secret sharing scheme. However, neither the original perfect SMT [DDWY93, SNR04, WD08, KS09, KKVS18] nor the non-malleable SMT [GK18] support a model allowing leakage on the wires. We give two models of SMT: a leakage resilient SMT and a leakage resilient non-malleable SMT. Further, we show how to get these variants using our LRSS and LRNMSS with good communication ($O(|m|)$ per wire, for message m being transmitted). We formally describe these models and their constructions below.

6.1 Leakage Resilient Message Transmission

We begin by describing the communication model. The sender S and receiver R are connected by N wires and the sender S transmits some message $m \in \mathcal{M}$ to R through these wires. We use $\pi(m, S, R)$ to denote the whole protocol execution (to transmit message m) between the sender S and receiver R. For leakage resilience, we consider an eavesdropping adversary $\mathcal{A}$, who can not only see a bounded number of wires completely, but also get a leakage on additional wires. Then, leakage resilience guarantees that the view of the adversary, denoted by $\pi_{\mathcal{A}}(m, S, R)$ is independent of m. We formalize this notion of leakage resilience below. We begin by defining a secure message transmission protocol (against an eavesdropping adversary) and then define the leakage resilient variant of it.

Definition 6 (Secure Message Transmission)**.** *Let S and R denote the sender and receiver of the message transmission protocol, respectively and $\mathcal{M}$ be the message space from which S wants to transmit a message m to R. S and R are connected by N wires. Let the messages sent through these wires be denoted by $m_1, \cdots, m_N$, during an execution of the protocol $\pi(m, S, R)$ for transmitting the message m and let $t \in [N]$. We say that the protocol $\pi(., S, R)$ is a (t, N, ϵ_s)-secure message transmission protocol if it satisfies the following properties.*

1. ***Correctness**: For every message $m \in \mathcal{M}$, at the end of an honest execution of the protocol execution $\pi(m, S, R)$, where the sender S is transmitting the message m, the receiver R receives m with probability* 1.

2. ***Statistical Privacy**: For every adversary $\mathcal{A}$ that can see the messages sent through at most $t-1$ of the wires between S and R and for each pair of messages $m, m' \in \mathcal{M}$,*

$$\mathbf{SD}\left(\pi_{\mathcal{A}}^{view}(m, S, R), \pi_{\mathcal{A}}^{view}(m', S, R)\right) \leq \epsilon_s,$$

where $\pi_{\mathcal{A}}^{view}(m, S, R)$ denotes the distribution corresponding to the view of $\mathcal{A}$ in the execution of the protocol $\pi(m, S, R)$, which includes the messages sent through at most $t-1$ wires between S and R.

Further, communication cost of the message transmission protocol is the total number of bits that the sender S sends per wire.

We now define a leakage resilient message transmission protocol with respect to some leakage family $\mathcal{F}$, which captures all the information that the adversary gets.

Definition 7 (Leakage Resilient Message Transmission)**.** *A (t, N, ϵ_s)-secure message transmission protocol $\pi(., S, R)$ is said to be a $(t, N, \epsilon_s, \epsilon_l)$-leakage resilient message transmission protocol against a leakage family $\mathcal{F}$, if for all functions $f \in \mathcal{F}$ and for any pair of messages $m, m' \in \mathcal{M}$,*

$$\mathbf{SD}\left(f(\pi^{view}(m, S, R)), f(\pi^{view}(m', S, R))\right) \leq \epsilon_l,$$

where $\pi^{view}(m, S, R)$ denotes the complete view (i.e., all messages sent) in the execution $\pi(m, S, R)$, of the protocol. Hence, $f(\pi^{view}(m, S, R))$ represents the complete view of the adversary, with respect to the leakage model allowed by $\mathcal{F}$.

We now describe our leakage model.

Joint and Adaptive Leakage Model. We allow the adversary $\mathcal{A}_{leak}$ to first, get an arbitrary bounded leakage from at most $t-1$ wires, jointly and adaptively and then see the messages sent through $t-1$ fresh wires (on which leakage queries were not made) in clear, exactly like our LRSS leakage model (section 3.1), $\mathsf{JALR}^{t,\tau}$. Clearly, this model is stronger than the standard statistical privacy in definition 6. We denote this leakage family by $\mathcal{F}_{t,\tau}^{leak}$. Formally, this model is defined by taking the joint and adaptive leakage model $\mathsf{JALR}^{t,\tau}$ of our LRSS scheme, specifically for the t-threshold access structure, and replacing the role of the shares $share_1, \cdots, share_N$ in the queries in $\mathsf{JALR}^{t,\tau}$ with the messages $\pi^{view}(m, S, R) = m_1, \cdots, m_N$, composing the complete view of the protocol $\pi(m, S, R)$.
We now give a construction of a leakage resilient message transmission protocol against the joint adaptive leakage model $\mathcal{F}_{t,\tau}^{leak}$.

6.1.1 Construction:

Let $(\mathsf{LRShare}_{(t,N)}, \mathsf{LRRec}_{(t,N)})$ be a $(t, N, \epsilon_s, \epsilon_l)$-LRSS against $\mathsf{JALR}^{t,\tau}$ (from section 4.1). We run the message transmission protocol $\pi(m, S, R)$ as follows: the Sender S with message m, generates the shares $(share_1, \cdots, share_N) \leftarrow \mathsf{LRShare}_{(t,N)}(m)$ and sends $share_i$ through the wire i, for each $i \in [N]$. The receiver R has all shares and can choose any subset $T = \{i_1, \cdots, i_t\} \subseteq [N]$ to get $m \leftarrow \mathsf{LRRec}_{(t,N)}(share_{i_1}, \cdots, share_{i_t})$.

Theorem 4. *Let $N \in \mathbb{N}$, $t \in [N]$ and $\mathcal{M}$ be the message space. If $(\mathsf{LRShare}_{(t,N)}, \mathsf{LRRec}_{(t,N)})$ is a $(t, N, \epsilon_s, \epsilon_l)$-LRSS against $\mathsf{JALR}^{t,\tau}$ (for messages in $\mathcal{M}$) with rate $O(1)$, then the protocol $\pi(., S, R)$ described above is a $(t, N, \epsilon_s, \epsilon_l)$-leakage resilient message transmission protocol against $\mathcal{F}_{t,\tau}^{leak}$ with a communication cost of $O(\log_2(|\mathcal{M}|))$ per wire.*

Proof. **Correctness.** The correctness follows directly from the correctness of the LRSS scheme.

Leakage Resilience. As the privacy is subsumed by leakage resilience, it suffices to prove leakage resilience. Now, observe that for any $f \in \mathcal{F}_{t,\tau}^{leak}$ and any $m \in \mathcal{M}$, $f(\pi^{view}(m, S, R)) \equiv f(share_1, \cdots, share_N)$ (where, $(share_1, \cdots, share_N) \leftarrow \mathsf{LRShare}_{(t,N)}(m)$). Moreover, by the description of the leakage model, $f \in \mathsf{JALR}^{t,\tau}$. Hence, by the leakage resilience of the underlying secret sharing scheme, it directly follows that for any pair of messages $m, m' \in \mathcal{M}$ and for all $f \in \mathcal{F}_{t,\tau}^{leak}$, $\mathbf{SD}\left(f(\pi^{view}(m, S, R)), f(\pi^{view}(m', S, R))\right) \leq \epsilon_l$.

Communication Cost. By theorem 2, we know that if we instantiate our LRSS construction with a rate $O(1)$ t-threshold secret sharing scheme, then the LRSS shares are each of size $O(\log_2(|\mathcal{M}|))$ and hence we get the desired communication for our message transmission protocol. □

6.2 Leakage Resilient Non-malleable Message Transmission

The communication model is exactly as described above in section 6: the sender S and receiver R are connected by N wires and S wishes to transmit some message $m \in \mathcal{M}$ to R. For the non-malleability of the protocol $\pi(m, S, R)$, we consider an active adversary $\mathcal{A}$, who can first get a leakage on some bounded number of wires, which then get destroyed and then $\mathcal{A}$ can tamper the messages sent through the remaining wires to R. Then, non-malleability guarantees that the modified message m' recovered by R is either the actual message m or is completely “unrelated” to and independent of m. We first describe our adversarial model, which gives both leakage resilience and non-malleability and then formalize the notion of non-malleable message transmission (similar to [GK18], but for our model).

Leakage Resilient Tampering Model. We allow the adversary $\mathcal{A}_{tamper}$ to first get an arbitrary bounded leakage from at most $t-4$ wires, jointly and adaptively (i.e., queries can be combined leakage on non-overlapping subsets of wires, of size upto $t-4$, made adaptively). Let $\mathcal{L}$ be the set of all wires on which the leakage queries were made. The messages on the wires in $\mathcal{L}$ are destructed and not delivered to the receiver. Now, $\mathcal{A}_{tamper}$ can tamper the messages sent through the remaining wires arbitrarily, but independent of each other and also mention a subset of size t that the receiver must use to recover the message[18]. Finally, the receiver recovers a modified message m' from the t messages mentioned by the adversary. Formally, we capture this model by $\mathcal{F}_{t,\tau}^{tamper}$, which is defined exactly like the leakage-resilient tampering family $\mathcal{F}_N^{t-4,\tau}$ of our LRNMSS scheme (section 3.2), with the only difference that here, the queries are made (by $\mathcal{A}_{tamper}$) on the messages $\pi^{view}(m, S, R) = (m_1, \cdots, m_N)$, composing of the complete view of the protocol $\pi(m, S, R)$ (instead of the shares $share_1, \cdots, share_N$, of m in the description of $\mathcal{F}_N^{t-4,\tau}$). Hence, $\mathcal{F}_{t,\tau}^{tamper}$ consists of functions of the form $(G, \mathcal{L}, \{f_i\}_{i \in [N] \setminus \mathcal{L}}, I)$, where G is the leakage function (capturing the leakage model described above), $\mathcal{L}$ consists of the total set of wires on which leakage queries were made, I is the function that takes all the leakage responses and outputs the set T ($|T| = t$) of wires which the receiver must use to recover the message and f_i's are the tampering functions used to modify the messages sent through these remaining wires.

We now define leakage-resilient non-malleable message transmission.

[18] We consider a setting where the receiver requires only t messages to recover the message and here, we allow the adversary to even pick that set. Note that $t \in [N]$ and in particular if $t = N$, no leakage can be received in our model (but all can be tampered), as all messages are required by the receiver to recover the message.

Definition 8 (Leakage Resilient Non-malleable Message Transmission)**.** *A (t, N, ϵ_s)-secure message transmission protocol $\pi(., S, R)$ is said to be ϵ_{nm}-leakage resilient non-malleable against the corruption model $\mathcal{F}_{t,\tau}^{tamper}$ (described above) if for each $(G, \mathcal{L}, \{f_i\}_{i\in[N]\setminus\mathcal{L}}, I) \in \mathcal{F}_{t,\tau}^{tamper}$, there exists a distribution $\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}$ over $\mathcal{M} \cup \{same^*, \perp\}$ such that, for all $m \in \mathcal{M}$,*

$$\mathbf{SD}\left(\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}, Copy(\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}, m)\right) \leq \epsilon_{nm},$$

where $\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}$ is defined as

$$\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I} = \left\{\begin{array}{l} (m_1, \cdots, m_N) \leftarrow \pi(m, S, R) \\ \mathsf{Leak} = G(\{m_i\}_{i\in\mathcal{L}}) \\ T = I(\mathsf{Leak}) \\ \forall i \in [N]\setminus\mathcal{L}, \widetilde{m_i} = f_i(m_i, \mathsf{Leak}) \\ \forall i \in \mathcal{L},\ \ set\ \widetilde{m_i} = \perp \\ \tilde{m} \leftarrow R(\{\widetilde{m_i}\}_{i\in T}) \\ Output : \mathsf{Leak}, \tilde{m} \end{array}\right\}$$

and $Copy(\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}, m)$ is defined as

$$Copy(\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}, m) = \begin{cases} (\mathsf{Leak}, \tilde{m}) & \leftarrow \mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I} \\ Output: & (\mathsf{Leak}, m)\ if\ \tilde{m} = same^* \\ & (\mathsf{Leak}, \tilde{m})\ otherwise \end{cases}$$

Further, $\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}$ should be efficiently samplable given oracle access to the functions $G, \mathcal{L}, \{f_i\}_{i\in[N]\setminus\mathcal{L}}, I$.

We now show how to get a leakage resilient non-malleable message transmission protocol.

6.2.1 Construction:

We consider the same construction of the message transmission protocol as for the leakage resilient case (section 6.1.1), with the only difference that we use the $(t, N, \epsilon_s, \epsilon_{nm})$-LRNMSS, $(\mathsf{Share}, \mathsf{Rec})$ against $\mathcal{F}_N^{t-4,\tau}$ (from section 5.1) to generate the shares $(share_1, \cdots, share_N) \leftarrow \mathsf{Share}(m)$ (instead of the LRSS).

Theorem 5. *Let $N \in \mathbb{N}$, $t \in [N]$ and M be the message space. If $(\mathsf{Share}, \mathsf{Rec})$ is a $(t, N, \epsilon_s, \epsilon_{nm})$-LRNMSS against $\mathcal{F}_N^{t-4,\tau}$ (for messages in $\mathcal{M}$) with rate $O(1)$, then the protocol $\pi(., S, R)$ described above is a $(t, N, \epsilon_s, \epsilon_{nm})$-leakage resilient non-malleable message transmission protocol against $\mathcal{F}_{t,\tau}^{tamper}$ with a communication cost of $O(\log_2(|\mathcal{M}|))$ per wire.*

Proof. **Correctness.** The correctness directly follows from the correctness of the LRNMSS scheme.

Statistical Privacy. The statistical privacy directly follows from the statistical privacy of the underlying LRNMSS.

Leakage Resilient Non-malleability. For any $(G, \mathcal{L}, \{f_i\}_{i\in[N]\setminus\mathcal{L}}, I) \in \mathcal{F}_{t,\tau}^{tamper}$ and for any $m \in \mathcal{M}$, clearly $\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i\in[N]\setminus\mathcal{L}},I}$ is identical to the tampering distribution of the underlying LRNMSS (as $(G, \mathcal{L}, \{f_i\}_{i \in [N] \setminus \mathcal{L}}, I) \in \mathcal{F}_N^{t-4,\tau}$). Hence, by the non-malleablity of the LRNMSS, there exists a distribution $\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i \in [N] \setminus \mathcal{L}},I}$ such that for all $m \in \mathcal{M}$, $\mathbf{SD}\left(\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i \in [N] \setminus \mathcal{L}},I}, Copy(\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i \in [N] \setminus \mathcal{L}},I}, m)\right) \leq \epsilon_{nm}$.

Communication Cost. By corollary D.5, if we instantiate our LRNMSS with a rate $O(1)$ t-threshold secret sharing scheme, then the LRNMSS shares are each of size $O(\log_2(|\mathcal{M}|))$ and hence we get the desired communication for our message transmission protocol. □

References

[ACM88] *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, Chicago, Illinois, 2–4 May 1988.

[ADKO15] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 459–468, 2015.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 774–783, 2014.

[ADN$^+$19] Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, João Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 510–539, 2019.

[AO19] Divesh Aggarwal and Maciej Obremski. Inception makes non-malleable codes shorter as well! *IACR Cryptology ePrint Archive*, 2019:399, 2019.

[BDIR18] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 531–561. Springer, 2018.

[BFO$^+$20] Gianluca Brian, Antonio Faonio, Maciej Obremski, Mark Simkin, and Daniele Venturi. Non-malleable secret sharing against bounded joint-tampering attacks in the plain model. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 127–155. Springer, 2020.

[BFV19] Gianluca Brian, Antonio Faonio, and Daniele Venturi. Continuously non-malleable secret sharing for general access structures. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg,* *Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 211–232. Springer, 2019.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In ACM [ACM88], pages 1–10.

[Bla79] G.R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, Monval, NJ, USA, 1979. AFIPS Press.

[BR07] Mihir Bellare and Phillip Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, page 172–184, New York, NY, USA, 2007. Association for Computing Machinery.

[BS19] Saikrishna Badrinarayanan and Akshayaram Srinivasan. Revisiting non-malleable secret sharing. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 593–622, 2019.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In ACM [ACM88], pages 11–19.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Leakage-resilient extractors and secret-sharing against bounded collusion protocols. *IACR Cryptol. ePrint Arch.*, 2020:478, 2020.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In Juan A. Garay and Roberto De Prisco, editors, *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2010.

[DDWY93] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. arXiv:cs/0602007.

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

[FRR$^+$10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 135–156, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[FV19] Antonio Faonio and Daniele Venturi. Non-malleable secret sharing in the computational setting: Adaptive tampering, noisy-leakage resilience, and improved rate. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 448–479, 2019.

[GK18] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 685–698, 2018.

[GUV07] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In *IEEE Conference on Computational Complexity*, pages 96–108, 2007.

[GW16] Venkatesan Guruswami and Mary Wootters. Repairing reed-solomon codes. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 216–226, New York, NY, USA, 2016. ACM.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*. Springer-Verlag, 2003.

[KKVS18] Ravi Kishore, Ashutosh Kumar, Chiranjeevi Vanarasa, and Kannan Srinathan. On the price of proactivizing round-optimal perfectly secret message transmission. *IEEE Trans. Inf. Theory*, 64(2):1404–1422, 2018.

[KMS19] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing against colluding parties. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 636–660. IEEE Computer Society, 2019.

[KMZ20] Ashutosh Kumar, Raghu Meka, and David Zuckerman. Bounded collusion protocols, cylinder-intersection extractors and leakage-resilient secret sharing. *IACR Cryptol. ePrint Arch.*, 2020:473, 2020.

[Koc96] Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology—CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer-Verlag, 18–22 August 1996.

[KS09] Kaoru Kurosawa and Kazuhiro Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. *IEEE Trans. Inf. Theory*, 55(11):5223–5232, 2009.

[LCG$^+$19] Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Non-malleable secret sharing against affine tampering. *CoRR*, abs/1902.06195, 2019.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA,* *August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 517–532. Springer, 2012.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–53, 1996.

[Rot12] Guy N. Rothblum. How to compute under ac0 leakage without secure hardware. In *Proceedings of the 32Nd Annual Cryptology Conference on Advances in Cryptology — CRYPTO 2012 - Volume 7417*, pages 552–569, Berlin, Heidelberg, 2012. Springer-Verlag.

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

[SNR04] K. Srinathan, Arvind Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.

[SV19] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 480–509, Cham, 2019. Springer International Publishing.

[Vad12] Salil Vadhan. *Pseudorandomness.* Foundations and Trends in Theoretical Computer Science. Now Publishers, 2012. Available at `http://people.seas.harvard.edu/~salil/pseudorandomness/`.

[WD08] Yongge Wang and Yvo Desmedt. Perfectly secure message transmission revisited. *IEEE Trans. Inf. Theory*, 54(6):2582–2595, 2008.

[Zim06] Marius Zimand. Exposure-resilient extractors. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 61–72. IEEE Computer Society, 2006.

A Some Definitions and Preliminary Lemmata

A.1 Statistical Distance and Entropy - Definitions and Lemmata

Statistical distance. Let X_1, X_2 be two probability distributions over some set S. Their *statistical distance* is

$$\mathbf{SD}\left(X_1, X_2\right) \stackrel{\text{def}}{=} \max_{T \subseteq S}\{\Pr[X_1 \in T] - \Pr[X_2 \in T]\} = \frac{1}{2}\sum_{s \in S}\left|\Pr_{X_1}[s] - \Pr_{X_2}[s]\right|$$

(they are said to be ε*-close* if $\mathbf{SD}\left(X_1, X_2\right) \leq \varepsilon$ and denoted by $X_1 \approx_\varepsilon X_2$).
For an event E, $\mathbf{SD}_E(A; B)$ denotes $\mathbf{SD}\left(A|E; B|E\right)$.

Entropy. The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) = -\log(\max_w \Pr[W = w])$.
For a joint distribution (W, Z), following [DORS08], we define the *(average) conditional min-entropy* of W given Z as

$$\widetilde{\mathbf{H}}_\infty(W \mid Z) = -\log(\mathop{\mathbf{E}}_{e \leftarrow Z}(2^{-\mathbf{H}_\infty(W|Z=z)}))$$

(here the expectation is taken over e for which $\Pr[E = e]$ is nonzero).
For any two random variable W, Z, $(W|Z)$ is said to be an (n, t')-average source if W is over $\{0,1\}^n$ and $\widetilde{\mathbf{H}}_\infty(W|Z) \geq t'$.
We require some basic properties of entropy and statistical distance, which are given by the following lemmata and propositions (proofs are given in the appendix).

Lemma 5. *[DORS08] Let A, B, C be random variables. Then if B has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(A \mid B) \geq \mathbf{H}_\infty(A, B) - \lambda \geq \mathbf{H}_\infty(A) - \lambda$. and, more generally, $\widetilde{\mathbf{H}}_\infty(A \mid B, C) \geq \widetilde{\mathbf{H}}_\infty(A, B \mid C) - \lambda \geq \widetilde{\mathbf{H}}_\infty(A \mid C) - \lambda$.*

For any three random variables A,B and C, $\widetilde{\mathbf{H}}_\infty(A|B) \geq \widetilde{\mathbf{H}}_\infty(A|B, C)$.

Proof. Let A, B, C be random variables over $\mathcal{A}, \mathcal{B}, \mathcal{C}$. Then,

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(A|B) &= -\log(\mathop{\mathbf{E}}_{b \leftarrow B}(2^{-\mathbf{H}_\infty(A|B=b)})) \\ &= -\log\sum_{b \in \mathcal{B}}\max_{a \in \mathcal{A}}\Pr[A = a, B = b] \\ &= -\log\sum_{b \in \mathcal{B}}\max_{a \in \mathcal{A}}\sum_{c \in \mathcal{C}}\Pr[A = a, B = b, C = c]\end{aligned}$$

Similarly,

$$\widetilde{\mathbf{H}}_\infty(A|B, C) = -\log\sum_{b \in \mathcal{B}}\sum_{c \in \mathcal{C}}\max_{a \in \mathcal{A}}\Pr[A = a, B = b, C = c]$$

. The proposition follows from the observation that for any $b \in \mathcal{B}$,

$$\sum_{c \in \mathcal{C}}\max_{a \in \mathcal{A}}\Pr[A = a, B = b, C = c] \geq \max_{a \in \mathcal{A}}\sum_{c \in \mathcal{C}}\Pr[A = a, B = b, C = c]$$

□

Lemma 6. *[Vad12] For any random variables A, B, if $A \approx_\epsilon B$, then for any function f, $f(A) \approx_\epsilon f(B)$.*

Lemma 7. *For any random variables A, B over $\mathcal{A}$, and events E, E' with non-zero probabilities,*

$$\Delta(A \wedge E; B \wedge E') \leq |\Pr[E] - \Pr[E']| + \Pr[E'] \cdot \Delta(A|E; B|E')$$

where,

$$\Delta(A \wedge E; B \wedge E') \overset{def}{=} \frac{1}{2} \sum_{a \in \mathcal{A}} |\Pr[A = a \wedge E] - \Pr[B = a \wedge E']|$$

and

$$\Delta(A|E; B|E') \overset{def}{=} \frac{1}{2} \sum_{a \in \mathcal{A}} |\Pr[A = a|E] - \Pr[B = a|E']|$$

Proof. Let $\mathcal{X} = \{a \in \mathcal{A} : \Pr[A = a \wedge E] > \Pr[B = a \wedge E']\}$, $\mathcal{Y} = \mathcal{A}/\mathcal{X}$ and $\epsilon = |\Pr[E] - \Pr[E']|$.

$$\begin{aligned}
&2\Delta(A \wedge E; B \wedge E') \\
&= \sum_{a \in \mathcal{X}} (\Pr[A = a \wedge E] - \Pr[B = a \wedge E']) + \sum_{a \in \mathcal{Y}} (\Pr[B = a \wedge E'] - \Pr[A = a \wedge E]) \\
&= \sum_{a \in \mathcal{X}} (\Pr[E] \Pr[A = a|E] - \Pr[E'][B = a|E']) + \\
&\qquad\qquad \sum_{a \in \mathcal{Y}} (\Pr[E'] \Pr[B = a|E'] - \Pr[E][A = a|E]) \\
&\leq \sum_{a \in \mathcal{X}} ((\Pr[E'] + \epsilon) \Pr[A = a|E] - \Pr[E'][B = a|E']) + \\
&\qquad\qquad \sum_{a \in \mathcal{Y}} (\Pr[E'] \Pr[B = a|E'] - (\Pr[E'] - \epsilon)[A = a|E]) \\
&= \sum_{a \in \mathcal{A}} \epsilon \cdot \Pr[A = a|E] + \sum_{a \in \mathcal{A}} \Pr[E'] \cdot |\Pr[A = a|E] - \Pr[B = a|E']| \\
&\leq \epsilon + 2\Pr[E']\Delta(A|E; B|E')
\end{aligned}$$

□

Lemma 8. *[ADL14] Let X, Y, X', Y' be random variables such that $\Delta(X, Y; X', Y') \leq \epsilon$ and S be any set such that $\Pr[Y \in S] > 0$ and $\Pr[Y' \in S] > 0$, then*

$$\Delta(X|Y \in S; X'|Y' \in S) \leq \frac{2\epsilon}{\Pr[Y' \in S]}$$

A.2 Proof of Theorem 1

Let $W|Z$ be an $(n, t_{ext} + m)$-average source (assume Z is over $\{0,1\}^{n'}$ for some n'), $f' \in \mathcal{F}_{a,m}$ and (f, g) are functions corresponding to f' in the definition of $\mathcal{F}_{a,m}$. Extraction property trivially holds as Ext is an $(n, t_{ext}, d, l, \epsilon)$-average case extractor. Then, to prove the adaptive leakage property, that is

$$U_d, Z, f'(W, E), E \approx_{2^{a+2}\epsilon} U_d, Z, f'(W, U_l), U_l$$

(where E is the random variable $\mathsf{Ext}(W, U_d)$) it suffices to show that,

$$U_d, Z, f(Z,E), g(W, f(Z,E)), E \approx_{2^{a+2}\epsilon} U_d, Z, f(Z,U_l), g(W, f(Z,U_l)), U_l$$

Let $\mathcal{B} = \{b : \Pr[f(Z,E) = b] > 0\}$ and $\mathcal{A} = \{0,1\}^{d+n'+m+l}$.

$$\begin{aligned}
&\Delta(U_d, Z, f(Z,E), g(W, f(Z,E)), E; U_d, Z, f(Z,U_l), g(W, f(Z,U_l)), U_l) \\
&\le \Pr[f(Z,U_l) \notin \mathcal{B}] + \sum_{b \in \mathcal{B}} \frac{1}{2} \sum_{a \in \mathcal{A}} |\Pr[(U_d, Z, g(W, f(Z,E)), E) = a \wedge f(Z,E) = b] \\
&\qquad\qquad - \Pr[(U_d, Z, g(W, f(Z,U_l)), U_l) = a \wedge f(Z,U_l) = b]| \\
&= \Pr[f(Z,U_l) \notin \mathcal{B}] + \sum_{b \in \mathcal{B}} \Delta((U_d, Z, g(W, f(Z,E), E) \wedge f(Z,E) = b; \\
&\qquad\qquad (U_d, Z, g(W, f(Z,U_l)), U_l) \wedge f(Z,U_l) = b) \\
&\qquad \text{(by notation defined in Lemma 7)} \\
&\le \Pr[f(Z,U_l) \notin \mathcal{B}] + \sum_{b \in \mathcal{B}} (|\Pr[f(Z,E) = b] - \Pr[f(Z,U_l) = b]| + \\
&\qquad \Pr[f(Z,U_l) = b] \cdot \Delta((U_d, Z, g(W, f(Z,E)), E) | f(Z,E) = b; \\
&\qquad\qquad (U_d, Z, g(W, f(Z,U_l)), U_l) | f(Z,U_l) = b)) \\
&\qquad \text{(by Lemma 7 with random variables } A, B \text{ and events } E, E' \text{ being} \\
&\qquad (U_d, Z, g(W, f(Z,E)), E), (U_d, Z, g(W, f(Z,U_l)), U_l), f(Z,E) = b, \\
&\qquad\qquad \text{and } f(Z,U_l) = b \text{ respectively.)} \\
&\le (|\mathcal{B}| + 1)\epsilon + \sum_{b \in \mathcal{B}} \Pr[f(Z,U_l) = b] \cdot \Delta((U_d, Z, g(W, f(Z,E)), E) | f(Z,E) = b; \\
&\qquad\qquad (U_d, Z, g(W, f(Z,U_l)), U_l) | f(Z,U_l) = b) \\
&\qquad \text{(by extractor security } Z, E \approx_\epsilon Z, U_l, \text{ then by Lemma 6, } f(Z,E) \approx_\epsilon \\
&\qquad f(Z,U_l)\text{. Then by the definition of statistical distance, we get} \\
&\qquad |\Pr[f(Z,E) = b] - \Pr[f(Z,U_l) = b]| \le \epsilon \text{ and} \\
&\qquad \Pr[f(Z,U_l) \notin \mathcal{B}] \le \epsilon (\text{as } \Pr[f(Z,E) \in \mathcal{B}] = 1]))
\end{aligned}$$

$$= (|\mathcal{B}|+1)\epsilon + \sum_{b\in\mathcal{B}} \Pr[f(Z,U_l)=b]\cdot \Delta((U_d,Z,g(W,b),E)|f(Z,E)=b;$$

$$(U_d,Z,g(W,b),U_l)|f(Z,U_l)=b)$$

$$= (|\mathcal{B}|+1)\epsilon + \sum_{b\in\mathcal{B}} \Pr[f(Z,U_l)=b]\cdot \frac{2\epsilon}{\Pr[f(Z,U_l)=b]}$$

(by Lemma 8 with $X=(U_d,Z,g(W,b),E), Y=f(Z,E)$,
$X'=(U_d,Z,g(W,b),U_l), Y'=f(Z,U_l)$ and $\varepsilon=\{b\}$,
as $U_d,Z,g(W,b),E \approx_\epsilon U_d,Z,g(W,b),U_l$ and
$U_d,Z,g(W,b),f(Z,E),E \approx_\epsilon U_d,Z,g(W,b),f(Z,U_l),U_l$ follow
from average case extractor security and Lemma 6 respectively)

$$\leq 4|\mathcal{B}|\epsilon \leq 2^{a+2}\epsilon$$

A.3 Proof of Lemma 2

- Let Ext' be an $(n,t_{ext},d,l',\epsilon)$ extractor as in Lemma 1
- Let γ,γ' be the assymptotic constants in $d=\mathcal{O}(\log(\frac{n}{\epsilon}))$ and $l'=(1-\nu)(t_{ext})-\mathcal{O}(\log(n)+\log(\frac{1}{\epsilon}))$ in Lemma 1. Therefore, we have $d\leq\gamma\log(\frac{n}{\epsilon})$ and $l'\geq(1-\nu)t_{ext}-\gamma'\log(\frac{n}{\epsilon})$ for large values.
- Set $\epsilon=2^{-\alpha t_{ext}}$ and $\nu=\frac{1}{2}$
- Let α,β be some parameters such that $\beta<\frac{1}{\gamma}$ and $\alpha<\frac{1}{2(\gamma'+1)}$.
- Let $l=\beta\gamma\log(\frac{n}{\epsilon})$. Note that, $l<l'$ by the setting of α and β.
- Let Ext be an extractor that computes Ext' and outputs only first l bits of the output.
- By definition, Ext is an (n,t_{ext},d,l,ϵ)-extractor.
- Adaptivity error (with respect to $\mathcal{F}_{l,m}$) of Ext from Theorem 1 is $\delta=4\epsilon.2^l\leq 4n\cdot 2^{-\alpha t_{ext}(1-\beta\gamma)}=2^{-\Omega(t_{ext})}$
- Summing up, we have $\epsilon=2^{-\alpha t_{ext}}, l=\beta\gamma(\alpha t_{ext}+\log n), d\leq\gamma(\alpha t_{ext}+\log n), \delta=2^{-\Omega(t_{ext})}$.

A.4 Proof of Lemma 3

Let $W|Z$ be the $(hn,(h-1)n+t_{ext})$- average source, (where W is parsed as $(W_1,\cdots,W_h)$) and $S\equiv U_d$. Then, by Lemma 5, $\widetilde{\mathbf{H}}_\infty(W_h|W_1,\cdots,W_{h-1},Z)\geq t_{ext}$. Therefore, by the security of Ext, we have

$$Z,W_1,\cdots,W_{h-1},\mathsf{Ext}(W_h;S),S\approx_\epsilon Z,W_1,\cdots,W_{h-1},U_l,S$$

Then by Lemma 6 it follows that,

$$Z,W_1,\cdots,W_{h-2},\mathsf{Ext}(W_{h-1};S),\mathsf{Ext}(W_h;S),S\approx_\epsilon$$

$$Z,W_1,\cdots,W_{h-2},\mathsf{Ext}(W_{h-1};S),U_l,S$$

We now aim to show $\mathsf{Ext}(W_{h-1};S)$ is close to uniform even given $Z, W_1, \cdots, W_{h-2}, U_l$ and S. Also, S remains uniform given $Z, W_1, \cdots, W_{h-2}, U_l$. Since U_l is independent of $W_1, \cdots, W_{h-2}, W_{h-1}, Z$, we have $\widetilde{\mathbf{H}}_\infty(W_{h-1}|W_1, \cdots, W_{h-2}, Z, U_l) = \widetilde{\mathbf{H}}_\infty(W_{h-1}|W_1, \cdots, W_{h-2}, Z)$ By Proposition A.1, we have $\widetilde{\mathbf{H}}_\infty(W_{h-1}|W_1, \cdots, W_{h-2}, Z) \geq \widetilde{\mathbf{H}}_\infty(W_{h-1}|W_1, \cdots, W_{h-2}, Z, W_h)$ which is atleast t_{ext}(by Lemma 5). Then by security of Ext, we have

$$Z, W_1, \cdots, W_{h-2}, \mathsf{Ext}(W_{h-1};S), U_l, S \approx_\epsilon Z, W_1, \cdots, W_{h-2}, U_l', U_l, S\text{[19]}$$

Thus, by triangle inequality,

$$Z, W_1, \cdots, W_{h-2}, \mathsf{Ext}(W_{h-1};S), \mathsf{Ext}(W_h;S), S \approx_{2\epsilon} Z, W_1, \cdots, W_{h-2}, U_l, U_l, S$$

Then by similar arguments, it is easy to see that

$$Z, (\mathsf{Ext}(W_1;S), \cdots, \mathsf{Ext}(W_{h-2};S), \mathsf{Ext}(W_{h-1};S), \mathsf{Ext}(W_h;S)), S \approx_{h\epsilon} Z, U_{hl}, S$$

A.5 Consistent Re-sampling

For any $(\mathcal{A}, N, \epsilon_s)$-secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$, for any message m and a subset $\mathcal{L} \subseteq [N]$, when we say "$(sh_1, .., sh_N) \leftarrow \mathsf{Share}(m)$ *consistent with* $sh^*_{\mathcal{L}}$ on $\mathcal{L}$" or "$(sh_1, .., sh_N) \leftarrow \mathsf{Share}(m|sh^*_{\mathcal{L}})$" we mean the following procedure:

- Sample and output $(sh_1, .., sh_N)$ uniformly from the distribution $\mathsf{Share}(m)$ conditioned on the event that $sh_{\mathcal{L}} = sh^*_{\mathcal{L}}$
- If the above event is a zero probability event then output a string of all zeroes (of appropriate length).

We would also use the following property of consistent re-sampling in our proofs[20] . Assume shares of a message m on indices T are a-priori chosen. Now, re-sampling all the other shares of m consistent with shares on T is equivalent to first re-sampling shares of m on some set T'(consistent with fixed shares on T) and then again re-sampling all other shares of m(consistent with shares on T and T'). This is formalized below. For any two sets T, T', the following distributions are identical for an $(\mathcal{A}, N, \epsilon_s)$-secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$. For $i_j \in T$, let $sh^*_{i_j}$ be an arbitrary value. Let m be an arbitrary message. Then the following distributions are identical[21].

• $(sh_1', .., sh_N') \leftarrow \mathsf{Share}(m\|sh^*_T)$ • $(sh_1, .., sh_N) \leftarrow \mathsf{Share}(m\|sh^*_T, sh'_{T'})$ • Output $(sh_1, .., sh_N)$	• $(sh_1, .., sh_N) \leftarrow \mathsf{Share}(m\|sh^*_T)$ • Output $(sh_1, .., sh_N)$

[19] U_l' is a uniform sample from $\{0,1\}^l$ independent of U_l

[20] Note that we only use the re-sampling in proofs and do not require the procedure to be efficient.

[21] The remark follows from the following general observation. If X be any distribution over some $\mathcal{B} \times \mathcal{C}$. Then, sampling $(b, c) \leftarrow X$ is equivalent to sampling $(b', c) \leftarrow X$ and re-sampling $(\hat{b}, c)$ from the conditional distribution $(X|c$ is the second coordinate).

A.6 Proof of Adaptive Privacy: Lemma 4

Proof. To prove the claim, it suffices to show that in $\mathsf{Game}_{\mathsf{Ad-Privacy}}$, the distribution of view of $\mathcal{D}$ when $b = 0$ is statistically close to the distribution of view of $\mathcal{D}$ when $b = 1$. We show this through a sequence of hybrids, whose description is given below.

Let $m_1^*, .., m_p^*$ be some fixed messages in $\mathcal{M}$.

$\mathsf{Hybrid}_\mathsf{k}^\mathsf{m}$ for $k \in \{0, 1, \cdots, q\}, m \in \{m_0, m_1\}$

1. For $j = 1$ to k
   - $\mathcal{D}$ queries with an index i_j(such that $i_{[j]} \notin \mathcal{A}$).
   - $(share_1^j, \cdots, share_N^j) \leftarrow \mathsf{Share}(m_j^*)$
   - $\mathcal{D}$ receives $share_{i_j}^j$
2. $(share_1, \cdots, share_N) \leftarrow \mathsf{Share}(m|\{share_{i_1}^1, \cdots, share_{i_k}^k\}$ on $\{i_1, .., i_k\})$.
3. For $j = k + 1$ to q
   - $\mathcal{D}$ queries with an index i_j(such that $i_{[j]} \notin \mathcal{A}$) and receives $share_{i_j}$

Observe that $\mathsf{Hybrid}_\mathsf{0}^{\mathsf{m_0}}$(resp. $\mathsf{Hybrid}_\mathsf{0}^{\mathsf{m_1}}$) is identical to the view of $\mathcal{D}$ when $b = 0$(resp. $b = 1$) in $\mathsf{Game}_{\mathsf{Ad-Privacy}}$. Also, $\mathsf{Hybrid}_\mathsf{q}^{\mathsf{m_0}}$ is identical to $\mathsf{Hybrid}_\mathsf{q}^{\mathsf{m_1}}$ as both the hybrids are independent of m_0 and m_1. Now we show that for any $k \in [q]$, statistical closeness of $\mathsf{Hybrid}_{\mathsf{k-1}}^{m_0}$ from $\mathsf{Hybrid}_\mathsf{k}^{\mathsf{m_0}}$ is ϵ_s, which would imply that $\mathsf{Hybrid}_\mathsf{0}^{\mathsf{m_0}}$ is $q\epsilon_s(\leq p\epsilon_s)$[22] close to $\mathsf{Hybrid}_\mathsf{q}^{\mathsf{m_0}}$ by triangle inequality, which concludes the proof of the lemma in conjunction with prior observations.

Let $k \in [q]$. We will prove statistical closeness of $\mathsf{Hybrid}_{\mathsf{k-1}}^{m_0}$ and $\mathsf{Hybrid}_\mathsf{k}^{m_0}$ with a reduction to statistical privacy. Let $\mathcal{P}_{ch}$ be the challenger for statistical privacy game, $\mathcal{R}$ be the reduction and $\mathcal{D}$ be a distinguisher for the hybrids. The reduction $\mathcal{R}$ does the following:

1. For $j = 1$ to $k - 1$
   - $\mathcal{D}$ queries $\mathcal{R}$ with an index i_j(such that $i_{[j]} \notin \mathcal{A}$).
   - $(share_1^j, \cdots, share_N^j) \leftarrow \mathsf{Share}(m_j^*)$
   - $\mathcal{D}$ receives $share_{i_j}^j$ from $\mathcal{R}$.
2. $\mathcal{D}$ queries $\mathcal{R}$ with an index i_k(such that $i_{[k]} \notin \mathcal{A}$)
3. $\mathcal{R}$ sends i_k and the two messages m_0 and m_k^* to $\mathcal{P}_{ch}$
4. $\mathcal{P}_{ch}$ chooses $b \in_R \{0, 1\}$ and computes $(share_1^k, \cdots, share_N^k) \leftarrow \mathsf{Share}(m_0)$ if $b = 0$ and $(share_1^k, \cdots, share_N^k) \leftarrow \mathsf{Share}(m_k^*)$ otherwise.
5. $\mathcal{P}_{ch}$ sends $share_{i_k}^k$ to $\mathcal{R}$, who forwards it to $\mathcal{D}$.

[22] In each query $\mathcal{D}$ has to specify a distinct index, while ensuring that union of all these indices is an unauthorized set. Therefore, the maximum number of queries he can make is bounded by p.

6. $\mathcal{R}$ computes $(share_1, \cdots, share_N) \leftarrow \mathsf{Share}(m_0 | \{share^1_{i_1}, \cdots, share^k_{i_k}\}$ on $\{i_1, .., i_k\})$.
7. For $j = k+1$ to q
   - $\mathcal{D}$ queries $\mathcal{R}$ with an index i_j(such that $i_{[j]} \notin \mathcal{A}$) and receives $share_{i_j}$
8. $\mathcal{D}$ sends a guess b' for b to $\mathcal{R}$, who would forward it to $\mathcal{P}_{ch}$.

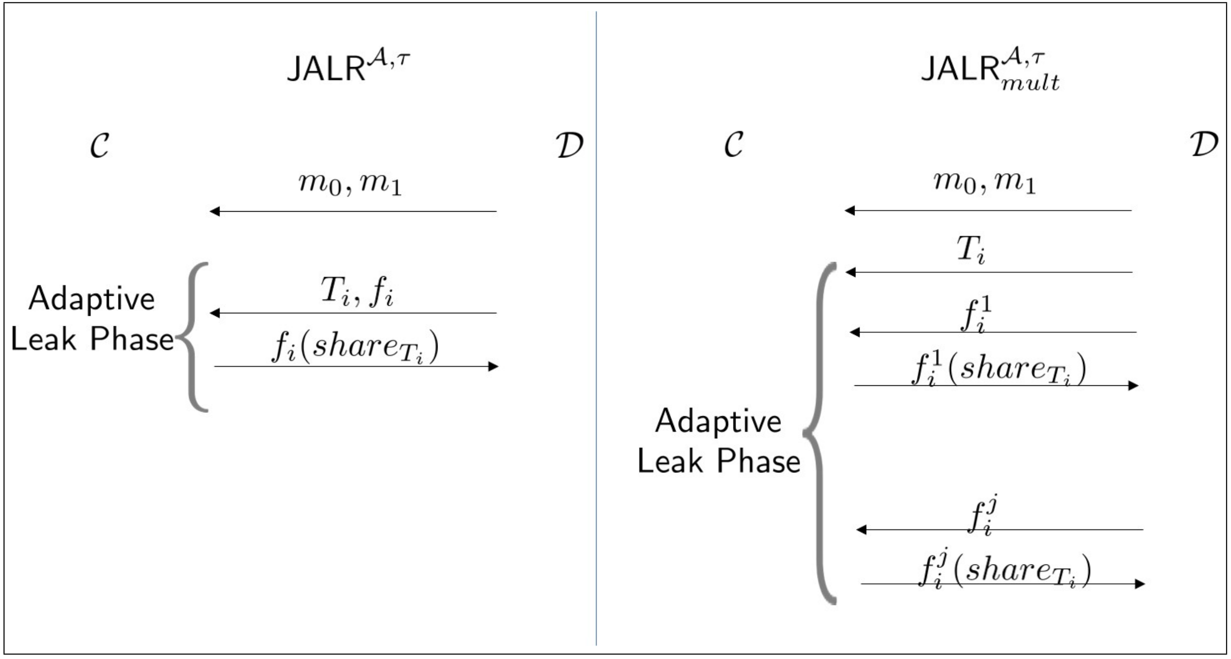
Observe that if $b = 0$, the view of $\mathcal{D}$ in this reduction is identical to view of $\mathcal{D}$ in $\mathsf{Hybrid}^{m_0}_{k-1}$ by Remark A.5(with $T = i_{[k-1]}$, $T' = \{i_k\}$, $m = m_0$, $sh^*_T = \{share^j_{i_j}\}_{j\in[k-1]})$). If $b = 1$, the view of $\mathcal{D}$ in this reduction is identical to view of $\mathcal{D}$ in $\mathsf{Hybrid}^{m_0}_{k}$. The advantage of $\mathcal{R}$ winning the privacy game with $\mathcal{P}_{ch}$ is atmost ϵ_s as $\{i_k\} \notin \mathcal{A}$. Hence the advantage of any D distinguishing the hybrids is atmost ϵ_s.

□

B Equivalent Models and Special Cases of LRSS

B.1 Collapsable Queries

In the description of $\mathsf{JALR}^{\mathcal{A},\tau}$(specifically in step 2(a)) $\mathcal{D}$ could specify a set T_i and a single function f_i. Now consider a variant where(in step 2(a)) $\mathcal{D}$ could specify a set T_i and a function f^1_i. $\mathcal{D}$ observes leakage with respect to f^1_i and can further specify one more leakage function f^2_i and observe leakage on the shares of the same set T_i but with respect to the function f^2_i and this procedure can repeat until querying leakage on T_{i+1}. Let us denote this variant as $\mathsf{JALR}^{\mathcal{A},\tau}_{mult}$[23].
We bring to the notice of readers that $\mathsf{JALR}^{\mathcal{A},\tau}_{mult}$ is equivalent to $\mathsf{JALR}^{\mathcal{A},\tau}$ in the information-theoretic setting.



[23] τ in $\mathsf{JALR}^{\mathcal{A},\tau}_{mult}$ denotes the maximum of total leakage observed on shares in any T_i. Let $f^1_i, ...f^j_i$ be leakage functions asked on a set T_i. Then $\tau \geq$ sum of output lengths of the functions $f^1_i, ...f^j_i$.

We mean multiple consecutive leakage queries on a set T_i can be "collapsed" into a single leakage query on set T_i. This follows because $\mathcal{D}$'s strategy to come up with $f_i^1, ...f_i^j$ can be encoded into a single function f_i. For completeness, we just show how to collapse two leakage functions f_i^1 and f_i^2(whose description depended on leakage observed with respect to f_i^1) into one query f_i: $f_i(share_{T_i})$:

- Description of f_i is parsed as $f_i^1 || f_i^a$
- $lk = f_i^1(share_{T_i})$
- $f_i^2 = f_i^a(lk)$ (output of f_i^a is parsed as description of a function f_i^2)
- $lk' = f_i^2(share_{T_i})$
- Output lk, lk'

B.2 Independent and Adaptive Leakage

Another interesting subclass of the family $\mathsf{JALR}^{\mathcal{A},\tau}$ is the family wherein each leakage query leakage is asked only on a single share and thus can obtain τ bits of leakage per every share (in an unauthorized set) while having the privilege of adaptivity and further can query full shares.

C Instantiation of the LRSS scheme

Let $(\mathsf{Share}, \mathsf{Rec})$ and $(\mathsf{Share}', \mathsf{Rec}')$ be R-rate secret sharing schemes, then by instantiating Ext with the extractor in Lemma 2, we have

- $l = \frac{l_{msg}}{R}$
- $l = \Theta(t_{ext}) = \Theta(d)$
- For each joint leakage query we can support upto $\tau \leq n - t_{ext}$ bits of leakage, as all w_i's were uniform to begin with(hence have entropy n).
- Setting $\tau = \Theta(n)$, gives $t_{ext} = \Theta(n)$
- Therefore, rate of the scheme is $\frac{l_{msg}}{l+d+n} = \frac{l_{msg}}{\Theta(\frac{l_{msg}}{R})} = \Theta(R)$.
  Thus if R is constant, the LRSS too has constant rate.
- Error will be $2p(2\epsilon_s + 2\epsilon_s' + 2^{-\Omega(\frac{l_{msg}}{R})})$

D Leakage Resilient Non-Malleable Secret Sharing for Threshold Access Structures

We begin by looking at a stronger guarantee of conditional independence that we require from the underlying LRSS scheme. We define this property and show that our LRSS scheme satisfies it.

D.1 Conditional Independence

To instantiate the non-malleable secret sharing construction in Section 5.1 with the leakage resilient secret sharing of Section 4.1, we need an additional stronger property from the LRSS scheme defined in Section 3.1, which is called conditional independence, defined as below.

Definition 9. *[BS19] A (t, N, ϵ_l) secret sharing scheme* $(\mathsf{LRShare}, \mathsf{LRRec})$ *for a message space $\mathcal{M}$ is said to be ϵ_l-leakage resilient against the leakage family $\mathsf{JALR}^{t,\tau}$ (for t-threshold access structure) with* conditional independence *if, for any $K, S \subseteq [N]$ such that $|K| = t-1$ and $|K \cap S| = 0$, there exists a function $\mathsf{aux}_{K,S}$ (over appropriate domain) such that the following properties hold:*

- ***Conditional Independence:*** *For any message $m \in \mathcal{M}$, the following two distributions are identical:*

  1. *$(share_1, \cdots, share_N) \leftarrow \mathsf{LRShare}(m; r)$ (for uniformly chosen r).*
  2. *$(share_S, share_{[N]\setminus S})$, which are generated by resampling procedure:*
     - *Sample $(share_1, \cdots, share_N) \leftarrow \mathsf{LRShare}(m; r)$.*
     - *Compute $a \leftarrow \mathsf{aux}_{K,S}(m; r)$.*
     - *Let R' be the set of all r' such that $a = \mathsf{aux}_{K,S}(m; r')$ and $share_K = \mathsf{LRShare}(m; r')_K$.*
     - *Sample $r' \leftarrow R'$ and let $share'_S \leftarrow \mathsf{LRShare}(m; r')_S$*
     - *Output $(share'_S, share_{[N]\setminus S})$ (replacing shares of S with corresponding shares $share'_S$)*

- ***Leakage Resilience (joint and adaptive):*** *For every $G_{\mathcal{L},K} \in \mathsf{JALR}^{t,\tau}$ (following the adaptive and joint leakage model of $\mathsf{JALR}^{t,\tau}$) acting on the total set of leakage query indices $\mathcal{L}$ (excluding the set of indices on which full shares were queried) and making full share queries on K, for every two messages $m_0, m_1 \in \mathcal{M}$,*

$$(\mathsf{aux}_{K,S}(m_0; r), G_{\mathcal{L},K}(\mathsf{LRShare}(m_0; r)_{\mathcal{L}\cup K})) \approx_{\epsilon_l} (\mathsf{aux}_{K,S}(m_1; r), G_{\mathcal{L},K}(\mathsf{LRShare}(m_1; r)_{\mathcal{L}\cup K}))$$

  Here, since we are in the adaptive world, we should mention that the $\mathsf{aux}_{K,S}(m_b; r)$ is given to the leakage adversary after all the leakage and full share queries.

The construction in Section 4.1 satisfies the desired conditional independence property. For completeness, we show this in the following lemma.

Lemma 9. *For any (t, N, ϵ_s)-secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *with rate R for message space $\mathcal{M}$ and an $(n, t_{ext}, d, l, \epsilon_{ext})$-average case extractor* Ext*, there exists a t-threshold leakage resilient secret sharing scheme* $(\mathsf{LRShare}, \mathsf{LRRec})$*, that is leakage resilient with respect to $\mathsf{JALR}^{t,\tau}$ (where $\tau \leq n-1$) with conditional independence. The rate of this scheme is $\Theta(R)$ and error is $2(2\epsilon_s + (t-1)\epsilon_{ext} \cdot 2^{(t-1)l})$.*

Proof. The proof of correctness and privacy follow directly from Theorem 2. We prove conditional independence and leakage resilience, as in Definition 9.

Conditional Independence. Fix sets $K \subseteq [N]$ such that $|K| = t-1$, $S \subseteq [N]\setminus K$ and $T = [N]\setminus(K \cup S)$. Fix some message $m \in \mathcal{M}$.

Define $\mathsf{aux}_{K,S}$ as a function, which, on input m and randomness $rand$, outputs $\mathsf{aux} = s$, where s is

the seed of the extractor (s is part of $rand$).
Now, we fix $share_K, \mathsf{aux}, m$. Then, it is clear that this fixes all the shares $(sh_1, \cdots, sh_N)$ of m (since $|K| = t-1$). The only randomness for sampling $share_i$ for any $i \in [N] \backslash K$ is in sampling w_i, which is independent for each i. Hence, conditioned on fixing $share_K, \mathsf{aux}, m$, the set of shares $share_S$ is independent of $share_T$. Hence, $share_S$ and $share'_S$ are distributed identically for every fixed $(s, sh_1, \cdots, sh_N)$ ($share'_S$ is the resampled distribution from the conditional independence definition 9).
Leakage Resilience. By definition of $\mathsf{aux}_{K,S}$, we wish to prove that for every two messages $m_0, m_1 \in \mathcal{M}$ and for every $G_{\mathcal{L},K} \in \mathsf{JALR}^{t,\tau}$ (acting on leakage query indices $\mathcal{L}$ and full share query indices K), we have

$$(\mathsf{aux}_{K,S}(m_0; r), G_{\mathcal{L},K}(\mathsf{LRShare}(m_0; r)_{\mathcal{L}\cup K})) \\ \approx_{\epsilon_l} (\mathsf{aux}_{K,S}(m_1; r), G_{\mathcal{L},K}(\mathsf{LRShare}(m_1; r)_{\mathcal{L}\cup K}))$$

And we have that $\mathsf{aux}_{K,S}(m; rand) = s$, where s is the seed of the extractor.
The proof of the above claim follows almost exactly from the proof of Theorem 2 with the small observation that all the hybrids in the proof could also output the seed s at the end of all the queries. In all the reduction games (of Claims 1,2 and 3 in the proof specifically), observe that the seed s can always be obtained by the reduction game (in the end) and hence it can complete the simulation, by forwarding the seed at the end. This completes the proof of the lemma. □

Now, we formally define our second building block, a 2-split-state non-malleable code.

D.2 Non-malleable Codes

We use non-malleable codes as a building block in our construction of non-malleable secret sharing. Non-malleable codes are coding schemes which provide a guarantee that, if the codeword is tampered with, then the message recovered is either same as the original message, or is independent of it. Formally, we define non-malleable codes w.r.t a tampering family $\mathcal{F}$ as below

Definition 10. *A coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with message and codeword spaces as* $\{0,1\}^l, \{0,1\}^n$ *respectively, is* ϵ*- non-malleable with respect to a function family* $\mathcal{F} \subseteq \{f : \{0,1\}^n \to \{0,1\}^n\}$ *if* $\forall\, f \in \mathcal{F}$*,* $\exists$ *a distribution* Sim_f *over* $\{0,1\}^l \cup \{same^*, \perp\}$ *such that* $\forall\, m \in \{0,1\}^l$

$$\mathsf{Tamper}_f^m \approx_\epsilon \mathsf{Copy}_{Sim_f}^m$$

where Tamper_f^m *denotes the distribution* $\mathsf{Dec}(f(\mathsf{Enc}(m)))$ *and* $\mathsf{Copy}_{Sim_f}^m$ *is defined as*

$$\tilde{m} \leftarrow Sim_f$$

$$\mathsf{Copy}_{Sim_f}^m = \begin{cases} m \ if \ \tilde{m} = same^* \\ \tilde{m} \ \mathsf{otherwise} \end{cases}$$

Sim_f should be efficiently samplable given oracle access to $f(.)$.

We also require the following secret sharing property of non-malleable codes in the 2-split-state model $\mathcal{F}_2$. It states that a 2-split-state non-malleable code is a 2-out-of-2 secret sharing scheme.

Lemma 10. *[ADKO15] Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{\beta_1} \times \{0,1\}^{\beta_2}$ *and* $\mathsf{Dec} : \{0,1\}^{\beta_1} \times \{0,1\}^{\beta_2} \to \{0,1\}^k$ *be a* ϵ*-non-malleable code in the* 2*-split-state model for some* $\epsilon < 1/2$*. For any pair of messages* $m_0, m_1 \in \{0,1\}^k$*,* $\mathsf{R}^{m_0} \approx_{2\epsilon} \mathsf{R}^{m_1}$*, where* $(\mathsf{L}^{m_0}, \mathsf{R}^{m_0}) \leftarrow \mathsf{Enc}(m_0)$ *and* $(\mathsf{L}^{m_1}, \mathsf{R}^{m_1}) \leftarrow \mathsf{Enc}(m_1)$*.*

The detailed parameters corresponding to the building blocks used in our construction are as given below.

D.3 Building Blocks

The construction uses the following building blocks.

- A 2-split-state ϵ_1-non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ (as defined in Section D.2), where Enc takes messages from $\mathcal{M}$ and outputs (L, R), of lengths β_1, β_2 respectively. Furthermore, $(\mathsf{Enc}, \mathsf{Dec})$ satisfies the secret sharing property that, for any two $m, m' \in \mathcal{M}$, $\mathsf{R} \approx_{\epsilon_2} \mathsf{R}'$, where $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$ and $(\mathsf{L}', \mathsf{R}') \leftarrow \mathsf{Enc}(m')$.
- A $(t, N, \epsilon_3', \epsilon_3)$-leakage resilient secret sharing scheme [24] $(\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)})$, with joint and adaptive leakage model JALR^{t,τ_1} for t-threshold access structure (of Definition 3.1) for message space $\{0,1\}^{\beta_1}$ with conditional independence (as in Definition 9). This means that the adversary can make leakage queries on any $t-1$ shares adaptively and jointly, with leakage threshold τ_1 (as interpreted in JALR^{t,τ_1}) and after making all the leakage queries, the adversary can get upto $t-1$ full shares. Let the size of each share be η_1.
- A $(t-1, N, \epsilon_4', \epsilon_4)$-leakage resilient secret sharing scheme[25] $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$, with joint and adaptive leakage model $\mathsf{JALR}^{t-1,\tau_2}$ for message space $\{0,1\}^{\beta_2}$ with conditional independence. This means that the adversary can make leakage queries on any $t-2$ shares adaptively and jointly, with leakage threshold τ_2 (as interpreted in $\mathsf{JALR}^{t-1,\tau_2}$) and after making all the leakage queries, the adversary can get upto $t-2$ full shares. Let the size of each share be η_2.

Now, we give the security proof of the construction described in Section 5.

D.4 Proof of Theorem 3

D.4.1 Correctness

The correctness of the scheme is straightforward from the correctness of the underlying non-malleable code and the leakage resilient secret sharing schemes.

D.4.2 Statistical Privacy

To prove the statistical privacy of the scheme, we use a hybrid argument. We wish to show that, for any unauthorized set T with $|T| < t$ and for any two messages $m_0 \neq m_1 \in \mathcal{M}$, $\mathsf{Share}(m_0)_T \approx_{2\epsilon_3'+\epsilon_2} \mathsf{Share}(m_1)_T$. The sequence of hybrids are:

[24] ϵ_3' denotes the privacy error and ϵ_3 denotes the leakage resilience error

[25] ϵ_4' denotes the privacy error and ϵ_4 denotes the leakage resilience error

- Hyb_0: This corresponds to the distribution of shares of m_0 in the unauthorized set T.
  Generate $(\mathsf{L},\mathsf{R}) \leftarrow \mathsf{Enc}(m_0)$. Further, get $(\mathsf{L}_1,\cdots,\mathsf{L}_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L})$ and $(\mathsf{R}_1,\cdots,\mathsf{R}_N) \leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R})$. Set $\mathsf{Sh}_i = \mathsf{L}_i,\mathsf{R}_i$, for each $i \in T$. Output: $\{\mathsf{Sh}_i\}_{i\in T}$.

- Hyb_1: Replace the shares of L in the set T with the shares of the left state L' corresponding to m_1.
  Generate $(\mathsf{L},\mathsf{R}) \leftarrow \mathsf{Enc}(m_0)$ and $(\mathsf{L}',\mathsf{R}') \leftarrow \mathsf{Enc}(m_1)$. Further, get $(\mathsf{L}'_1,\cdots,\mathsf{L}'_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L}')$ and $(\mathsf{R}_1,\cdots,\mathsf{R}_N) \leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R})$. Set $\mathsf{Sh}_i = \mathsf{L}'_i,\mathsf{R}_i$, for each $i \in T$. Output: $\{\mathsf{Sh}_i\}_{i\in T}$.

- Hyb_2: Replace the right state R corresponding to m_0 in share generation to the right state R'' corresponding to m_1. Note that, while both L_is and R_is are generated from m_1 in this hybrid, they are generated from different copies of the encoding of m_1.
  Generate $(\mathsf{L}',\mathsf{R}') \leftarrow \mathsf{Enc}(m_1)$ and $(\mathsf{L}'',\mathsf{R}'') \leftarrow \mathsf{Enc}(m_1)$. Further, get $(\mathsf{L}'_1,\cdots,\mathsf{L}'_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L}')$ and $(\mathsf{R}''_1,\cdots,\mathsf{R}''_N) \leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R}'')$. Set $\mathsf{Sh}_i = \mathsf{L}'_i,\mathsf{R}''_i$, for each $i \in T$. Output: $\{\mathsf{Sh}_i\}_{i\in T}$.

- Hyb_3: This corresponds to the distribution of shares of m_1 in the unauthorized set T.
  Generate $(\mathsf{L},\mathsf{R}) \leftarrow \mathsf{Enc}(m_1)$. Further, get $(\mathsf{L}_1,\cdots,\mathsf{L}_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L})$ and $(\mathsf{R}_1,\cdots,\mathsf{R}_N) \leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R})$. Set $\mathsf{Sh}_i = \mathsf{L}_i,\mathsf{R}_i$ for each $i \in T$. Output: $\{\mathsf{Sh}_i\}_{i\in T}$.

Clearly $\mathsf{Hyb}_0 \equiv \mathsf{Share}(m_0)_T$ and $\mathsf{Hyb}_3 \equiv \mathsf{Share}(m_1)_T$.
Now, by the statistical privacy of $(\mathsf{LRShare}^1_{(t,N)},\mathsf{LRRec}^1_{(t,N)})$, it is straightforward to see that $\mathsf{Hyb}_0 \approx_{\epsilon'_3} \mathsf{Hyb}_1$.
As the NMC satisfies the secret sharing property that $\mathsf{R} \approx_{\epsilon_2} \mathsf{R}''$, for $(\mathsf{L},\mathsf{R}) \leftarrow \mathsf{Enc}(m_0)$ and $(\mathsf{L}'',\mathsf{R}'') \leftarrow \mathsf{Enc}(m_1)$, it directly follows that $\mathsf{Hyb}_1 \approx_{\epsilon_2} \mathsf{Hyb}_2$.
Finally, to get the distribution identical to $\mathsf{Share}(m_1)_T$, we apply the statistical privacy of $(\mathsf{LRShare}^1_{(t,N)},\mathsf{LRRec}^1_{(t,N)})$ again and it follows that $\mathsf{Hyb}_2 \approx_{\epsilon'_3} \mathsf{Hyb}_3$. Hence, we get $\mathsf{Hyb}_0 \equiv \mathsf{Share}(m_0)_T \approx_{2\cdot\epsilon'_3+\epsilon'_4} \mathsf{Hyb}_2 \equiv \mathsf{Share}(m_1)_T$

D.4.3 Leakage Resilient Non-Malleability

We prove this through a sequence of hybrids. We first describe the simulator $\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}$ for $(G,\mathcal{L},f_1,\cdots,f_N,I) \in \mathcal{F}_N^{t-4,\tau}$.

$\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}$:

1. $(\mathsf{L}^{\$},\mathsf{R}^{\$}) \leftarrow \mathsf{Enc}(m^{\$})$, where $m^{\$}$ is a random message.
2. $(\mathsf{L}^{\$}_1,\cdots,\mathsf{L}^{\$}_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L}^{\$};r_L)$
   $(\mathsf{R}^{\$}_1,\cdots,\mathsf{R}^{\$}_N) \leftarrow \mathsf{LRShare}^2_{(t-1,N)}(\mathsf{R}^{\$}:r_R)$
3. For each $i \in [N]$, set $\mathsf{Sh}^{\$}_i = (\mathsf{L}^{\$}_i,\mathsf{R}^{\$}_i)$.
4. Get $\mathsf{Leak} \leftarrow G(\{\mathsf{Sh}^{\$}_i\}_{i\in\mathcal{L}})$. Recall that $|\mathcal{L}| \le t-4$
5. Get the reconstruction set $T := I(\mathsf{Leak}) = \{i_1,\cdots,i_t\}$. Recall that T is such that $\mathcal{L}\cap T = \phi$.
6. Let $\mathsf{aux}^1 \leftarrow \mathsf{aux}^1_{\{i_1,\cdots,i_{t-1}\},\{i_t\}}(\mathsf{L}^{\$};r_L)$ and $\mathsf{aux}^2 \leftarrow \mathsf{aux}^2_{\{i_3,\cdots,i_t\},\{i_1,i_2\}}(\mathsf{R}^{\$};r_R)$, where $\mathsf{aux}^1_{\{i_1,\cdots,i_{t-1}\},\{i_t\}}$ and $\mathsf{aux}^2_{\{i_3,\cdots,i_t\},\{i_1,i_2\}}$ are the functions guaranteed by the conditional independence of $\mathsf{LRShare}^1_{(t,N)}$ and $\mathsf{LRShare}^2_{(t-1,N)}$ respectively.
7. Define a hardcoding h, for the tampering functions of underlying NMC as:
   Set $h := (\{\mathsf{L}^{\$}_{i_j},\widetilde{\mathsf{L}^{\$}_{i_j}}\}_{j=1,\cdots,t-1},\{\mathsf{R}^{\$}_{i_j},\widetilde{\mathsf{R}^{\$}_{i_j}}\}_{j=3,\cdots,t-1},\mathsf{R}^{\$}_{i_t},\mathsf{aux}^1,\mathsf{aux}^2,\mathsf{Leak})$,
   where $(\widetilde{\mathsf{L}^{\$}_k},\widetilde{\mathsf{R}^{\$}_k}) = f_k(\mathsf{L}^{\$}_k,\mathsf{R}^{\$}_k,\mathsf{Leak})\ \forall k \in T$
8. Define the tampering functions F_h and G_h on underlying NMC code as:
   $F_h(\mathsf{L})$:
   - Pick L_{i_t} satisfying the following condition:
     L_{i_t} is consistent with $(\mathsf{L}^{\$}_{i_1},\cdots,\mathsf{L}^{\$}_{i_{t-1}},\mathsf{aux}^1,\mathsf{L})$.
     As in Definition 9, this means that $\mathsf{L}^{\$}_{i_t} = \mathsf{LRShare}^1_{(t,N)}(\mathsf{L};r'_L)_{i_t}$, where r'_L is such that $\mathsf{aux}^1 = \mathsf{aux}^1_{\{i_1,\cdots,i_{t-1}\},\{i_t\}}(\mathsf{L};r'_L)$ and $\mathsf{L}^{\$}_{T\setminus\{i_t\}} = \mathsf{LRShare}^1_{(t,N)}(\mathsf{L};r'_L)_{T\setminus\{i_t\}}$.
   - If no such L_{i_t} is found, output $\perp$.
   - $(\widetilde{\mathsf{L}_{i_t}},.) = f_{i_t}(\mathsf{L}_{i_t},\mathsf{R}^{\$}_{i_t},\mathsf{Leak})$.
   - Output $\tilde{\mathsf{L}} := \mathsf{LRRec}^1_{(t,N)}(\{\widetilde{\mathsf{L}^{\$}_{i_j}}\}_{j=1,\cdots,t-1},\widetilde{\mathsf{L}_{i_t}})$

   $G_h(\mathsf{R})$:
   - Pick $\mathsf{R}_{i_1},\mathsf{R}_{i_2}$ satisfying the following conditions:
     a) $\mathsf{R}_{i_1},\mathsf{R}_{i_2}$ are consistent with $(\mathsf{R}^{\$}_{i_3},\cdots,\mathsf{R}^{\$}_{i_t},\mathsf{aux}^2,\mathsf{R})$. (Again as in Definition 9)
     b) For each $j = 1,2$, $f_{i_j}(\mathsf{L}^{\$}_{i_j},\mathsf{R}_{i_j}) = (\widetilde{\mathsf{L}^{\$}_{i_j}},.)$.
   - If no such sampling is possible, output $\perp$.
   - For $j = 1,2$, $(.,\widetilde{\mathsf{R}_{i_j}}) = f_{i_j}(\mathsf{L}^{\$}_{i_j},\mathsf{R}_{i_j},\mathsf{Leak})$.
   - Output $\tilde{\mathsf{R}} := \mathsf{LRRec}^2_{(t-1,N)}(\widetilde{\mathsf{R}_{i_1}},\widetilde{\mathsf{R}_{i_2}},\{\widetilde{\mathsf{R}^{\$}_{i_j}}\}_{j=3,\cdots,t-1})$
9. Obtain $\tilde{m} \leftarrow \mathsf{NMSim}_{F_h,G_h}$ and
   Output: $\mathsf{Leak},\tilde{m}$.

Now, we follow a sequence of hybrids to show that $Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}, m) \approx_{\epsilon_1+\epsilon_3+\epsilon_4} \mathsf{STamper}_m^{G,\mathcal{L},f_1,\cdots,f_N,I}$.

$\underline{\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}}$: This hybrid is same as $Copy(\mathsf{Sim}^{G,f_1,\cdots,f_N,I}, m)$ with $\mathsf{Sim}^{G,f_1,\cdots,f_N,I}$ as described above, except we **change Step 9** to be the tamper random variable of the underlying NMC, $\mathsf{NMTamper}_{F_h,G_h}^m$. $Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}, m) \approx_{\epsilon_1} \mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}$

Proof. The proof of the claim is straightforward. We reduce the indistinguishability to the non-malleability of the underlying split-state NMC $(\mathsf{Enc}, \mathsf{Dec})$. The reduction algorithm can generate the leakage Leak and the hardcoding bit h completely on its own. Hence, the functions F_h, G_h (which are in the split-state model) for the tampering of the NMC code can be forwarded to the NMC challenger, along with message m. The response of the challenger exactly decides whether it is $Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}, m)$ or $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}$. Hence, this claim is proved. □

$\underline{\mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I}}$: In this hybrid, we replace the use of shares $\mathsf{L}_1^\$, \cdots, \mathsf{L}_N^\$$ in the hardcoding h and in generating the leakage Leak, with the left shares $\mathsf{L}_1, \cdots, \mathsf{L}_N$ corresponding to the actual message m. So, instead of using $\mathsf{L}^\$$, we use L generated from m in the whole hybrid. Rest of the steps are exactly as in $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}$. $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I} \approx_{\epsilon_3} \mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I}$

Proof. Suppose for contradiction that the statistical distance between $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}$ and $\mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I}$ is greater than ϵ_3. Here is the reduction, which breaks the leakage resilience of $(\mathsf{LRShare}_{(t,N)}^1, \mathsf{LRRec}_{(t,N)}^1)$ (as in Defintion 9):

1. Generate $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$ and $(\mathsf{L}^\$, \mathsf{R}^\$) \leftarrow \mathsf{Enc}(m^\$)$.

2. Further generate $(\mathsf{R}_1^\$, \cdots, \mathsf{R}_N^\$) \leftarrow \mathsf{LRShare}_{(t-1,N)}^2(\mathsf{R}^\$; r_R)$ and
$\mathsf{aux}^2 \leftarrow \mathsf{aux}_{\{i_3,\cdots,i_{t-1}\},\{i_1,i_2\}}^2(\mathsf{R}^\$; r_R)$.

3. Give L and $\mathsf{L}^\$$ as the two messages to the leakage resilience challenger.

4. For the leakage function G over the total set of indices $\mathcal{L}$, forward the leakage queries $G_{\{\mathsf{R}_k^\$\}_{k\in\mathcal{L}}}$, with the corresponding R_k's hardwired. Hence, the leakage $\mathsf{Leak}_b := G(\{\mathsf{L}_k^b, \mathsf{R}_k^\$\}_{k\in\mathcal{L}})$ can be obtained from the leakage resilience challenger. Here b denotes the choice bit of the leakage resilience challenger.

5. After all leakage queries, generate $T := I(\mathsf{Leak}_b) = \{i_1, \cdots, i_t\}$.

6. Now query the leakage challenger for $t-1$ full shares $\{\mathsf{L}_{i_1}^b, \cdots, \mathsf{L}_{i_{t-1}}^b\}$. Further, it also receives aux_b^1 from the leakage resilience challenger. Now, evaluate $(\widetilde{\mathsf{L}_{i_j}^b}, \widetilde{\mathsf{R}_{i_j}^\$}) = f_{i_j}(\mathsf{L}_{i_j}^b, \mathsf{R}_{i_j}^\$, \mathsf{Leak}_b)$, for each $j = 1, \cdots, t-1$.

7. Set $h := (\{\mathsf{L}_{i_j}^b, \widetilde{\mathsf{L}_{i_j}^b}\}_{j=1,\cdots,t-1}, \{\mathsf{R}_{i_j}^\$, \widetilde{\mathsf{R}_{i_j}^\$}\}_{j=3,\cdots,t-1}, \mathsf{R}_{i_t}^\$, \mathsf{aux}_b^1, \mathsf{aux}^2, \mathsf{Leak}_b)$.

8. Now the reduction outputs $\tilde{m} \leftarrow \mathsf{NMTamper}_{F_h,G_h}^m$, where F_h and G_h are as defined in $\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_N,I}$ and the leakage Leak_b.

The reduction makes joint and adaptive leakage queries on at most $|\mathcal{L}| \leq t-4 < t-1$ shares in all. At the end of the joint and adaptive leakage queries, it makes the full share queries for $t-1$ fresh shares (since $T \cap \mathcal{L} = \phi$). So clearly the leakage model is in the family JALR^{t,τ_1}, for $\tau_1 = \tau$ (since no additional leakage queries are made by the reduction to the leakage resilience challenger). If the leakage challenger uses $\mathsf{L}^{\$}$, then the reduction output is identical to $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I}$ and else, if it uses L, then the reduction output is identical to $\mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I}$. Hence, this breaks the leakage resilience of $(\mathsf{LRShare}^1_{(t,N)}, \mathsf{LRRec}^1_{(t,N)})$. □

$\underline{\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}}$: In this hybrid, instead of the function F_h sampling L_{i_t} again such that it satisfies the consistency condition, we now let F_h use the same share L_{i_t} that was used to generate h. $\mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}$

Proof. The proof of this claim is direct from the conditional independence of $\mathsf{LRShare}^1_{(t,N)}$ (with $K = \{i_1, \cdots, i_{t-1}\}$ and $S = \{i_t\}$). □

$\underline{\mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I}}$: In this hybrid, we replace the use of the $\mathsf{R}_1^{\$}, \cdots, \mathsf{R}_N^{\$}$ in the hardcoding h and in generating the leakage Leak, with the right shares $\mathsf{R}_1, \cdots, \mathsf{R}_N$ corresponding to the actual message m. So, instead of using $\mathsf{R}^{\$}$, we use R generated from m in the whole hybrid. Rest of the steps are exactly as in $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}$. $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I} \approx_{\epsilon_4} \mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I}$

Proof. Suppose for contradiction that the statistical distance between $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}$ and $\mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I}$ is greater than ϵ_4. Here is the reduction, which breaks the leakage resilience of $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$ (as in Definition 9):

1. Generate $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$ and $(\mathsf{L}^{\$}, \mathsf{R}^{\$}) \leftarrow \mathsf{Enc}(m^{\$})$.

2. Further generate $(\mathsf{L}_1, \cdots, \mathsf{L}_N) \leftarrow \mathsf{LRShare}^1_{(t,N)}(\mathsf{L}; r_L)$ and
$\mathsf{aux}^1 \leftarrow \mathsf{aux}^1_{\{i_1,\cdots,i_{t-1}\},\{i_t\}}(\mathsf{L}; r_L)$[26].

3. Give R and $\mathsf{R}^{\$}$ as the two messages to the leakage resilience challenger.

4. For the leakage function G over the total set of indices $\mathcal{L}$, forward the leakage queries $G_{\{\mathsf{L}_k\}_{k\in\mathcal{L}}}$, with corresponding L_ks hardwired. Hence, the leakage $\mathsf{Leak}_b := G(\{\mathsf{L}_k, \mathsf{R}_k^b\}_{k\in\mathcal{L}})$ can be obtained from the leakage resilience challenger. Here b denotes the choice bit of the leakage resilience challenger.

5. After all leakage queries, generate $T := I(\mathsf{Leak}_b) = \{i_1, \cdots, i_t\}$.

6. Now, we make an additional joint leakage query on indices $i_1, i_2 \notin \mathcal{L}$ (Since $T \cap \mathcal{L} = \phi$). Query the leakage resilience challenger on leakage function g_{i_1,i_2} on set of indices $\{i_1, i_2\}$, with hardcoded values Leak_b and $\{\mathsf{L}_{i_1}, \mathsf{L}_{i_2}\}$. g_{i_1,i_2} is defined as:

   On Input: $\{\mathsf{R}^b_{i_1}, \mathsf{R}^b_{i_2}\}$

   Evaluate $(\widetilde{\mathsf{L}_{i_j}}, .) = f_{i_j}(\mathsf{L}_{i_j}, \mathsf{R}^b_{i_j}, \mathsf{Leak}_b)$, for $j = 1, 2$.

   Output: $\{\widetilde{\mathsf{L}_{i_1}}, \widetilde{\mathsf{L}_{i_2}}\}$

[26] We are defining aux^1 only for completion in setting h but note that aux^1 will not be used anymore, as we are not resampling shares of L anymore

7. Now query the leakage challenger for $t-2$ full shares $\{\mathsf{R}^b_{i_3}, \cdots, \mathsf{R}^b_{i_t}\}$. Further, it also receives aux^2_b from the leakage resilience challenger. Now, evaluate $(\widetilde{\mathsf{L}_{i_j}}, \widetilde{\mathsf{R}^b_{i_j}}) = f_{i_j}(\mathsf{L}_{i_j}, \mathsf{R}^b_{i_j}, \mathsf{Leak}_b)$, for each $j = 3, \cdots, t$.

8. Reconstruct to get $\widetilde{\mathsf{L}} = \mathsf{LRRec}^1_{(t,N)}(\widetilde{\mathsf{L}_{i_1}}, \cdots, \widetilde{\mathsf{L}_{i_t}})$.

9. Set $h := (\{\mathsf{L}_{i_j}, \widetilde{\mathsf{L}_{i_j}}\}_{j=1,\cdots,t-1}, \{\mathsf{R}^b_{i_j}, \widetilde{\mathsf{R}^b_{i_j}}\}_{j=3,\cdots,t-1}, \mathsf{R}^b_{i_t}, \mathsf{aux}^1, \mathsf{aux}^2_b, \mathsf{Leak}_b)$.

10. With G_h as defined in $\mathsf{Sim}^{G,f_1,\cdots,f_N,I}$, get $\widetilde{\mathsf{R}} = G_h(\mathsf{R})$.

11. The reduction outputs $\tilde{m} = \mathsf{Dec}(\widetilde{\mathsf{L}}, \widetilde{\mathsf{R}})$ and the leakage Leak_b.

The reduction makes joint and adaptive leakage queries on at most $|\mathcal{L}| + 2 \leq (t-4) + 2 = t-2$ shares in all. At the end of all these queries, it makes the full share queries for $t-2$ fresh shares (as $T \cap \mathcal{L} = \phi$). So clearly the leakage model is in the family $\mathsf{JALR}^{t-1,\tau_2}$, for $\tau_2 = \tau + \eta_1$ (since $|\widetilde{\mathsf{L}_k}| = \eta_1$ and the query made can be viewed as independent query on two shares of R). If the leakage challenger uses $\mathsf{R}^\$$, then the reduction output is identical to $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}$ and else, if it uses R, then the reduction output is identical to $\mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I}$. Hence, this breaks the leakage resilience of $(\mathsf{LRShare}^2_{(t-1,N)}, \mathsf{LRRec}^2_{(t-1,N)})$. □

$\underline{\mathsf{Hyb}_5^{G,\mathcal{L},f_1,\cdots,f_N,I}}$: Finally, we repeat what we did in $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I}$ with respect to the right shares. Instead of G_h sampling $\mathsf{R}_{i_1}, \mathsf{R}_{i_2}$ again such that they satisfy the consistency conditions, we let G_h use the same shares $\mathsf{R}_{i_1}, \mathsf{R}_{i_2}$ that were used in generating h. $\mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{Hyb}_5^{G,\mathcal{L},f_1,\cdots,f_N,I}$

Proof. The proof of this claim is direct from the conditional independence of $\mathsf{LRShare}^2_{(t-1,N)}$ (with $K = \{i_3, \cdots, i_{t-1}\}$ and $S = \{i_1, i_2\}$). □

Now, notice that $\mathsf{Hyb}_5^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{STamper}_m^{G,f_1,\cdots,f_N,I}$. Hence, $Copy(\mathsf{Sim}^{G,f_1,\cdots,f_N,I}, m) \approx_{\epsilon_1} \mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_N,I} \approx_{\epsilon_3} \mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_N,I} \approx_{\epsilon_4} \mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{Hyb}_5^{G,\mathcal{L},f_1,\cdots,f_N,I} \equiv \mathsf{STamper}_m^{G,f_1,\cdots,f_N,I}$. This proves the leakage resilient non-malleability of the construction.

D.5 Rate Analysis

We instantiate our leakage resilient non-malleable secret sharing construction for $\mathcal{F}_N^{t-4,\tau}$ with the following underlying primitives:

- We use the constant rate 2-split-state non-malleable code of [AO19]:

  Theorem 6. *[AO19] There exists an efficient, information-theoretically secure ϵ-non-malleable code in the 2-split-state model with rate $O(1)$ and error $\epsilon = 2^{-k^{\Omega(1)}}$, where k is the message length.*

  Hence, for $|m| = k$, we get $\epsilon_1 = 2^{-k^{\Omega(1)}}$, $|\mathsf{L}| = O(k)$ bits and $|\mathsf{R}| = O(k)$ bits.

- Further, we instantiate $(\mathsf{LRShare}^1_{(t,n)}, \mathsf{LRRec}^1_{(t,n)})$ and $(\mathsf{LRShare}^2_{(t-1,n)}, \mathsf{LRRec}^2_{(t-1,n)})$ with the construction from Section 4.1 (with instantiation C specifically for threshold access structure), with leakage thresholds $\tau_1 = \tau$ and $\tau_2 = \tau + \eta_1$ respectively ($\eta_1 = |\mathsf{L}_i|$). This gives us that $\eta_1 = |\mathsf{L}_i| = O(|\mathsf{L}|/R) = O(k/R)$ and $\eta_2 = |\mathsf{R}_i| = O(k/R)$. Further $\epsilon_3 = 2^{-\Omega(k)}$ and $\epsilon_4 = 2^{-\Omega(k)}$ [27].

Combining these two instantiations, we get: $|\mathsf{Sh}_i| = |\mathsf{L}_i| + |\mathsf{R}_i| = O(k/R)$ and hence, we get the rate $\Omega(R)$. The error is $\epsilon_1 + \epsilon_3 + \epsilon_4 = 2^{-k^{\Omega(1)}}$.
We obtain the following corollary: For any $N \in \mathbb{N}$, if there exists a statistically private (t, N, ϵ)-threshold secret sharing scheme with rate R, then there exists a leakage resilient non-malleable secret sharing scheme against $\mathcal{F}_N^{t-4,\tau}$ with rate $\Omega(R)$ and simulation error $\epsilon + 2^{-k^{\Omega(1)}}$.

[27] We take the R with appropriate padding to ensure that additional leakage of size η_1 can be obtained from R_i, but this is only a constant blow-up in size and hence η_2 remains $O(k/R)$