Adaptive Extractors and their Application to Leakage Resilient Secret Sharing

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

June 24, 2021

Abstract

We introduce Adaptive Extractors, which unlike traditional randomness extractors, guarantee security even when an adversary obtains leakage on the source *after* observing the extractor output. We make a compelling case for the study of such extractors by demonstrating their use in obtaining adaptive leakage in secret sharing schemes.

Specifically, at FOCS 2020, Chattopadhyay, Goodman, Goyal, Kumar, Li, Meka, Zuckerman, built an adaptively secure leakage resilient secret sharing scheme (LRSS) with both rate and leakage rate being $\mathcal{O}(1/n)$, where n is the number of parties. In this work, we build an adaptively secure LRSS that offers an interesting trade-off between rate, leakage rate, and the total number of shares from which an adversary can obtain leakage. As a special case, when considering t-out-of-n secret sharing schemes for threshold $t = \alpha n$ (constant $0 < \alpha < 1$), we build a scheme with a constant rate, constant leakage rate, and allow the adversary leakage from all but $t - 1$ of the shares, while giving her the remaining $t - 1$ shares completely in the clear. (Prior to this, constant rate LRSS scheme tolerating adaptive leakage was unknown for *any* threshold.)

Finally, we show applications of our techniques to both non-malleable secret sharing and secure message transmission.

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

†Department of Computer Science and Automation, Indian Institute of Science, Email: bhavana@iisc.ac.in. Research supported by Microsoft Research Grant.

‡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. Research supported by TCS Research Grant.

Contents

1 Introduction

Randomness extractors [32] are a fundamental primitive in the world of theoretical computer science, which have found widespread applications in derandomization techniques, cryptography, and so on. 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) and outputs $\mathsf{Ext}(W;S)$ such that $\mathsf{Ext}(W;S)$ "looks uniform" to an unbounded eavesdropper Eve even given the seed S. Unfortunately, the standard notion of extractors offers no guarantees whatsoever if the adversary Eve obtains some information about W, after observing, the output of the extractor. In this work, we address this gap.

Does the security of extractors hold even after the adversary obtains some information on W, "after the fact"?

Naturally, we have to be careful about what information Eve can learn about W and S, after the fact. For instance, the function f, which on input w, s and the extractor challenge y, outputs 1 if and only if $y = \mathsf{Ext}(w;s)$, is an after the fact leakage function, which can break extractor security, with high probability, with only 1 bit of leakage. Hence, one needs to define the leakage function family carefully.

In this work, we introduce the notion of *adaptive extractors* with respect to an after the fact leakage family $\mathcal{F}$. Formally, we say that an extractor is an adaptive extractor with respect to a function family $\mathcal{F}$, if for each $f \in \mathcal{F}$, an adversary cannot (statistically) distinguish $(S, f(W, \mathsf{Ext}(W;S)), \mathsf{Ext}(W;S))$ from $(S, f(W,U), U)$. Our notion of adaptive extractors can be seen as a generalization of exposure-resilient extractors introduced by Zimand [39] (Zimand's extractors allow the adversary to adaptively learn up to n^δ bits of the source, for some $\delta < 1$ bits; the adversary can determine which bits to query based on an arbitrary function of the extractor output.), and of the notion of adaptive non-malleable extractors introduced by Aggarwal *et al.* in [2] (where adaptive non-malleability particularly guarantees that the adversary cannot distibuish between $(S, \mathsf{Ext}(W; g(S, \mathsf{Ext}(W;S))), \mathsf{Ext}(W;S))$ and $(S, \mathsf{Ext}(W; g(S,U)), U))$. We then observe 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 note that this observation is similar to how the authors in [2, Lemma 3.5] show that every non-malleable extractor is adaptive non-malleable, with some loss in parameters. We demonstrate that, in spite of the loss in parameters that adaptivity incurs, such extractors can be powerful. In particular, we use them to build constant-rate secret sharing schemes resilient to adaptive leakage. We now describe these contributions in greater detail.

Secret Sharing. Secret sharing schemes [34, 11] are a fundamental cryptographic primitive and have many applications, such as in multi-party computation [8, 15], and leakage-resilient circuit compilers [25, 21, 33]. 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 $t-$out-of-n threshold secret sharing scheme, there are n parties, and any collection of t ($t \leq n$) or more parties would correspond to 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 [27] tells 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 [19] introduced the problem of leakage resilience in secret sharing schemes. This problem has received much attention (for example, [16, 31, 3, 22, 10, 36, 28, 1, 20, 13], [14, 12]), 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 explore the stronger 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. In this adaptive leakage setting[2], the construction of [14] achieved a rate of $\mathcal{O}(1/n)$ as well as a leakage rate of $\mathcal{O}(1/n)$. A consequence of this is that there currently does not exist a scheme with constant rate and leakage rate for any threshold in this strong leakage model, whereas we do know of such constructions for the non-adaptive leakage model. Our work fills this gap precisely.

1.1 Our Results

Our first and main result on the LRSS scheme in the adaptive leakage model is as follows. Here n denotes the number of parties, t denotes the threshold and l denotes the message length.

Result 1: *We build an LRSS scheme, tolerating ψ adaptive queries, each dependent on X shares (with $\psi \cdot X \leq n - t + 1$) and the reveal of the remaining $t - 1$ shares, such that it achieves a rate of $(X^{\Theta(\psi X/t)})^{-1}$, while allowing $\Theta(l)$ bits of leakage per query, for threshold access structures. In particular, for a constant X and $n = \Theta(t)$, this gives the first constant-rate adaptive LRSS scheme for the threshold access structure. Finally, we also generalize our constructions to the first constant-rate adaptive LRSS for general access structures.*

Further, we show the following applications of our LRSS scheme.

Result 2: *As an application of our LRSS, we show compilers to get a leakage resilient non-malleable secret sharing (LRNMSS) scheme (which are LRSS schemes, additionally resilient to tampering attacks), and an information-theoretic secure message transmission protocol (SMT), tolerant against leakage and tampering attacks. The rates of both these schemes translate appropriately from the rate of the LRSS. In particular, for a constant LRSS, we get constant-rate schemes for both LRNMSS and SMTs.*

[1] For example, Guruswami and Wooters [24] 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.

[2] We note that here we only compare in an adaptive leakage model, without any joint leakage queries on multiple shares (which is called the *bounded collusion protocols* (BCP) model), for ease of expostion, and discuss the joint model in the technical section later.

1.2 Our Techniques

We begin by describing the leakage model for LRSS and then give a technical overview of our scheme. For simplicity, we provide our technical overview for threshold access structures (which we can extend to general access structures as well). Let t denote the threshold and n, the number of parties.

Leakage Model. We allow the adversary to obtain adaptive leakage on $n-(t-1)$ shares and then reveal the full shares of the remaining $t-1$ shares. Each adaptive query can be on a set of at most X shares (where X is some value between 1 and $t-1$), and different queries must be on sets that are disjoint from the prior queries. For the purposes of this exposition, we make the following restriction to our model: we assume that the adversary makes adaptive queries but only on a single share each time, i.e., it doesn't make any leakage query on multiple shares.

Warm-up construction. To motivate our construction, we consider the following modification[3] of a construction due to Srinivasan and Vasudevan in [36, Section 3.2.1]. Take any t-out-of-n secret sharing scheme $(\mathsf{MShare}, \mathsf{MRec})$ and then do as follows:

- Sample shares $(m_1, .., m_n)$ of the message m using MShare.
- Choose an extractor seed s and split s into $(sd_1, .., sd_n)$ using a t-out-of-n secret sharing scheme.
- Now, for every m_i, choose an extractor source w_i uniformly and compute $y_i = m_i \oplus \mathsf{Ext}(w_i; s)$.
- Finally, output the final shares $\{sh_i\}$ as $\{(w_i, y_i, sd_i)\}$.

For now, consider a weak model, where the adversary obtains only non-adaptive and independent leakage from a total of (say) $t-1$ shares, in addition to $t-1$ full shares. The hope is to show that the $t-1$ leakage queries are independent of the message shares m_i, following which the privacy of MShare can be used to get the $t-1$ full shares. One might hope to show this independence of leakage from the m_i's, using the security of the extractor as follows: Pick sd_i uniformly at random and independent of s; then the leakage function on $\{sh_i\}$, can be answered as an auxiliary leakage query on the source w_i. Once s is revealed in the extractor security game, the reduction can pick the other sd_j values in a consistent manner. However, this proof strategy has a flaw. For extractor security, it is important that the auxiliary leakage query on w is independent of s; however, there is a dependence on s via y_i. In other words, it is unclear how to prove that this construction satisfies leakage resilience even in a weak model where the adversary obtains leakage only independently and non-adaptively.

Fortunately, with adaptive extractors, we can show that this construction is secure not only in this weak model but also in a stronger model where the adversary is allowed to leak from $t-1$ shares adaptively, before receiving $t-1$ full shares. Furthermore, this construction even has a constant rate! The high-level idea of security is as follows. We wish to reduce the adaptive leakage queries on the shares to an adaptive extractor leakage query. Since the adaptive leakage query on w_j cannot depend on the seed, we need to first show that the share sd_j in the corresponding query is independent of the seed s. Indeed, using the privacy of secret sharing[4], we can show that for

[3] We note that the original construction of [36] only aimed to achieve non-adaptive security, and we consider a modification, with the aim to expand to adaptive security.

[4] Since the leakage queries are adaptive, we require adaptive privacy of the underlying secret sharing scheme, and we show instantiations of the same.

the first $t-1$ queries, the shares sd_j in sh_j can be replaced with shares of 0 (hence removing the dependence on s). Then, using the adaptive extractor security, we can replace the y_j's (for the first $t-1$ queries) with uniform, where the leakage can be asked on the w_j's. Now, the privacy of MShare can be invoked to get the $t-1$ full shares.

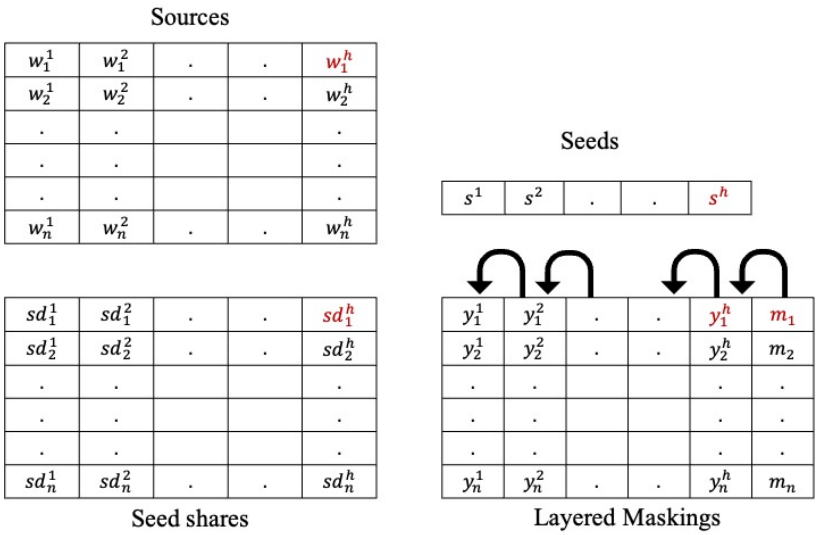
Main construction. Our next goal is to leverage adaptive extractors to go beyond leaking from just $t-1$ shares. The main bottleneck is that for any subsequent leakage query (beyond $t-1$), the sd_j's will reveal s, and hence the adaptive leakage query on subsequent w_j's will no longer remain independent of the seed s. Thus, extractor security fails. This is the challenge we must address to achieve our main result where the adversary is allowed to obtain adaptive leakage on $n-(t-1)$ shares (in total) and reveal $t-1$ of the remaining shares.

One approach to facilitating leakage from more than $t-1$ shares could be to use independent extractor seeds to extract independent random masks. Consider the following modification of the above construction: mask the share of a message m_i not just with one extractor output but with many. In particular, let $y_i = m_i \oplus \mathsf{Ext}(w_i; s^1) \oplus \mathsf{Ext}(w_i; s^2) \ldots \oplus \mathsf{Ext}(w_i; s^h)$, for some parameter h, where $s^1 \ldots s^h$ are independent seeds. We might hope that because we are using h seeds, we could hope to leak from $h(t-1)$ shares and use the security of each seed per batch of $t-1$ shares. Unfortunately, this doesn't work for the following reason: reconstruction is only possible if we recover all h seeds. This means that we ultimately need to somehow share all the seeds in a manner where they can be reconstructed from $t-1$ shares. In other words, once we leak from $t-1$ shares, we can no longer argue security by leveraging any of the seeds because they can all be reconstructed from $t-1$ shares. We overcome this challenge by carefully using a multi-layered approach for both masking the message shares as well as for reconstructing the seeds.

Construction Overview:

1. Pick h extractor seeds $s^1, \ldots, s^h$ and hn extractor sources $w_1^1, \ldots w_1^h, \ldots, w_n^1, \ldots, w_n^h$.
2. Secret share each of the h seeds using a t-out-of-n secret sharing scheme to obtain shares; let the share of s^j be $sd_1^j, \ldots, sd_n^j$
3. Each share m_j is masked using the h seeds in a layered manner as follows:
   (a) In level $h+1$: Set $y_j^{h+1} = m_j$.
   (b) For every subsequent lower level $i (i \geq 1)$, compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s_i)$ and set $y_j^i = (x_j^i || sd_j^i)$. [Note that we use a different extractor per-level since the length of the extractor outputs (and the length of y_j^is they mask) increase with level.]
   Finally set $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$.
4. Output $(Sh_1, \cdots, Sh_n)$

A pictorial representation of the construction can be found in Figure 1. In order to give an overview of the proof, we first recall that we are in a setting where each adaptive query of an adversary is a query on a single share – we can extend our results to the case of joint leakage but, for the sake of simplicity, we don't focus on that for now.



Each entry of the layered maskings matrix appropriately uses the corresponding entries of the sources, seeds and seed shares matrices. In addition, each entry y_i^j $(j \leq h)$ also depends on the subsequent value i.e., y_i^{j+1}. Example: $y_1^h = m_1 \oplus \mathsf{Ext}^h(w_1^h; s^h)||sd_1^h$ (colored red)

Figure 1: The Main Construction

At a high-level, the idea of the security proof is that we view the leakage queries in batches of $t-1$ queries. For the first set of $t-1$ queries, we rely on the adaptive security of the extractor outputs evaluated using seed s^1 and, in particular, all of these outputs can be replaced by uniform. (This also relies on the *adaptive* privacy of the secret sharing scheme, a notion we define and instantiate.) For the second set of $t-1$ queries, we can no longer assume that s^1 is hidden, since we can not use the privacy of the secret sharing scheme any more. However, two things come to our rescue: first, the second batch of queries helps unmask at most $t-1$ shares of s^2 and therefore, adaptive extractor security on seed s^2 can be leveraged; second, the extractor outputs $\mathsf{Ext}(w_j^1; s^1)$ (where j was a share that was leaked from in the first batch) continue to remain uniform. The reason for the latter is that all extractor sources are uniformly chosen, and our model requires a disjoint set of indices to be leaked from across batches. In short, for the first batch of queries, we use adaptive security of the extractor outputs evaluated on the first seed and, for every subsequent batch, we move to argue extractor security using the subsequent seed. Since we have h independent seeds, we can do this h times and therefore answer h batches of queries, i.e., we can obtain leakage on $h(t-1)$ shares.

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 [31, 10, 22, 6, 36, 1, 20, 13, 28, 30], [14, 12]. 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 [28, 14]. While our model allows adaptive queries on up to $n - t + 1$ shares, each dependent on at most X shares (where X is some value between 1 and $t - 1$), before fully revealing the remaining $t - 1$ shares, [14] allows adaptive queries on all n shares, each dependent on at most $t - 1$ shares before revealing $t - 1$ full shares. Both the schemes require the adaptive queries to be on disjoint sets of shares. However, our scheme/analysis offers a more fine-grained trade-off between the various parameters and allows us to obtain better results for certain settings. In particular, when we consider the instance where X is constant (and $t = \alpha n$, for a constant $\alpha < 1$), we get a constant-rate adaptive LRSS achieving a constant leakage rate, while [14] gets a rate and leakage rate of $\mathcal{O}(1/n)$ each, in all instances. To put this in context, even if [14] makes independent adaptive leakage queries on all shares, their rate is $\mathcal{O}(1/n)$ and 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 $(n - t + 1)$ times a constant fraction of the size of a single share!

The work of [14] also consider a variant of joint leakage, allowing overlap of the query sets, the detailed parameters of which are given in Table 1. We give a detailed comparison of the parameters achieved by the various schemes in Table 1, for the threshold setting with $t = \alpha n$ (for a constant $\alpha < 1$).

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	$\log(n)$
[CGG+20]	$O\left(\frac{1}{n}\right)$	Yes	n	$t-1$	$\approx \Theta\left(\frac{1}{n}\right)$	Yes	Unauthorized
[CGG+20] (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
[CGG+20] (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	$n-t+1$	*constant*	$\approx \Theta(1)$	Yes	Unauthorized**

Table 1: LRSS Prior Work

- *All works mentioned here are information-theoretic. We write all comparisons for the threshold setting with threshold $t = \alpha n$ (where $\alpha < 1$ is a constant and n denotes the total number of parties).
- ** For our result, the unauthorized queries cannot overlap with the leakage queries.
- c is a small constant and l_{msg} is the message length.
- All schemes (except the joint overlapping schemes of [14] (threshold and n-out-of-n) actually work for general access structures.
- **Full Shares**: Number of complete shares that an adversary can see (at the end of all leakage queries, in the adaptive schemes).

Open Problems. We believe that it would be interesting to explore the direction of building adaptive extractors against restricted classes of leakage families such as those captured by computational/bounded depth circuits, local functions, etc.

1.4 Organization of the Paper

We provide the preliminaries and definitions in Section 2. Then, we define and build adaptive extractors in Section 3. We define and build leakage resilient secret sharing schemes in Section 4.

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 $\{1, 2, \cdots, n\}$ and $[0]$ denotes a null set. $s \in_R S$ 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. 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 $\cup_{i \in S} T_i$.

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 [17], 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.

Lemma 1. *[17] 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, $\widetilde{\mathbf{H}}_\infty(A \mid B, C) \geq \widetilde{\mathbf{H}}_\infty(A, B \mid C) - \lambda \geq \widetilde{\mathbf{H}}_\infty(A \mid C) - \lambda$.*

Lemma 2. *[37] For any random variables A, B, if $A \approx_\epsilon B$, then for any function f, $f(A) \approx_\epsilon f(B)$.*

Lemma 3. *For any random variables A, B over $\mathcal{A}$, and events E, E' with non-zero probabilities,*

$$\mathbf{SD}\left(A \wedge E, B \wedge E'\right) \leq |\Pr[E] - \Pr[E']| + \Pr[E'] \cdot \mathbf{SD}\left(A|E, B|E'\right)$$

where,

$$\mathbf{SD}\left(A \wedge E, B \wedge E'\right) \stackrel{def}{=} \frac{1}{2} \sum_{a \in \mathcal{A}} |\Pr[A = a \wedge E] - \Pr[B = a \wedge E']|$$

and

$$\mathbf{SD}\left(A|E, B|E'\right) \stackrel{def}{=} \frac{1}{2} \sum_{a \in \mathcal{A}} |\Pr[A = a|E] - \Pr[B = a|E']|$$

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 $\epsilon = |\Pr[E] - \Pr[E']|$.

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

□

Lemma 4. *[4] Let X, Y, X', Y' be random variables such that $\mathbf{SD}\left((X, Y), (X', Y')\right) \leq \epsilon$ and S be any set such that $\Pr[Y \in S] > 0$ and $\Pr[Y' \in S] > 0$, then*

$$\mathbf{SD}\left(X|Y \in S, X'|Y' \in S\right) \leq \frac{2\epsilon}{\Pr[Y' \in S]}$$

2.2 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 1. *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 $\mathsf{Share} : \mathcal{M} \to (\{0,1\}^l)^n$ is a $(\mathcal{A}, n, \epsilon_s)$- **secret sharing scheme** with respect to a monotone access structure*[5] *$\mathcal{A}$ if the following two properties hold :*

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 (n,t)-threshold if and only if $\mathcal{A}$ contains all subsets of $[n]$ of size at least t.
***Rate** of a secret sharing scheme is defined as $\frac{\text{message size}}{\text{share size}}$ (which would be equal to $\frac{\log |\mathcal{M}|}{l}$).*

We now study a stronger privacy requirement, *adaptive privacy* (introduced by Bellare and Rogaway [7][6])

2.2.1 Adaptive Privacy

Statistical privacy captures privacy against any non-adaptively chosen unauthorized 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.
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 [7]
   - $\mathcal{D}$ 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 $b = b'$

[5] $\mathcal{A}$ 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.

[6] In [7], the authors refer to adaptive privacy as privacy against dynamic adversaries.

[7] 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 $\mathcal{D}$

While generally, any secret sharing scheme may not be adaptively private, we show in Appendix A.2.2 that for the threshold setting, the scheme of [34] and for the general access structures, the scheme of [9] are both adaptively private. We use them to instantiate our schemes.

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 require the following consistent re-sampling feature[8], which informally states that for any $(\mathcal{A}, n, \epsilon_s)$-secret sharing scheme and any message m, the distribution of shares which are re-sampled as shares of m, conditioned on some set T of shares (which are also generated as shares of m) chosen adaptively, is identical to the distribution of shares of m generated directly.

Lemma 5. *For any $(\mathcal{A}, n, \epsilon_s)$-secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ and for any message m, the following two distributions are identical.*

$\mathcal{D}_1$:	$\mathcal{D}_2$:
• $(sh'_1, .., sh'_n) \leftarrow \mathsf{Share}(m)$	• $(sh_1, .., sh_n) \leftarrow \mathsf{Share}(m)$
• $(sh_1, .., sh_n) \leftarrow \mathsf{Share}(m|sh'_T)$	
• *Output* $(sh_1, .., sh_n)$	• *Output* $(sh_1, .., sh_n)$

Here, $T \subseteq [N]$ can be any subset chosen as: every index (except the first) depends arbitrarily on the shares corresponding to all the previous indices.

We give a full proof of the above lemma in Appendix A.2.1.

3 Adaptive Extractors

Extractors (introduced by Nissan and Zuckerman [32]) output a near uniform string y, from a source w that only has min-entropy, using a short uniform string s, called the *seed*, as a catalyst. Average-case extractors are extractors whose output remains close to uniform, even given the seed and some auxiliary information (or leakage) about the source (independent of the seed), as long as the source has enough average entropy given this leakage. We give their formal definition below.

Definition 2. *[17] Let $\mathsf{Ext} : \{0,1\}^\eta \times \{0,1\}^d \to \{0,1\}^l$ be a polynomial time computable function. We say that Ext is an efficient average-case $(\eta, \mu, d, l, \epsilon)$-strong extractor if for all pairs of random variables (W, Z) such that W is an η-bit string satisfying $\widetilde{\mathbf{H}}_\infty(W|Z) \geq \mu$ (refer to Appendix **??** for the definition of min-entropy), we have*

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

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

3.1 Definition

Average-case extractors, unfortunately, provide no guarantees on the extractor output being uniform when an adversary can obtain an 'adaptive' leakage on the source, that is *dependent on the extractor output* and the seed. This is not surprising, as if an adversary can obtain *arbitrary adaptive leakage* on the source, then we cannot hope for the extractor output to remain uniform. For example, given $y = \mathsf{Ext}(w, s)$, an adversary can distinguish the extractor output from uniform with high probability by querying a single bit of auxiliary information that tells her whether $\mathsf{Ext}(w, s) = y$. However, as we will see later, in many applications, the adaptive leakage that the adversary obtains comes from a specific function family. Hence, by carefully defining this function family, we show how to obtain useful notions of extractors that guarantee security even in the presence of an adaptive auxiliary information. We introduce and call this notion *adaptive extractors* and now proceed to formally define them.

Definition 3. *An $(\eta, \mu, d, l, \epsilon)$- extractor Ext is said to be an $(\mathcal{F}, \delta)$-**adaptive extractor** if for all pairs of random variables (W, Z) such that W is an η-bit string satisfying $\widetilde{\mathbf{H}}_\infty(W|Z) \geq \mu$, and any function f in the function family $\mathcal{F}$, it holds that*

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

*We call δ, the **adaptive error** of the extractor.*

3.2 Construction

Generic relation. We show that every extractor is in fact an adaptive extractor for the family of leakage functions where the adaptive leakage depends only on the source and the extractor output (i.e., it doesn't depend on the seed except via the extractor output), with some loss in security. This loss, in fact, depends only on the number of bits of the extractor output that the adaptive leakage function depends on. For ease of exposition, we omit auxiliary information z that depends only on the source (but not on the extractor output or seed) from the notation below. We now explicitly define this family below:

$$\mathcal{F}_{a,\zeta} \subseteq \{f' : \{0,1\}^\eta \times \{0,1\}^l \to \{0,1\}^\zeta\}$$

$$\text{such that for every } f' \in \mathcal{F}_{a,\zeta} \text{ there exists two functions } f : \{0,1\}^l \to \{0,1\}^a \text{ and}$$

$$g : \{0,1\}^{\eta+a} \to \{0,1\}^\zeta \text{ such that } \forall w, y,\ f'(w, y) = g(w, f(y))\}$$

Here, 'ζ' denotes the number of bits of adaptive leakage and 'a' denotes the number of bits of the extractor output (or the uniform string) that 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 $f'(w, y) = g(w, f(y))$ where f's output is only a bits long. w and y should be interpreted as the source and the extractor output (or the uniform string) respectively.

The following theorem shows that any $(\eta, \mu, d, l, \epsilon)$- average case extractor can be shown to be adaptive secure against the above family $\mathcal{F}_{a,\zeta}$, with an adaptive error of $2^{a+2}\epsilon$. Informally, we can reduce the adaptive security to the extractor security (as in Definition 2) in the following way: to answer the adaptive leakage query, the reduction makes a guess, v, for the extractor challenge dependent value $f(y_b)$ (where, y_b is the extractor challenge), which is of a-bits, and gets the leakage $g(w, v)$ from the source. Now, it gets the challenge y_b from the extractor challenger

and if $f(y_b)$ matches the guess v, then the reduction can successfully simulate the challenge and the adaptive leakage response, else it cannot proceed (and aborts). Hence, the winning probability in the extractor game is the probability of a correct guess (2^{-a}), multiplied with the winning probability of the adaptive extractor adversary. We formalize this proof in the theorem below.

Theorem 1. *Every $(\eta, \mu, d, l, \epsilon)$- average case extractor Ext is an $(\eta, \mu+\zeta, d, l, \epsilon)$- extractor that is $(\mathcal{F}_{a,\zeta}, 2^{a+2}\epsilon)$-adaptive, for any $\mu + \zeta \leq \eta$ and $a \leq l$.*

Proof. For simplicity, we omit the auxiliary information Z, that depends only on the source (and not on the extractor output). Let W be the source of η bits, such that $\mathbf{H}_\infty(W) \geq \mu + \zeta$. Consider $f' \in \mathcal{F}_{a,\zeta}$, with the corresponding functions (f, g) (recall $f'(w, y) = g(w, f(y))$, where f outputs a bits and g outputs ζ bits). To prove adaptive security (definition 3), we need to show that:

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

where Y is the random variable $\mathsf{Ext}(W; U_d)$. Expanding the description of f', this gives:

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

To prove this, we consider the following two sets $\mathcal{B} = \{b : \Pr[f(Y) = b] > 0\}$ and $\mathcal{A} = \{0,1\}^{d+\zeta+l}$. For each $b \in \mathcal{B}$, we begin by using the statistical distance Lemma 3 with random variables A, B and events E, E' set as $(U_d, g(W, f(Y)), Y)$, $(U_d, g(W, f(U_l)), U_l)$, $f(Y) = b$ and $f(U_l) = b$, respectively. By use of law of total probability and Lemma 3, we get:

$$\begin{aligned}
&\mathbf{SD}((U_d, g(W, f(Y)), Y),\ (U_d, g(W, f(U_l)), U_l)) \\
&\quad \leq \Pr[f(U_l) \notin \mathcal{B}] + \sum_{b \in \mathcal{B}} \mathbf{SD}\left(A \wedge E, B \wedge E'\right) \\
&\quad \leq \Pr[f(U_l) \notin \mathcal{B}] + \sum_{b \in \mathcal{B}} ((|\Pr[E] - \Pr[E']|) + \Pr[E'] \cdot \mathbf{SD}\left(A|E, B|E'\right))
\end{aligned}$$

But now, note that, by extractor security, since $Y \approx_\epsilon U_l$, by applying Lemma 2, we have $f(Y) \approx_\epsilon f(U_l)$. Further, by the definition of statistical distance, we have that, for each $b \in \mathcal{B}$, $|\Pr[f(Y) = b] - \Pr[f(U_l) = b]| \leq \epsilon$ and $\Pr[f(U_l) \notin \mathcal{B}] \leq \epsilon$ (since $\Pr[f(Y) \notin \mathcal{B}] = 0$]). Applying this to above inequality, we get:

$$\begin{aligned}
&\mathbf{SD}((U_d, g(W, f(Y)), Y),\ (U_d, g(W, f(U_l)), U_l)) \\
&\quad \leq \epsilon + \sum_{b \in \mathcal{B}} (\epsilon + \Pr[E'] \cdot \mathbf{SD}\left(A|E, B|E'\right)) \\
&\quad = (|\mathcal{B}| + 1)\epsilon + \sum_{b \in \mathcal{B}} \Pr[E'] \cdot \mathbf{SD}(A|E,\ B|E')
\end{aligned}$$

Finally, we apply the statistical distance lemma 4 on the random variables $(A, f(Y))$ and $(B, f(U_l))$ with set $S = \{b\}$. Note that, given events E and E' the value of $f(Y)$ and $f(U_l)$ are fixed to a b, which means the leakage $g(W, b)$ is only a leakage on W. Thus, we can use extractor security to get: $(U_d, g(W, b), Y) \approx_\epsilon (U_d, g(W, b), U_l)$. Hence, applying this to the above inequality, we get:

$$\begin{aligned}
&\mathbf{SD}((U_d, g(W, f(Y)), Y),\ (U_d, g(W, f(U_l)), U_l)) \\
&\quad \leq (|\mathcal{B}| + 1)\epsilon + \sum_{b \in \mathcal{B}} \Pr[E'] \cdot \frac{2\epsilon}{\Pr[f(U_l) = b]} \\
&\quad \leq 4|\mathcal{B}|\epsilon \leq 2^{a+2}\epsilon
\end{aligned}$$

□

Concrete Instantiation. We show that the extractor due to Guruswami *et al.* [23] is an adaptive extractor even when the leakage depends on the entire extractor output. We state the result from [23] below.

Lemma 6. *[23] For every constant $\nu > 0$ all integers $\eta \geq \mu$ and all $\epsilon \geq 0$, there is an explicit (efficient) $(\eta, \mu, d, l, \epsilon)-$strong extractor with $l = (1-\nu)\mu - \mathcal{O}(\log(\eta) + \log(\frac{1}{\epsilon}))$ and $d \leq \mathcal{O}(\log(\eta) + \log(\frac{1}{\epsilon}))$.*

Let Full_ζ $(= \mathcal{F}_{l,\zeta})$, denote the leakage function family which computes leakage (of size ζ) dependent on the entire extractor output and the source. The following lemma shows that one can appropriately set the parameters of the [23] extractor to get negligible error, while extracting a constant fraction of the bits from the source, and while adaptively leaking a constant fraction of bits from it.

Lemma 7. *For all positive integers l, ζ, every constant $\nu > 1$ and $\epsilon \geq 0$, there is an explicit (efficient) $(\eta, \mu+\zeta, d, l, \epsilon)-$extractor that is $(\mathsf{Full}_\zeta, \delta)$-adaptive with $d = \mathcal{O}(\log(\frac{\eta}{\epsilon}))$, $\mu = \nu l + \mathcal{O}(\log(\frac{\eta}{\epsilon}))$, any $\eta \geq \mu + \zeta$ and $\delta = \epsilon \cdot 2^{l+2}$.*
On further implication, for any $c > 1$, there exists constants α, β such that $d \leq \alpha l$, $\mu \leq \beta l$, $\eta \geq \beta l + \zeta$, $\epsilon = 2^{-cl}$ and $\delta = 2^{(1-c)l+2}$ when $l = \omega(\log \eta)$.

Proof. The proof of the first part of the lemma follows directly from Theorem 1 and Lemma 6 and the further implication can be obtained by simple substitution. □

Further, we use the following generalization of adaptive extractors: for an adaptive extractor Ext, if we consider k independent sources $W_1, \cdots, W_k$ and a single seed S, all the extractor outputs $(\mathsf{Ext}(W_i; S))_{i\in[k]}$ look uniform, even given adaptive leakage on each W_i, dependent on not just $\mathsf{Ext}(W_i; S)$ (or uniform), but also all the prior extractor outputs and adaptive leakages (queried before i). As the sources are independent, this lemma can be proved using a simple hybrid argument (the detailed proof is given in Appendix A.1.1).

Lemma 8. *Let k be an arbitrary positive integer, $W_1, \cdots, W_k$ be k independent $(\eta, \mu+\zeta)$ sources and S be the uniform distribution on $\{0,1\}^d$. Let Ext be an $(\eta, \mu+\zeta, d, l, \delta')$-extractor that is $(\mathsf{Full}_\zeta, \delta)$-adaptive. For each $i \in [k]$, let E_i^0 denotes $\mathsf{Ext}(W_i; S)$, E_i^1 denotes uniform distribution on $\{0,1\}^l$. For $b \in \{0,1\}$, we define AdLeak^b as follows. Then for any stateful distinguisher $\mathcal{D}'$ we have $\mathsf{AdLeak}^0 \approx_{k\delta} \mathsf{AdLeak}^1$.*

AdLeak^b :

- *Let Tr and $\mathcal{S}$ be a null string and null set respectively.*
- *For upto k times*
  - *$(j, g_j) \leftarrow \mathcal{D}'(Tr)$ where $j \in [k] \backslash \mathcal{S}$ and $g_j : \{0,1\}^{\eta+l} \to \{0,1\}^\zeta$.*
  - *Append $(j, g_j, g_j(w_j, E_j^b), E_j^b)$ to Tr.*
  - *Add j to $\mathcal{S}$.*
- *Output Tr.*

4 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 $m, m', \mathbf{SD}\left(f(Share(m)), f(Share(m'))\right) \leq \epsilon_l$.*

4.1 Leakage Models

We consider two leakage models in this paper. For now, we restrict our discussion to an (n, t)-threshold access structure.

- **Adaptive Leakage and Reveal Model**: The adversary can adaptively obtain leakage on individual shares for any $n - t + 1$ shares. After this, he can additionally even get all the remaining $t - 1$ shares in their entirety.
- **Joint Leakage and Reveal Model**: The adversary can ask any number of joint leakage queries on disjoint sets of size X (a parameter). After this, he can additionally get any (at most $t - 1$) of the remaining shares in their entirety. While this model completely subsumes the adaptive leakage and reveal model, the amount of leakage per share supported in the latter would be lesser.

We provide a formal description of the adaptive leakage and reveal model and the joint leakage and reveal model in Section 4.1.1 and Section 4.5 respectively. We give a construction that is leakage resilient with respect to both these models in Section 4.2. We prove leakage resilience of this scheme in the adaptive leakage and reveal model in Section 4.3. We provide a proof sketch of leakage resilience in the joint adaptive and reveal model in Section 4.5.2. We also briefly discuss the extension to general access structures in Section 4.6.

4.1.1 Adaptive Leakage and Reveal Model $\mathcal{F}_{leak}^{\psi,\tau}$

The model allows for leakage on individual shares and then also reveals at most $t-1$ of the remaining shares in clear. We have two parameters in the model τ and ψ where τ denotes the amount of leakage provided in each leakage query and ψ captures the maximum number of leakage queries allowed. We allow ψ ranging from 1 to $n - t + 1$. Though we allow ψ to be $n - t + 1$, we have it as an explicit parameter because lower ψ would imply a weaker leakage model and possibly have better constructions. In fact, our multi-layered construction in Sec. 4.2 becomes compact (and offers better rate) as ψ decreases.

Let $(Share, Rec)$ (where $Share : \{0,1\}^l \to (\{0,1\}^\gamma)^n$) be a t-out-of-n secret sharing scheme. We formalize leakage obtained in this model on shares of a message m as $\mathsf{Leak}_{Share}^{m}$ in Figure 2, where an arbitrary stateful distinguisher $\mathcal{D}$ makes the queries. For any two messages m and m', we require $\mathsf{Leak}_{Share}^{m} \approx_{\epsilon_{lr}} \mathsf{Leak}_{Share}^{m'}$, for $(Share, Rec)$ to be ϵ_{lr} leakage resilient against the adaptive leakage and reveal model.

$\mathsf{Leak}_{Share}^{m}$:

- Initialize Z to be a null string and $\mathcal{S}$ to be a null set.

- $(Sh_1, \cdots, Sh_n) \leftarrow Share(m)$
- **Leakage Phase:**

  For upto ψ times

  - $(j, f_j) \leftarrow \mathcal{D}(Z)$ where $f_j : \{0,1\}^\gamma \rightarrow \{0,1\}^\tau$
  - If $j \in [n] \backslash \mathcal{S}$, add j to $\mathcal{S}$ and append $(j, f_j, f_j(Sh_j))$ to Z
- **Reveal phase**

  For upto $t-1$ times

  - $j \leftarrow \mathcal{D}(Z)$
  - If $j \in [n] \backslash \mathcal{S}$, append (j, Sh_j) to Z
- $\mathcal{D}$ updates Z with any relevant state information.
- Output Z.

Figure 2: LRSS Definition- Leak^m_{Share} Distribution

4.2 LRSS Construction for the Adaptive Leakage and Reveal Model

We refer the reader to the Introduction (Section 1.2) for a high-level overview of the construction and proof. We proceed to describe the construction in detail in Figure 3 and prove its security in Section 4.3.

Let n be the number of parties and t be the reconstruction threshold. Let $h > 0$ be a parameter guaranteed to be less than $\lceil n/(t-1) \rceil$.

Building Blocks. Let $(\mathsf{MShare}, \mathsf{MRec})$ be an $((n,t), \varepsilon, \epsilon)$-adaptive secret sharing scheme for messages in $\{0,1\}^l$ with share space being $\{0,1\}^{l'}$. For $i \in [h]$, let $(\mathsf{SdShare}^i, \mathsf{SdRec}^i)$ be an $((n,t), \varepsilon'_i, \epsilon'_i)$-adaptive secret sharing scheme for messages in $\{0,1\}^{d_i}$ with share space being $\{0,1\}^{d'_i}$. For $i \in [h]$, let Ext^i be an $(\eta_i, \mu_i + \tau, d_i, \ell_i, \delta'_i)$-extractor that is $(\mathsf{Full}_\tau, \delta_i)$-adaptive. We set $\ell_1 = l'$ and for $i \in [h] \backslash \{1\}$ we set $l_i = l_{i-1} + d'_{i-1}$.

$\mathsf{Share}^h(m)$:

- $(m_1, \cdots, m_n) \leftarrow \mathsf{MShare}(m)$.
- For $i \in [h]$, pick seeds $s^i \in_R \{0,1\}^{d_i}$ and compute their shares $(sd^i_1, \cdots, sd^i_n) \leftarrow \mathsf{SdShare}^i(s^i)$.
- For $i \in [h]$ and $j \in [n]$, pick sources $w^i_j \in_R \{0,1\}^{\eta_i}$.
- For $j \in [n]$:
  - Define $y^{h+1}_j = m_j$.

- For $i \leftarrow h$ to 1, compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$.

- For $j \in [n]$, define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$.
- Output $(Sh_1, \cdots, Sh_n)$.

$\mathsf{Rec}^h(Sh_T)$: (where T is the reconstruction set)

- For $j \in T$, parse Sh_j as $(w_j^1, \cdots, w_j^h, y_j^1)$, where $y_j^1 = x_j^1 || sd_j^1$.
- For $i \leftarrow 1$ to h:
  - $s^i = \mathsf{SdRec}^i(sd_T^i)$.
  - For each $j \in T$, $y_j^{i+1} = x_j^i \oplus \mathsf{Ext}^i(w_j^i; s^i)$. For each $i \in [h-1]$, parse y_j^{i+1} as $x_j^{i+1} || sd_j^{i+1}$.
- Parse y_j^{h+1} as m_j. Recover $m = \mathsf{MRec}(m_T)$.
- Output m.

Figure 3: LRSS Construction

4.3 Proof of Leakage Resilience in the Adaptive Leakage and Reveal Model

Theorem 2. *For any $\psi \leq n - t + 1$ and $l, \tau > 0$, $(\mathsf{Share}^h, \mathsf{Rec}^h)$ is an $((n,t), \varepsilon)$-secret sharing scheme for l bit messages and is $2(\epsilon + h(\epsilon' + (t-1)\delta))$-leakage resilient in the Adaptive Leakage and Reveal model $\mathcal{F}_{leak}^{\psi,\tau}$ where $h = \lceil \psi/(t-1) \rceil$.*
Further, there exists an instantiation of the scheme with rate is $(2^{\Theta(h)} + h\tau/l)^{-1}$. When $\tau = \Theta(l)$ and either $n = \Theta(t)$ or h is a constant, the scheme achieves constant rate and constant leakage rate asymptotically.

Proof. The correctness of the scheme follows directly from the correctness of underlying extractors and secret sharing schemes. The (adaptive) privacy of the scheme is directly implied by the leakage resilience (against the adaptive leakage and reveal model).

Leakage Resilience. For any message m we define the following the sequence of hybrids. In these hybrids we assume that $\mathcal{D}$ always asks legitimate queries as per the model and won't write explicit checks for legitimacy (for example, we assume that $\mathcal{D}$ doesn't ask leakage on same share twice).
We analyze the leakage queries made by $\mathcal{D}$ as bunches of $(t-1)$ queries. We now introduce some useful notation. Let $\mathcal{S}_1, \cdots, \mathcal{S}_h$ denote the sets of indices queried by $\mathcal{D}$, where $\mathcal{S}_i$ contains the indices queried by $\mathcal{D}$ from the $((i-1)(t-1)+1)^{th}$ query to $i(t-1)^{th}$ leakage queries (i.e., $\mathcal{S}_1$ contains the first $t-1$ queries, S_2 the next $t-1$ queries and so on). For $i \in [h]$, we use $\mathcal{S}_{[i]}$ to denote $\bigcup_{j=1}^{i} \mathcal{S}_j$, which captures the set of indices queried in the first $i(t-1)$ leakage queries. For $i \in [h]$, let $Z_{[i]}$ denotes the set of leakage queries and the corresponding responses to the first $i(t-1)$ leakage queries. $Z_{[h+1]}$ denotes $Z_{[h]}$ together with the final reveal queries as well as any relevant state information. We prove leakage resilience using a hybrid argument, with the following sequence

of hybrids, LeakB_0^m, $\{\mathsf{LeakA}_q^m, \mathsf{LeakB}_q^m\}_{q\in[h]}$ and LeakC^m. The order of the hybrids is LeakB_0^m, LeakA_1^m, $\mathsf{LeakB}_1^m, \cdots, \mathsf{LeakA}_h^m$, LeakB_h^m, LeakC^m, where we will show that LeakC^m is independent of m, and LeakB_0^m will correspond to the distribution $\mathsf{Leak}^m_{\mathsf{Share}^h}$. This will allow us to show that $\mathsf{Leak}^m_{\mathsf{Share}^h}$ is indistinguishable from $\mathsf{Leak}^{m'}_{\mathsf{Share}^h}$. We begin by giving an informal description of these hybrids.

$\underline{\mathsf{LeakA}_q^m}$**:** We start with $q = 1$. LeakA_1^m follows the actual leakage game i.e., $\mathsf{Leak}^m_{Share^h}(\equiv \mathsf{LeakB}_0^m)$ except for the following change: we replace the shares sd_j^1, for each $j \in \mathcal{S}_1$ (the shares of s^1 corresponding to the first $t-1$ leakage queries), with shares of a dummy seed $\widetilde{s}^1 = 0^d$. In general, for each $1 < q \leq h$, the only change we make in LeakA_q^m (in comparison to the previous hybrid LeakB_{q-1}^m) is that we replace the shares sd_j^q, for each $j \in \mathcal{S}_q$ (the shares of s^q corresponding to the q-th set of $t-1$ leakage queries), with shares of a dummy seed $\widetilde{s}^q$. After answering the leakage queries corresponding to $\mathcal{S}_q$, shares of s^q are re-sampled consistent with the dummy seed shares used so far. The hybrid is formally described in Figure 4.

$\underline{\mathsf{LeakB}_q^m}$**:** For $q = 1$, LeakB_1^m follows the hybrid LeakA_1^m except for the following change: in LeakB_1^m, we replace the values x_j^1, for each $j \in \mathcal{S}_1$ with random, instead of evaluating the h layers of masking to get x_j^1 (and hence x_j^1's for $j \in \mathcal{S}_1$ are independent of $m_{\mathcal{S}_1}$, s^i and the shares of s^i, for each $1 < i \leq h$). Note that in LeakA_1^m, the shares Sh_j corresponding to $\mathcal{S}_1$ no longer depend on the seed s^1. We carefully use the adaptive extractor security of Ext^1 to move to LeakB_1^m. In general, for each $1 < q \leq h$, the only change we make in LeakB_q^m (in comparison to the previous hybrid LeakA_q^m) is that we replace the values x_j^q, for each $j \in \mathcal{S}_q$ with random, instead of evaluating the $h-(q-1)$ layers of masking to get x_j^q (and hence, for these queries in $\mathcal{S}_q$, s^i and the shares of s^i, for each $q < i \leq h$, and the shares m are not used to evaluate x_j^q). Further, we continue the steps of masking to evaluate $x_j^{q-1}, x_j^{q-2}, \cdots, x_j^1$, for each $j \in \mathcal{S}_q$ as in the previous hybrid. The hybrid is formally described in Figure 5.

$\underline{\mathsf{LeakC}^m}$**:** In the hybrid LeakB_h^m, all the shares used in the leakage phase are independent of the shares of the message m. Hence, the only part of the view of $\mathcal{D}$ that depends on the shares of m corresponds to the reveal phase. In the final hybrid LeakC^m, we replace the $t-1$ shares of m used in the reveal phase by shares of 0^l. This hybrid is formally described in Figure 6.
The formal descriptions of all hybrids are given below with the change from the prior hybrid highlighted in red color.

LeakA_q^m:

1. Initialize Z to be a null string and $\mathcal{S}_1, \cdots, \mathcal{S}_h$ to be null sets.
2. $(m_1, \cdots, m_n) \leftarrow \mathsf{MShare}(m)$
3. For $i \in [h]$, choose $s^i \in_R \{0,1\}^{d_i}$
4. For $i \in [h]$ and $j \in [n]$, choose $w_j^i \in_R \{0,1\}^{\eta_i}$
5. For $i \in [h]\backslash[q]$, compute $(sd_1^i, \cdots, sd_n^i) \leftarrow \mathsf{SdShare}^i(s^i)$

6. For $i \in [q]$, let $\widetilde{s}^i = 0^d$
7. For $j \in [n]$, define $y_j^{h+1} = m_j$
8. **Leakage Phase:**
   (a) For $c \leftarrow 1$ to q
      i. $(\widetilde{sd}_1^c, \cdots, \widetilde{sd}_n^c) \leftarrow \mathsf{SdShare}^c(\widetilde{s}^c)$
      ii. For up to $(t-1)$ times
         A. $(j, f_j) \leftarrow \mathcal{D}(Z)$
         B. If $c < q$,
            * Choose $x_j^c \in_R \{0,1\}^{l_c}$ and compute $y_j^c = (x_j^c || \widetilde{sd}_j^c)$
            * For $i \leftarrow c-1$ down to 1,
              compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
         C. If $c = q$, for $i \leftarrow h$ down to 1 compute
            $\begin{cases} x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i) \text{ and } y_j^i = (x_j^i || sd_j^i) \text{ when } i \neq q \\ x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i) \text{ and } y_j^i = (x_j^i || \widetilde{sd}_j^i) \text{ when } i = q \end{cases}$
         D. Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
         E. Add j to $\mathcal{S}_c$ and append $(j, f_j, f_j(Sh_j))$ to Z
      iii. $(sd_1^c, \cdots, sd_n^c) \leftarrow \mathsf{SdShare}^c(s^c | \widetilde{sd}_{\mathcal{S}_c}^c)$
   (b) For $j \in [n] \backslash (\mathcal{S}_{[q]})$ and $i \leftarrow h$ down to 1,
   compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
   (c) Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
   (d) For $c \leftarrow q+1$ to h
      i. For upto $t-1$ times
         A. $(j, f_j) \leftarrow \mathcal{D}(Z)$
         B. Add j to $\mathcal{S}_c$ and append $(j, f_j, f_j(Sh_j))$ to Z
9. **Reveal phase**
   (a) For upto $t-1$ times
      i. $j \leftarrow \mathcal{D}(Z)$
      ii. Append (j, Sh_j) to Z
10. $\mathcal{D}$ updates Z with any relevant state information.
11. Output Z.

Figure 4: Hybrid LeakA_q^m

LeakB_q^m

1. Initialize Z to be a null string and $\mathcal{S}_1, \cdots, \mathcal{S}_h$ to be null sets.

2. $(m_1, \cdots, m_n) \leftarrow \mathsf{MShare}(m)$
3. For $i \in [h]$, choose $s^i \in_R \{0,1\}^{d_i}$
4. For $i \in [h]$ and $j \in [n]$, choose $w_j^i \in_R \{0,1\}^{\eta_i}$
5. For $i \in [h] \backslash [q]$, compute $(sd_1^i, \cdots, sd_n^i) \leftarrow \mathsf{SdShare}^i(s^i)$
6. For $i \in [q]$, let $\widetilde{s}^i = 0^d$
7. For $j \in [n]$, define $y_j^{h+1} = m_j$
8. **Leakage Phase:**
   (a) For $c \leftarrow 1$ to q
      i. $(\widetilde{sd}_1^c, \cdots, \widetilde{sd}_n^c) \leftarrow \mathsf{SdShare}^c(\widetilde{s}^c)$
      ii. For upto $(t-1)$ times
         A. $(j, f_j) \leftarrow \mathcal{D}(Z)$
         B. Choose $x_j^c \in_R \{0,1\}^{l_c}$ and compute $y_j^c = (x_j^c || \widetilde{sd}_j^c)$
         C. For $i \leftarrow c-1$ down to 1
            compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
         D. Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
         E. Add j to $\mathcal{S}_c$ and append $(j, f_j, f_j(Sh_j))$ to Z
      iii. $(sd_1^c, \cdots, sd_n^c) \leftarrow \mathsf{SdShare}^c(s^c | \widetilde{sd}_{\mathcal{S}_c}^c)$
   (b) For $j \in [n] \backslash \mathcal{S}_{[q]}$ and $i \leftarrow h$ to 1, ($\mathcal{S}_{[q]}$ denotes a null set when $q = 0$)
   compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
   (c) Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
   (d) For $c \leftarrow q+1$ to h
      i. For upto $t-1$ times
         A. $(j, f_j) \leftarrow \mathcal{D}(Z)$
         B. Add j to $\mathcal{S}_c$ and append $(j, f_j, f_j(Sh_j))$ to Z
9. **Reveal phase**
   (a) For upto $t-1$ times
      i. $j \leftarrow \mathcal{D}(Z)$
      ii. Append (j, Sh_j) to Z
10. $\mathcal{D}$ updates Z with any relevant state information.
11. Output Z.

Figure 5: Hybrid LeakB_q^m

LeakC^m

1. Initialize Z to be a null string and $\mathcal{S}_1, \cdots, \mathcal{S}_h$ to be null sets.
2. Let $\tilde{m} = 0^l$ and $(\tilde{m}_1, \cdots, \tilde{m}_n) \leftarrow \mathsf{MShare}(\tilde{m})$
3. For $i \in [h]$, choose $s^i \in_R \{0,1\}^{d_i}$
4. For $i \in [h]$, let $\widetilde{s}^i = 0^d$
5. For $i \in [h]$ and $j \in [n]$, choose $w_j^i \in_R \{0,1\}^{\eta_i}$
6. **Leakage Phase:**
   (a) For $c \leftarrow 1$ to h
      i. $(\widetilde{sd}_1^c, \cdots, \widetilde{sd}_n^c) \leftarrow \mathsf{SdShare}^c(\widetilde{s}^c)$
      ii. For upto $(t-1)$ times
         A. $(j, f_j) \leftarrow \mathcal{D}(Z)$
         B. Choose $x_j^c \in_R \{0,1\}^{l_c}$ and compute $y_j^c = (x_j^c || \widetilde{sd}_j^c)$
         C. For $i \leftarrow c-1$ down to 1
            compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
         D. Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
         E. Add j to $\mathcal{S}_c$ and append $(j, f_j, f_j(Sh_j))$ to Z
      iii. $(sd_1^c, \cdots, sd_n^c) \leftarrow \mathsf{SdShare}^c(s^c | \widetilde{sd}_{\mathcal{S}_c}^c)$
7. **Reveal phase**
   (a) For upto $t-1$ times
      i. $j \leftarrow \mathcal{D}(Z)$
      ii. Define $y_j^{h+1} = \tilde{m}_j$
      iii. For $i \leftarrow h$ to 1, compute $x_j^i = y_j^{i+1} \oplus \mathsf{Ext}^i(w_j^i; s^i)$ and $y_j^i = (x_j^i || sd_j^i)$
      iv. Define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$
      v. Append (j, Sh_j) to Z
8. $\mathcal{D}$ updates Z with any relevant state information.
9. Output Z.

Figure 6: Hybrid LeakC^m

We begin by proving the statistical closeness of LeakA_q^m and LeakB_{q-1}^m, for each $q \in [h]$, which follows from adaptive privacy of $\mathsf{SdShare}^q$, as atmost only $t-1$ dummy seed shares are used.

Claim 1. *For $q \in [h]$, if $\mathsf{SdShare}^q$ is ϵ'_q-adaptively private against (n,t)-threshold access structures, then $\mathsf{LeakA}_q^m \approx_{\epsilon'_q} \mathsf{LeakB}_{q-1}^m$*

Proof. **Answering the first $(q-1)$ sets of leakage queries (when $q > 1$):** Observe that the

hybrids are identical up to answering the first $(q-1)(t-1)$ leakage queries and differ in answering the remaining queries. For any $k \in [q-1]$ and, $j \in \mathcal{S}_k$ the leakage response only depends on $\widetilde{sd}_j^k$, $w_j^1, \cdots, w_j^h$ and $\{s^i, sd_j^i\}_{1 \le i < k}$ (as x_j^k is chosen uniformly). We let Pre denote the union of these random variables upon which the leakage responses to $j \in \mathcal{S}_{[q-1]}$ depend.

Answering the q^{th} set of leakage queries: Consider $j \in \mathcal{S}_q$. To answer this leakage query, it suffices to compute $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$. The hybrids only differ in computation of y_j^1 (particularly in computation of y_j^q, which is used to compute y_j^1) and the distribution of extractor sources is identical in both. We highlight the differences here. LeakA_q^m (Step 8-(a)-ii-C), iteratively computes $y_j^h, \cdots, y_j^q, \cdots, y_j^1$ as follows.

- $(y_j^h, \cdots, y_j^{q+1})$ are computed using y_j^{h+1} and $\{w_j^i, sd_j^i, s^i\}_{i \in [h] \setminus [q]}$. Note that the distribution of $y_j^h, \cdots, y_j^{q+1}$ is identical in both hybrids.
- x_j^q is computed using y_j^{q+1}, w^q and s^q. x_j^q is also identical in both hybrids.
- y_j^q is computed as $x_j^q || \widetilde{sd}_j^q$ (where $\widetilde{sd}_{[n]}^q$ are shares of a dummy seed $\widetilde{s}^q$ which are generated before answering any queries in $\mathcal{S}_q$ in Step 8-(a)-i (when $c = q$)). Whereas in LeakB_{q-1}^m, $y_j^q = x_j^q || sd_j^q$ (where $sd_{[n]}^q$ are shares of s^q)
- $(y_j^{q-1}, \cdots, y_j^1)$ are computed using y_j^q and $\{sd_j^i, w_j^i, s^i\}_{i \in [q-1]}$. The computation of $(y_j^{q-1}, \cdots, y_j^1)$ given the later random variables is again identical to LeakB_{q-1}^m.
- Now LeakA_q^m defines $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$

For convenience, in this proof we distinguish (whenever necessary) the random variables that have same literal in both the hybrids but are distributionally different with subscripts A and B respectively. For example, $y_{j,A}^q$ and $y_{j,B}^q$ denote the distributions of y_j^q in LeakA_q^m and, LeakB_{q-1}^m respectively.

Let $\mathsf{Pre}' = (\{w_j^q, \{sd_j^i, w_j^i, s^i\}_{i \in [h] \setminus \{q\}}\}_{j \in [n] \setminus \mathcal{S}_{[q-1]}})$. Pre' captures the information required to answer all queries after the first $q-1$ sets of leakage queries, except for any information regarding $s^q, \widetilde{s}^q$ and their shares. Note that Pre' is identical in both hybrids[9]. Since, $|\mathcal{S}_q| \le t-1$, with a reduction to adaptive privacy of $\mathsf{SdShare}^q$ we have

$$\mathsf{Pre}, \mathsf{Pre}', s^q, \widetilde{s}^q, \{\widetilde{sd}_j^q\}_{j \in \mathcal{S}_{q,A}} \approx_{\epsilon'_q} \mathsf{Pre}, \mathsf{Pre}', s^q, \widetilde{s}^q, \{sd_j^q\}_{j \in \mathcal{S}_{q,B}}$$

as $(\mathsf{Pre}, \mathsf{Pre}')$ is independent of the randomness used to generate the shares of $\widetilde{s}^q$ and s^q. Note that the information on LHS suffices to answer the first q sets of queries as per LeakA_q^m. Similarly, RHS suffices to answer queries in $\mathcal{S}_{[q]}$ as per LeakB_{q-1}^m. Therefore, we have,

$$\mathsf{Pre}, \mathsf{Pre}', s^q, \tilde{s}^q, \{\widetilde{sd}_j^q\}_{j \in \mathcal{S}_{q,A}}, Z_{[q],A} \approx_{\epsilon'_q} \mathsf{Pre}, \mathsf{Pre}', s^q, \tilde{s}^q, \{sd_j^q\}_{j \in \mathcal{S}_{q,B}}, Z_{[q],B} \tag{1}$$

Answering the leakage and reveal queries made after the q^{th} set of leakage queries: After all the q^{th} set leakage queries are answered, LeakA_q^m computes $(sd_1^q, \cdots, sd_n^q) \leftarrow \mathsf{SdShare}^q(s^q | \widetilde{sd}_{\mathcal{S}_{q,A}}^q)$.

[9] Pre' possibly repeats some information already there in Pre. For example for $q = 2$, s^1 is there in both Pre and Pre'. It is for the ease of exposition that we have this repetition.

Given $(sd_1^q, \cdots, sd_n^q), s^q, \mathsf{Pre}$ and Pre', for any $j \in [n] \backslash \mathcal{S}_q$, Sh_j is easily computed (Steps 8-(b) and 8-(c)). With this, any further queries can be correctly answered as per LeakA_q^m. Let $(\widehat{sd}_1^q, \cdots, \widehat{sd}_n^q) \leftarrow \mathsf{SdShare}^q(s^q | sd^q_{\mathcal{S}_{q,B}})$. By Lemma 2, we have

$$\mathsf{Pre}, \mathsf{Pre}', s^q, \tilde{s}^q, Z_{[q],A}, sd^q_{[n],A} \approx_{\epsilon'_q} \mathsf{Pre}, \mathsf{Pre}', s^q, \tilde{s}^q, Z_{[q],B}, \widehat{sd}^q_{[n],B}$$

Note that $\widehat{sd}^q_{[n]}$ is identical to $sd^q_{[n],B}$ (of LeakB^m_{q-1}) even given s^q and $\{sd_j^q\}_{j \in \mathcal{S}_q}$ by the property of consistent resampling in Claim 5. Therefore, we have,

$$\mathsf{Pre}, \mathsf{Pre}', s^q, Z_{[q],A}, sd^q_{[n],A} \approx_{\epsilon'_q} \mathsf{Pre}, \mathsf{Pre}', s^q, Z_{[q],B}, sd^q_{[n],B}$$

Since the above LHS and RHS are sufficient to answer any further queries, we have

$$Z_{[h+1],A} \approx_{\epsilon'_q} Z_{[h+1],B}$$

which proves the claim. □

Now, we prove the statistical closeness of LeakA_q^m and LeakB_q^m, for each $q \in [h]$ using the adaptive extractor security. The high-level idea behind the reduction is that in hybrid LeakA_q^m, the shares corresponding to the first $q(t-1)$ queries (i.e., $\mathcal{S}_{[q]}$) no longer depend on the seed s^q and hence, we can use the adaptive extractor security of Ext^q to move to LeakB_q^m.

Claim 2. *For $q \in [h]$, if Ext^q is an $(\eta_q, \mu_q + \tau, d_q, l_q, \delta'_q)$- extractor that is $(\mathsf{Full}_\tau, \delta_q)$-adaptive, then $\mathsf{LeakA}_q^m \approx_{(t-1)\delta_q} \mathsf{LeakB}_q^m$*

Proof. Observe that the hybrids are identical up to answering the first $(q-1)(t-1)$ leakage queries and differ in answering the q^{th} set of queries. Further, after answering the q^{th} set of leakage queries, the responses to all remaining leakage/reveal queries are answered identically in both hybrids.

Answering the first $(q-1)$ sets of leakage queries (when $q > 1$):
For any $k \in [q-1]$ and $j \in \mathcal{S}_k$ the leakage response only depends on $\widetilde{sd}_j^k$, $w_j^1, \cdots, w_j^h$, $\{s^i, sd_j^i\}_{1 \le i < k}$ and x_j^k, where the latter is uniformly chosen. We let Pre denote the leakage responses $Z_{[q-1]}$ and the union of these random variables upon which the leakage responses to $j \in \mathcal{S}_{[q-1]}$ depend.
Answering the q^{th} set of leakage queries:
Consider $j \in \mathcal{S}_q$ and f_j be the corresponding leakage function. To answer this leakage query, we require computing $f_j(Sh_j)$ where $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$. The hybrids only differ in computation of y_j^1 (particularly in computation of x_j^q, which is used to compute y_j^1) and the distribution of extractor sources is identical in both. The hybrids iteratively computes $y_j^q, \cdots, y_j^1$ as follows.

- x_j^q is chosen uniformly from $\{0,1\}^{l_q}$ in LeakB_q^m. In contrast, x_j^q of LeakA_q^m depended on $\mathsf{Ext}^q(w_j^q; s^q)$ and y_j^{q+1}.
- $(y_j^q, \cdots, y_j^1)$ is determined given $x_j^q, \widetilde{sd}_j^q$ and $\{sd_j^i, w_j^i, s^i\}_{i \in [q-1]}$ in both the hybrids.
- Both hybrids define $Sh_j = (w_j^1, \cdots, w_j^h, y_j^1)$

Let $\mathsf{Pre}' = \{w_j^i, sd_j^i, s^i, y_j^{h+1}, \widetilde{sd}_j^q\}_{i\in[h]\setminus\{q\}, j\in[n]\setminus\mathcal{S}_{[q-1]}}$.We capture Pre' as the information which along with $\{w_j^q, s^q\}_{j\in\mathcal{S}_q}$ is sufficient to answer any leakage queries on $j \in \mathcal{S}_q$. Also, Pre' is identical in both hybrids.

Let $j_1, \cdots, j_{t-1}$ be the order of indices in which leakage queries are made in $\mathcal{S}_q$. Firstly, we prove that $(\mathsf{Pre}, \mathsf{Pre}', f_{j_1}(Sh_{j_1}))$ of both hybrids are statistically close. After that we proceed to show that $(\mathsf{Pre}, \mathsf{Pre}', f_{j_1}(Sh_{j_1}), \cdots, f_{j_{(t-1)}}(Sh_{j_{(t-1)}}))$ of both the hybrids are statistically close, which implies that the hybrids are statistically close up to answering first q sets of queries. For convenience, in this proof we distinguish (whenever necessary) the random variables that have same literal in the hybrids but are distributionally different with subscripts A and B respectively. For example, $x_{j,A}^q$ and $x_{j,B}^q$ denote the distributions of x_j^q in LeakA_q^m and LeakB_q^m respectively.

Firstly, in both hybrids the distribution of (j_1, f_{j_1}) only depends on $Z_{[q-1]}$ (and any internal randomness of $\mathcal{D}$) and hence are identical. Note that given Pre', $f_{j_1}(Sh_{j_1})$ in LeakA_q^m, can be captured as Full_τ-adaptive leakage on the extractor source $w_{j_1}^q$ and $(x_{j_1,A}^q =)\ \mathsf{Ext}^q(w_{j_1}^q; s^q) \oplus y_{j_1}^{q+1}$. This is because $(y_{j_1}^{q+1}, \mathsf{Pre}')$ are independent of $(w_{j_1}^q, s^q)$. Let g_1 be a function that takes $\mathsf{Pre}', w_{j_1}^q$ and $x_{j_1,A}^q$(or $x_{j_1,B}^q$) as input, computes $y_{j_1,A}^1$ (or $y_{j_1,B}^1$) and outputs $f_j(w_{j_1}^1, \cdots, w_{j_1}^h, y_{j_1,A}^1)$ (or $f_j(w_{j_1}^1, \cdots, w_{j_1}^h, y_{j_1,B}^1)$). With a reduction to adaptive security of Ext^q we have

$$\begin{aligned}
&\mathsf{Pre}, \mathsf{Pre}', s^q, g_1(\mathsf{Pre}', w_{j_1}^q, \mathsf{Ext}^q(w_{j_1}^q; s^q) \oplus y_j^{q+1}) \\
&\approx_{\delta_q} \mathsf{Pre}, \mathsf{Pre}', s^q, g_1(\mathsf{Pre}', w_{j_1}^q, U_{l_q} \oplus y_j^{q+1}) \\
&\equiv \mathsf{Pre}, \mathsf{Pre}', s^q, g_1(\mathsf{Pre}', w_{j_1}^q, x_{j_1,B}^q)
\end{aligned}$$

Therefore

$$\mathsf{Pre}, \mathsf{Pre}', s^q, f_{j_1}(Sh_{j_1,A}) \approx_{\delta_q} \mathsf{Pre}, \mathsf{Pre}', s^q, f_{j_1}(Sh_{j_1,B})$$

With this, we showed that the hybrids are statistically close up to responding to the first query in the q^{th} set. Although, superficially, it may seem that all the leakage responses corresponding to $j \in \mathcal{S}_q$ can be captured as adaptive extractor leakage on the source w_j^q, but it's not the case because of the following subtlety. The extractor sources used in each query are independent of each other, but the seed is the same. For example, one cannot directly capture $f_{j_2}(Sh_{j_2})$ as Full_τ-adaptive leakage (as we did with $f_{j_1}(Sh_{j_1})$). This is because the choice of j_2, f_{j_2} depends on $f_{j_1}(Sh_{j_1})$ which in turn depends on $\mathsf{Ext}^q(w_j^q; s^q)$, and hence is not independent of the seed s^q. We observe in Lemma 8 that adaptive extractors allow us to handle even such (stronger) form of adaptive leakages across different sources with same seed.

Proceeding, with a reduction to Lemma 8 with $k = (t-1), \{W_i = W_{j_i}^q : i \in [k]\}$, $S = s^q$ and $\mathsf{Ext} = \mathsf{Ext}^q$ and the i^{th} leakage function being g_i such that g_i (hardwired with $\mathsf{Pre}', y_{j_i}^{q+1}$) takes $w_{j_i}^q$ and $\mathsf{Ext}^q(w_{j_i}^q; s^q)$(resp. U_{l_q}) as input, computes $y_{j_i,A}^1$ (resp. $y_{j_i,B}^1$) and outputs $f_{j_i}(w_{j_i}^1, \cdots, w_{j_i}^h, y_{j_i,A}^1)$ (resp. $f_{j_i}(w_{j_i}^1, \cdots, w_{j_i}^h, y_{j_i,B}^1)$).

$$\mathsf{Pre}, \mathsf{Pre}', s^q, \{f_{j_i}, f_{j_i}(Sh_{j_i,A})\}_{j_i\in\mathcal{S}_{q,A}}, \mathcal{S}_{q,A}$$

$$\approx_{(t-1)\delta_q} \mathsf{Pre}, \mathsf{Pre}', s^q, \{f_{j_i}, f_{j_i}(Sh_{j_i,B})\}_{j_i\in\mathcal{S}_{q,B}}, \mathcal{S}_{q,B}$$

This shows that the hybrids are statistically close up to answering the first q sets of leakage queries.

Answering the leakage and reveal queries made after the q^{th} set of leakage queries: After all the q^{th} set of leakage queries are answered, both hybrids compute $(sd_1^q, \cdots, sd_n^q) \leftarrow$

$\mathsf{SdShare}(s^q|\widetilde{sd}^q_{\mathcal{S}_q})$. Let $\mathsf{Pre}'' = \{w^q_j, sd^q_j, s^q\}_{j\in[n]\setminus\mathcal{S}_q}$. Note that Pre' in conjunction with Pre'' completely defines Sh_j for any $j \in [n]\setminus\mathcal{S}_{[q]}$. Since Pre'' corresponding to LeakA^m_q(resp. LeakB^m_q) is only correlated to $\mathcal{S}_q, s^q$ and $\widetilde{sd}^q_{\mathcal{S}_q}$(which is in Pre') of the respective hybrids, we have

$$\mathsf{Pre}, \mathsf{Pre}', \mathsf{Pre}''_A, s^q, \{f_{j_i}, f_{j_i}(Sh_{j_i,A})\}_{j_i\in\mathcal{S}_{q,A}}, \mathcal{S}_{q,A}$$

$$\approx_{(t-1)\delta_q} \mathsf{Pre}, \mathsf{Pre}', \mathsf{Pre}''_B, s^q, \{f_{j_i}, f_{j_i}(Sh_{j_i,B})\}_{j_i\in\mathcal{S}_{q,B}}, \mathcal{S}_{q,B}$$

Since responses to leakage/reveal queries after the q^{th} set are can be derived from the LHS and RHS respectively depending on the hybrid, we have

$$Z_{[h+1],A} \approx_{(t-1)\delta_q} Z_{[h+1],B}$$

This proves the claim. □

Finally, we use the adaptive security of MShare to show that LeakC^m is statistically close to LeakB^m_h.

Claim 3. *If* MShare *is* ϵ*-adaptively private against* (n, t)*-threshold access structures, then* $\mathsf{LeakC}^m \approx_\epsilon \mathsf{LeakB}^m_h$

Proof. The hybrids answer the leakage queries identically and differ only in answering the reveal queries.

Answering the leakage queries:

For any $k \in [h]$ and $j \in \mathcal{S}_k$ the leakage response only depends on $\widetilde{sd}^k_j$, $w^1_j, \cdots, w^h_j$, $\{s^i, sd^i_j\}_{1\le i<k}$ and x^k_j, where the latter is uniformly chosen. We let Pre denote the leakage responses $Z_{[h]}$ and the union of these random variables upon which the leakage responses to $j \in \mathcal{S}_{[h]}$ depend.

Answering the reveal queries: Let $\mathsf{Pre}' = \{w^i_j, sd^i_j, s^i\}_{i\in[h],j\in[n]\setminus\mathcal{S}_{[h]}}$. Note that given y^{h+1}_j for all j queried in the reveal phase, $(\mathsf{Pre}, \mathsf{Pre}')$ has sufficient information to answer all the reveal queries.

- LeakB^m_h samples $(m_1, \cdots, m_n) \leftarrow \mathsf{MShare}(m)$ and sets $y^{h+1}_j = m_j$ for all j queried in the reveal phase.
- LeakC^m samples $(\tilde{m}_0, \cdots, \tilde{m}) \leftarrow \mathsf{MShare}(\tilde{m})$ and sets $y^{h+1}_j = \tilde{m}_j$ for all j queried in the reveal phase.

Let $\mathsf{RevealB}$ and $\mathsf{RevealC}$ denote the sets of indices queried in the reveal phase of LeakB^m_h and LeakC^m respectively. As reveal queries are at most $t-1$ in number, we now invoke adaptive privacy of MShare and get

$$\mathsf{Pre}, \mathsf{Pre}', \tilde{m}, m, \{m_j\}_{j\in\mathsf{RevealB}} \approx_\epsilon \mathsf{Pre}, \mathsf{Pre}', \tilde{m}, m, \{\tilde{m}_j\}_{j\in\mathsf{RevealC}}$$

Note that $(\mathsf{Pre}, \mathsf{Pre}')$ is independent of the randomness used in generating shares of m and $\tilde{m}$, therefore adaptive privacy of MShare can be invoked even given these random variables.

Since Sh_j for j queried in reveal phase of LeakB^m_h (resp. LeakC^m) is determined by the above LHS (resp. RHS) we have

$$\underbrace{Z_{[h+1]}}_{of\ \mathsf{LeakB}^m_q} \approx_\epsilon \underbrace{Z_{[h+1]}}_{of\ \mathsf{LeakC}^m}$$

□

With the above claims and use of triangle inequality we know that for any message m, $\mathsf{Leak}^m_{Share^h} \approx_{\epsilon+\sum_{i\in[h]}((t-1)\delta}$ LeakC^m. Note that the description of LeakC^m is independent of m. Hence for any message $m \neq m'$, we have $\mathsf{LeakC}^m \equiv \mathsf{LeakC}^{m'}$. Since, $\mathsf{Leak}^{m'}_{Share^h} \approx_{h\epsilon'+h(t-1)\delta+\epsilon} \mathsf{LeakC}^{m'}$ we get

$$\mathsf{Leak}^m_{Share^h} \approx_{2\epsilon+2\sum_{i\in[h]}((t-1)\delta_i+\epsilon'_i)} \mathsf{Leak}^{m'}_{Share^h}$$

□

4.4 Parameters

For $i \in [h]$, we instantiate $\mathsf{SdShare}^i$ on seeds of length d_i with the (adaptively) private Shamir secret sharing scheme, which results in individual seed share length being d_i. We instantiate MShare on messages of length l_i with the (adaptively) private Shamir secret sharing scheme, which results in individual seed share length being l_i.

Recall Lemma 7 which states that for any $c > 1$, there exists constants α, β such that $d \leq \alpha l$, $\mu \leq \beta l$, $\eta \geq \beta l + \tau$, $\epsilon = 2^{-cl}$ and $\delta = 2^{-(c-1)l+2}$ when $l = \omega(\log \eta)$. Fix any $c > 1$, and constants α, β corresponding to this c given by Lemma 7. For each $i \in [h]$, we instantiate $(\eta_i, \mu_i + \tau, d_i, l_i, \delta'_i)$-extractor Ext^i that is $(\mathsf{Full}_\tau, \delta_i)$-adaptive as per this lemma as follows.

- We set $l_1 = l$, $\delta'_1 = 2^{-cl}$, $\delta_1 = 2^{-\Omega(l)}$, $d_1 \leq \alpha l_1$, $\mu_1 \leq \beta l_1$ and $\eta_1 = \beta l_1 + \tau$.
- For $i > 1$, we set $l_i = l_{i-1} + d_{i-1}$, $\delta'_i = 2^{-cl_i}$, $\delta_i = 2^{-\Omega(l_i)}$, $d_i \leq \alpha l_i$, $\mu_i \leq \beta l_i$ and $\eta_i = \beta l_i + \tau$.

With this setting, individual share length of Share^h is $l_h + d_h + \sum_{i\in[h]} \eta_i = h\tau + \Theta((1+\alpha)^h l)$. Therefore, Share^h acheives constant rate and constant leakage rate whenever $\tau = \mathcal{O}(l)$ and either $n = \Theta(t)$ or h is a constant.

As our instantiations of $\mathsf{SdShare}^i$'s and MShare are perfectly adaptively private, we have Share^h to be a perfectly adaptively private secret sharing scheme which is $t \cdot 2^{-\Omega(l)}$-leakage resilient against the adaptive leakage and reveal model.

4.5 LRSS for Joint Leakage and Reveal Model

4.5.1 Joint Leakage and Reveal Model $\mathcal{J}^{X,\psi,\tau}$

The model allows for ψ number of joint leakage queries on disjoint sets where each query depends on X number of shares and additionally also reveals $t-1$ of the remaining shares (on which leakage isn't queried) in clear. The parameter τ captures the amount of leakage provided in each leakage query.

Let $(Share, Rec)$ (where $Share : \{0,1\}^l \to (\{0,1\}^\gamma)^n$) be a secret sharing scheme for an (n,t)-threshold access structure. We formalize leakage obtained in this model on shares of a message m as JLeak^m_{Share} in Figure 7, where an arbitrary stateful distinguisher $\mathcal{D}$ makes the queries. For any two messages m and m', we require $\mathsf{JLeak}^m_{Share} \approx_{\epsilon_{lr}} \mathsf{JLeak}^{m'}_{Share}$, for $(Share, Rec)$ to be ϵ_{lr} leakage resilient against this model.

JLeak^m_{Share}:

- Initialize Z be a null string and $\mathcal{S}$ to be a null set.
- $(Sh_1, \cdots, Sh_n) \leftarrow Share(m)$

- **Leakage Phase:**

  For upto ψ times

  - $(Q_j, f_j) \leftarrow \mathcal{D}(Z)$ where $Q_j \subseteq [n]$ and $f_j : \{0,1\}^{|Q_j|\gamma} \rightarrow \{0,1\}^{\tau}$
  - If $Q_j \in [n] \backslash \mathcal{S}$ and $|Q_j| \leq X$,
    add elements of Q_j to $\mathcal{S}$ and append $(Q_j, f_j, f_j(Sh_{Q_j}))$ to Z

- **Reveal phase**

  For upto $t-1$ times

  - $j \leftarrow \mathcal{D}(Z)$
  - If $j \in [n] \backslash \mathcal{S}$, append (j, Sh_j) to Z

- $\mathcal{D}$ updates Z to include any relevant state information.

- Output Z

Figure 7: Joint LRSS Definition- $\mathsf{JLeak}^{m}_{Share}$ Distribution

4.5.2 Leakage resilience of $(\mathsf{Share}^h, \mathsf{Rec}^h)$ in $\mathcal{J}^{X,\psi,\tau}$ model

Theorem 3. *For any $\psi, X > 0$ such that $\psi \cdot X \leq n - t + 1$ and $l, \tau > 0$, $(\mathsf{Share}^h, \mathsf{Rec}^h)$ is an $((n,t),\varepsilon)$-secret sharing scheme for l bit messages and is ϵ_{lr}-leakage resilient in the joint leakage and reveal model $\mathcal{J}^{X,\psi,\tau}$ where $h = \lceil \frac{\psi}{\lfloor (t-1)/X \rfloor} \rceil$ and $\epsilon_{lr} = 2(\epsilon + h\epsilon' + (t-1)\sum_{i\in[h]} 2^{Xl_i}\delta'_i))$.*
Further, there exists an instantiation of the scheme with rate is $(X^{\Theta(h)} + h\tau/l)^{-1}$. When $\tau = \Theta(l)$, X is a constant and when either $n = \Theta(t)$ or h is a constant, the scheme achieves constant rate and leakage rate asymptotically.

The proof for the joint leakage setting is very similar to the proof of Theorem 2 for the adaptive setting (on single shares). For completeness, we give the proof in Appendix B.1. Further, we also give a detailed analysis of the parameters in Appendix B.2.

4.6 LRSS for General Access Structures

Our construction $(\mathsf{Share}^h, \mathsf{Rec}^h)$ can be easily adapted to provide security against general access structures as well. For $(\mathsf{Share}^h, \mathsf{Rec}^h)$ to be secure against a general access structure $\mathcal{A}$, the only modification is to instantiate MShare and $\mathsf{SdShare}^i$ in the construction to be secret sharing schemes that are secure (adaptively private) against $\mathcal{A}$ (for instance, the [9] scheme given in A.2.2).

Although, the leakage models we discussed previously in this section are formalized with respect to threshold access structures, the models naturally extend for general monotone access structures too. We would like to note that $(\mathsf{Share}^h, \mathsf{Rec}^h)$ (with the above modification) is also leakage resilient against these models for general access structures, and we defer more details on this to full version.

5 Applications of Our LRSS

We give two applications of our LRSS scheme: first, to build a leakage resilient non-malleable secret sharing scheme in the joint and adaptive leakage model, for the threshold access structure,

and second, to build a secure message transmission protocol, tolerating leakage and tampering attacks. We can instantiate these schemes with the constant-rate LRSS, to get a constant-rate LRNMSS and SMT.

5.1 Leakage Resilient Non-malleable Secret Sharing

Using our LRSS for the threshold access structures, we build an LRNMSS achieving the following leakage and tampering model: the adversary is allowed ψ joint and adaptive leakage queries on disjoint sets, with each query depending on X shares (exactly as in our joint leakage model of LRSS in section 4.5.1), with the restriction that the total number of leakage queries $\psi \cdot X \leq n - t - 2$. Post the leakage queries, the adversary can mention a reconstruction set (of size t), disjoint from the shares on which leakage queries were made, along with the tampering functions $f_1, \cdots, f_n$, acting independently on each share. By non-malleability, the tampered, reconstructed message $\tilde{m}$, is guaranteed to be either the same as the original message m, or is completely independent of it.

To build such an LRNMSS scheme, we use a modification of the [22] compiler: the message m is encoded using a 2-split-state non-malleable code[10] $(\mathsf{Enc}, \mathsf{Dec})$ to (L, R). Now, L is secret shared using a t-out-of-n LRSS to get $(\mathsf{L}_1, \cdots, \mathsf{L}_n)$ and R is secret shared using a $t-1$-out-of-n LRSS to get $(\mathsf{R}_1, \cdots, \mathsf{R}_n)$. The final share is $(\mathsf{L}_i, \mathsf{R}_i)$, for each $i \in [n]$.

For proving the leakage resilient non-malleability of this scheme, we need to simulate the tampering as split-state tampering (on the underlying NMC) 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, and obtain two of the tampered shares of L as leakage queries on the second LRSS. Note that, keeping the underlying leakage model in mind (and since the reconstruction set must be disjoint from the leakage query set), we restrict the number of leakage queries to be on at most $n-t-2$ 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. We provide details of the construction and proof, along with the rate analysis, in Appendix C.

5.2 Leakage Resilient (Non-malleable) Secure Message Transmission

We apply our LRSS and LRNMSS to the problem of secure message transmission (SMT) introduced in [18]. 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. Through an application of our LRSS and LRNMSS, we provide the first constructions of SMT protocols

[10] 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.

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 $n-t+1$ wires, in addition to viewing the complete contents on the $t-1$ remaining wires. Next, for the case of active adversaries, we obtain a leakage-resilient non-malleable SMT[11] protocol where the adversary can obtain leakage from messages sent on $n-t-2$ other wires in addition to viewing and completely modifying the contents on t wires. The detailed description of the models, along with the constructions and rate analysis, are given in Appendix D.

Acknowledgement. We thank all the anonymous reviewers who provided their valuable comments on an earlier version of this manuscript.

References

[1] Aggarwal, D., Damgård, I., Nielsen, J.B., Obremski, M., Purwanto, E., Ribeiro, J., Simkin, M.: Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In: Advances in Cryptology - CRYPTO 2019 (2019). https://doi.org/10.1007/978-3-030-26951-7_18, https://doi.org/10.1007/978-3-030-26951-7_18

[2] Aggarwal, D., Dodis, Y., Jafargholi, Z., Miles, E., Reyzin, L.: Amplifying privacy in privacy amplification. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II. Lecture Notes in Computer Science, vol. 8617, pp. 183–198. Springer (2014). https://doi.org/10.1007/978-3-662-44381-1_11, https://doi.org/10.1007/978-3-662-44381-1_11

[3] Aggarwal, D., Dodis, Y., Kazana, T., Obremski, M.: Non-malleable reductions and applications. In: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015 (2015). https://doi.org/10.1145/2746539.2746544, http://doi.acm.org/10.1145/2746539.2746544

[4] Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. In: Symposium on Theory of Computing, STOC 2014 (2014). https://doi.org/10.1145/2591796.2591804, http://doi.acm.org/10.1145/2591796.2591804

[5] Aggarwal, D., Obremski, M.: A constant rate non-malleable code in the split-state model. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020. pp. 1285–1294. IEEE (2020). https://doi.org/10.1109/FOCS46700.2020.00122, https://doi.org/10.1109/FOCS46700.2020.00122

[6] Badrinarayanan, S., Srinivasan, A.: Revisiting non-malleable secret sharing. In: Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques (2019). https://doi.org/10.1007/978-3-030-17653-2_20, https://doi.org/10.1007/978-3-030-17653-2_20

[11] The notion of non-malleable SMT without leakage was introduced in a recent work of [22]. We strengthen their adversarial model to incorporate leakage.

[7] Bellare, M., Rogaway, P.: 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, Association for Computing Machinery (2007). https://doi.org/10.1145/1315245.1315268, https://doi.org/10.1145/1315245.1315268

[8] Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 1–10. ACM (1988). https://doi.org/10.1145/62212.62213, https://doi.org/10.1145/62212.62213

[9] Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference. Springer (1988). https://doi.org/10.1007/0-387-34799-2_3, https://doi.org/10.1007/0-387-34799-2_3

[10] Benhamouda, F., Degwekar, A., Ishai, Y., Rabin, T.: On the local leakage resilience of linear secret sharing schemes. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference. e (2018). https://doi.org/10.1007/978-3-319-96884-1_18, https://doi.org/10.1007/978-3-319-96884-1_18

[11] Blakley, G.: Safeguarding cryptographic keys. In: Proceedings of the 1979 AFIPS National Computer Conference. AFIPS Press (1979)

[12] Brian, G., Faonio, A., Obremski, M., Simkin, M., Venturi, D.: Non-malleable secret sharing against bounded joint-tampering attacks in the plain model. In: Advances in Cryptology - CRYPTO 2020 (2020). https://doi.org/10.1007/978-3-030-56877-1_5, https://doi.org/10.1007/978-3-030-56877-1_5

[13] Brian, G., Faonio, A., Venturi, D.: Continuously non-malleable secret sharing for general access structures. In: Theory of Cryptography - 17th International Conference, TCC 2019 (2019). https://doi.org/10.1007/978-3-030-36033-7_8, https://doi.org/10.1007/978-3-030-36033-7_8

[14] Chattopadhyay, E., Goodman, J., Goyal, V., Kumar, A., Li, X., Meka, R., Zuckerman, D.: Extractors and secret sharing against bounded collusion protocols. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020 (2020). https://doi.org/10.1109/FOCS46700.2020.00117, https://doi.org/10.1109/FOCS46700.2020.00117

[15] Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 11–19. ACM (1988). https://doi.org/10.1145/62212.62214, https://doi.org/10.1145/62212.62214

[16] Davì, F., Dziembowski, S., Venturi, D.: Leakage-resilient storage. In: Security and Cryptography for Networks, 7th International Conference, SCN 2010 (2010). https://doi.org/10.1007/978-3-642-15317-4_9, https://doi.org/10.1007/978-3-642-15317-4_9

[17] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: 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

[18] Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM **40**(1), 17–47 (1993). https://doi.org/10.1145/138027.138036, https://doi.org/10.1145/138027.138036

[19] Dziembowski, S., Pietrzak, K.: Intrusion-resilient secret sharing. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science. FOCS '07 (2007). https://doi.org/10.1109/FOCS.2007.35, http://dx.doi.org/10.1109/FOCS.2007.35

[20] Faonio, A., Venturi, D.: 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 (2019). https://doi.org/10.1007/978-3-030-26951-7_16, https://doi.org/10.1007/978-3-030-26951-7_16

[21] Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from leakage: the computationally-bounded and noisy cases. In: Advances in Cryptology - EUROCRYPT 2010 (2010)

[22] Goyal, V., Kumar, A.: Non-malleable secret sharing. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018 (2018). https://doi.org/10.1145/3188745.3188872, https://doi.org/10.1145/3188745.3188872

[23] Guruswami, V., Umans, C., Vadhan, S.P.: Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In: IEEE Conference on Computational Complexity. pp. 96–108 (2007)

[24] Guruswami, V., Wootters, M.: Repairing reed-solomon codes. In: Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing. STOC '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2897518.2897525, http://doi.acm.org/10.1145/2897518.2897525

[25] Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology—CRYPTO 2003. LNCS, vol. 2729. Springer-Verlag (2003)

[26] Kishore, R., Kumar, A., Vanarasa, C., Srinathan, K.: On the price of proactivizing round-optimal perfectly secret message transmission. IEEE Trans. Inf. Theory **64**(2), 1404–1422 (2018). https://doi.org/10.1109/TIT.2017.2776099, https://doi.org/10.1109/TIT.2017.2776099

[27] Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology—CRYPTO '96. LNCS, vol. 1109. Springer-Verlag (18–22 Aug 1996)

[28] Kumar, A., Meka, R., Sahai, A.: Leakage-resilient secret sharing against colluding parties. In: 60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019 (2019). https://doi.org/10.1109/FOCS.2019.00045, https://doi.org/10.1109/FOCS.2019.00045

[29] Kurosawa, K., Suzuki, K.: Truly efficient 2-round perfectly secure message transmission scheme. IEEE Trans. Inf. Theory **55**(11), 5223–5232 (2009). https://doi.org/10.1109/TIT.2009.2030434, https://doi.org/10.1109/TIT.2009.2030434

[30] Lin, F., Cheraghchi, M., Guruswami, V., Safavi-Naini, R., Wang, H.: Non-malleable secret sharing against affine tampering. CoRR **abs/1902.06195** (2019), http://arxiv.org/abs/1902.06195

[31] Liu, F., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference (2012). https://doi.org/10.1007/978-3-642-32009-5_30, https://doi.org/10.1007/978-3-642-32009-5_30

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

[33] Rothblum, G.N.: How to compute under ac0 leakage without secure hardware. In: Proceedings of the 32Nd Annual Cryptology Conference on Advances in Cryptology — CRYPTO 2012 (2012). https://doi.org/10.1007/978-3-642-32009-5_32, https://doi.org/10.1007/978-3-642-32009-5_32

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

[35] Srinathan, K., Narayanan, A., Rangan, C.P.: Optimal perfectly secure message transmission. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference (2004). https://doi.org/10.1007/978-3-540-28628-8_33, https://doi.org/10.1007/978-3-540-28628-8_33

[36] Srinivasan, A., Vasudevan, P.N.: Leakage resilient secret sharing and applications. In: Advances in Cryptology – CRYPTO 2019. Springer International Publishing (2019)

[37] Vadhan, S.: Pseudorandomness. Foundations and Trends in Theoretical Computer Science, Now Publishers (2012), available at http://people.seas.harvard.edu/~salil/pseudorandomness/

[38] Wang, Y., Desmedt, Y.: Perfectly secure message transmission revisited. IEEE Trans. Inf. Theory (2008). https://doi.org/10.1109/TIT.2008.921676, https://doi.org/10.1109/TIT.2008.921676

[39] Zimand, M.: Exposure-resilient extractors. In: 21st Annual IEEE Conference on Computational Complexity (CCC 2006). IEEE Computer Society (2006). https://doi.org/10.1109/CCC.2006.19, https://doi.org/10.1109/CCC.2006.19

A Some Definitions and Preliminary Lemmata

A.1 Properties of Randomness Extractors

We 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 9. *If* Ext *is an* $(\eta, \mu, d, l, \epsilon)$*-strong average case extractor, then for any* $u \geq 1$ *and any* $\delta > 0$*,* Ext_u *is an* $(u\eta, (u-1)\eta + \mu, d, ul, u\epsilon)$*-strong average case extractor, where* Ext_u *is defined as follows:*

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

Further when the source(w*) has uniform distribution,* Ext_u *is an* $(u\eta, u\eta, d, ul, u\epsilon)$*-extractor that is* $(\mathsf{Full}_{\eta-\mu}, 2^{ul+2}u\epsilon)$*-adaptive.*

Proof. Let $W|Z$ be the $(u\eta, (u-1)\eta + \mu)$- average source, (where W is parsed as $(W_1, \cdots, W_u)$) and $S \equiv U_d$. Then, by Lemma 1, $\widetilde{\mathbf{H}}_\infty(W_u|W_1, \cdots, W_{u-1}, Z) \geq \mu$. Therefore, by the security of Ext, we have

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

Then by Lemma 2 it follows that,

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

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

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

$$Z, W_1, \cdots, W_{u-2}, \mathsf{Ext}(W_{u-1}; S), U_l, S \approx_\epsilon Z, W_1, \cdots, W_{u-2}, U_l', U_l, S^{12}$$

Thus, by triangle inequality,

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

Then by similar arguments, it is easy to see that

$$Z, (\mathsf{Ext}(W_1; S), \cdots, \mathsf{Ext}(W_{u-2}; S), \mathsf{Ext}(W_{u-1}; S), \mathsf{Ext}(W_u; S)), S \approx_{u\epsilon} Z, U_{ul}, S$$

The adaptivity property follows from application of Theorem 1 on Ext_u. □

Further, we prove Lemma 8 on adaptive extractors, stated in Section 3 and used in our proofs.

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

A.1.1 Proof of Lemma 8

We prove the lemma by hybrid approach. If $k = 1$, the lemma trivially follows from adaptive security of Ext. Let (j_1, g_{j_1}) be the first query made by $\mathcal{D}'$. Firstly, as $g_{j_1} \in \mathsf{Full}_m$ by adaptive extractor security of Ext on the source W_{j_1} and the seed S, we have

$$j_1, g_{j_1}, S, E^0_{j_1}, g_{j_1}(E^0_1) \approx_\delta j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_1).$$

The second query $(j^0_2, g_{j^0_2})$ (resp. $(j^1_2, g_{j^1_2})$) made by $\mathcal{D}'$ in Adleak^0 (resp. Adleak^1) is a function of the LHS (resp. RHS). Therefore

$$j^0_2, g_{j^0_2}, j_1, g_{j_1}, S, E^0_{j_1}, g_{j_1}(E^0_1) \approx_\delta j^1_2, g_{j^1_2}, j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_1).$$

Since $W_{j^0_2}, W_{j^1_2}$ are identical and independent of W_{j_1} we have

$$W_{j^0_2}, j^0_2, g_{j^0_2}, j_1, g_{j_1}, S, E^0_{j_1}, g_{j_1}(E^0_{j_1}) \approx_\delta W_{j^1_2}, j^1_2, g_{j^1_2}, j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_{j_1}).$$

$$j^0_2, g_{j^0_2}, j_1, g_{j_1}, S, E^0_{j_1}, g_{j_1}(E^0_{j_1}), E^0_{j^0_2}, g_{j^0_2}(E^0_{j^0_2}) \approx_\delta j^1_2, g_{j^1_2}, j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_{j_1}), E^0_{j^1_2}, g_{j^1_2}(E^0_{j^1_2}).$$

Further by adaptive security of the source $W_{j^1_2}$, we have RHS of the above expression to be δ close to $(j^1_2, g_{j^1_2}, j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_{j_1}), E^1_{j^1_2}, g_{j^1_2}(E^1_{j^1_2}))$. By triangle inequality we get,

$$j^0_2, g_{j^0_2}, j_1, g_{j_1}, S, E^0_{j_1}, g_{j_1}(E^0_{j_1}), E^0_{j^0_2}, g_{j^0_2}(E^0_{j^0_2}) \approx_{2\delta} j^1_2, g_{j^1_2}, j_1, g_{j_1}, S, E^1_{j_1}, g_{j_1}(E^1_{j_1}), E^1_{j^1_2}, g_{j^1_2}(E^1_{j^1_2}).$$

With similar arguments, we can show that

$$S, \{j^0_i, g_{j^0_i}, E^0_{j^0_i}, g_{j^0_i}(E^0_{j^0_i})\}_{i\in[k]} \approx_{k\delta} S, \{j^1_i, g_{j^1_i}, E^0_{j^1_i}, g_{j^1_i}(E^1_{j^1_i})\}_{i\in[k]}$$

which concludes the proof.

A.2 Properties of Secret Sharing Schemes

A.2.1 Proof of Lemma 5

Let message m be fixed. We prove the lemma inductively (on size of T). First, suppose $T = \{i_1\} \subseteq [N]$. Let $\mathsf{Share}(m; \mathcal{R}) \equiv (\mathcal{SH}^m_j)_{j\in[N]}$ denote the distribution on the shares of m, where $\mathcal{R}$ is the uniform distribution on space of randomness R. Consider for any $(sh_1, \cdots, sh_N) \in \mathsf{Share}(m; R)$ and for $(sh'_j)_{j\in[N]} \leftarrow \mathsf{Share}(m; \mathcal{R})$:

$$\begin{aligned}
\Pr[\mathcal{D}_1 = (sh_1, \cdots, sh_N)] &= \Pr[\mathcal{SH}^m_j = sh_j \ \forall\ j \in [N] | sh_{i_1} = sh'_{i_1}] \\
&= \Pr[\mathcal{SH}^m_{i_1} = sh'_{i_1}, \mathcal{SH}^m_j = sh_j \ \forall j \neq i_1 \ \in [N]] \\
&= \Pr[\mathcal{D}_2 = (sh_1, \cdots, sh_N)] \qquad (2)
\end{aligned}$$

The last equality follows since $sh'_{i_1} = sh_{i_1}$.
Now, suppose that the distributions are equivalent for $|T| = k-1$ (for any $2 \leq k \leq N-1$), then we show that for $|T| = k$ as well, they are equivalent. Consider for any $(sh_1, \cdots, sh_N) \in \mathsf{Share}(m; R)$,

for $(sh'_j)_{j\in[N]} \leftarrow \mathsf{Share}(m;\mathcal{R})$ and for $T = \{i_1, \cdots, i_k\} \subseteq [N]$, where, for each $j \in [k]\backslash\{1\}$, i_j is generated as an arbitrary function of $\{sh'_m : m \in \{i_1, \cdots, i_{j-1}\}\}$:

$$\begin{aligned}\Pr[\mathcal{D}_1 = (sh_1, \cdots, sh_N)] &= \Pr[\mathcal{SH}^m_j = sh_j \ \forall\ j \in [N] | sh_{i_j} = sh'_{i_j}\ \forall j \in [k]] \\ &= \Pr[\mathcal{SH}^m_{i_1} = sh'_{i_1}, \mathcal{SH}^m_j = sh_j\ \forall j \neq i_1\ \in [N] | sh_{i_j} = sh'_{i_j}\ \forall j \in [k]\backslash\{1\}] \\ &= \Pr[\mathcal{SH}^m_{i_1} = sh_{i_1}, \mathcal{SH}^m_j = sh_j\ \forall j \neq i_1\ \in [N] | sh_{i_j} = sh'_{i_j}\ \forall j \in [k]\backslash\{1\}] \qquad (3)\end{aligned}$$

which follows by equation 2 (the first case). Now, since, fixing sh'_{i_1}, $i_2 = f(sh'_{i_1})$ is a fixed index, and since the equivalence is true for $|T| = k-1$ by the induction hypothesis, the above equation 3 is equal to:

$$\Pr[\mathcal{D}_2 = (sh_1, \cdots, sh_N)]$$

Hence, we proved the lemma for $T \subseteq [N]$, for all $1 \leq |T| \leq N-1$.

A.2.2 Instantiations of Adaptively Private Secret Sharing Schemes

Adaptive privacy of Shamir Secret Sharing. We begin by looking at the Shamir secret sharing scheme [34] for threshold access structures and show that it is, in fact, adaptively private. Consider the field $\mathbb{Z}_q$ for prime q such that $n < q$. The shamir secret sharing scheme is described below.

$\mathsf{ShamirShare}^t_n(m)$:

- Set $a_0 = m$ and pick $a_1, \cdots, a_{t-1} \in_R \mathbb{Z}_q$.
- Define the polynomial $p(x) = a_0 + a_1 x + \cdots + a_t x^{t-1}$.
- Output: $(p(1), \cdots, p(n))$.

$\mathsf{ShamirRec}^t_n(\{y_i\}_{i\in T})$: (where $|T| = t$ is some reconstruction set. For simplicity in writing, say $T = \{1, \cdots, t\}$)

- Use Lagrange interpolation to reconstruct the polynomial:
  - For $1 \leq i \leq t$: the Lagrange polynomial L_i is defined as the degree t polynomial for which $L_i(i) = 1$ and $L_i(j) = 0$, for each $j \neq i, 1 \leq j \leq t$.
  - $p(x) = \sum_{i=1}^t y_i L_i(x)$.
- Output $m = p(0)$.

Lemma 10. *The Shamir secret sharing scheme (described above) is adaptively private against the t-threshold access structure.*

Proof. To prove that the scheme is adaptively private, according to definition in section 2.2.1, it is sufficient to prove the following claim.

Claim 1. *For any $U \subseteq [n]$ such that $|U| \leq t-1$ (unauthorized set), the following two distributions are equivalent.*

$\mathcal{X}$:

- $a_0 = m$, $a_1, \cdots, a_{t-1} \in_R \mathbb{Z}_q$.
- $p(x) = \sum_{i=0}^{t-1} a_i x^i$.
- *Output* $(p(i))_{i \in U}$.

$\mathcal{Y}$:

- $y_i \in_R \mathbb{Z}_q$, *for each* $i \in U$.
- *Output* $(y_i)_{i \in U}$.

Proof. Let $U = \{i_1, \cdots, i_{t-1}\}$, vandermonde matrix V be defined as: $V_{kj} = i_k^j$, for each $k, j \in [t-1]$, A be the column matrix: $(a_1 \cdots a_{t-1})^T$, Z be the column matrix $(z_{i_1} \cdots z_{i_{t-1}})^T$ and A_0 be the $(t-1) \times 1$ column matrix whose all entries are a_0. For any $z_{i_1}, \cdots, z_{i_{t-1}}$ from the field $\mathbb{Z}_q$,

$$\begin{aligned} \Pr[\mathcal{X} = (z_i)_{i \in U}] &= \Pr[VA = Z - A_0] \\ &= \Pr[A = V^{-1}(Z - A_0)] \\ &= 1/q^{t-1} = \Pr[\mathcal{Y} = (z_i)_{i \in U}] \end{aligned}$$

□

Given the above claim, clearly, each adaptive unauthorized query can be responded with a random field element. Hence, the scheme is adaptively private. □

Adaptive Privacy of Benaloh-Leichter Secret Sharing. We begin by describing the n-party Benaloh-Leichter secret sharing scheme [9] for general monotone access structure, $\mathcal{A}$. Any monotone access structure $\mathcal{A}$ can be associated with an equivalent monotone formula $\mathcal{F}$, on n variables (the access structure $\mathcal{A}$ defined by F will be the set of subsets of parties, A, for which F is true when the variables indexed by A are set to true, and vice versa). Further, any monotone formula can be implemented using only AND and OR operators. We denote the variables of the monotone formulae by $\{v_i : i \in [n]\}$. We describe the sharing procedure, $\mathsf{BLShare}(s, F)$, for secret $s \in \mathbb{Z}_q$ (where $\mathbb{Z}_q$ is a field of prime order q such that $n < q$) and a monotone formula F, in a recursive manner as below.

$\mathsf{BLShare}(s, F)$:

- $\mathsf{BLShare}(s, v_i)$ assigns the share s to party P_i.
- $\mathsf{BLShare}(s, A \vee B) = \mathsf{BLShare}(s, A) \cup \mathsf{BLShare}(s, B)$, for every clause A and B in F.
- $\mathsf{BLShare}(s, A \wedge B) = \mathsf{BLShare}(s_1, A) \cup \mathsf{BLShare}(s_2, B)$, where s_1 and s_2 are uniformly chosen from $\mathbb{Z}_q$ such that $s = (s_1 + s_2) \mod q$, for every clause A and B in F.

In general for operators with more than 2 arguments and if $\mathsf{THRESHOLD}_k$ operators (which are true if and only if at least k of its clauses are true) are used, do the following:

- $\mathsf{BLShare}(s, \vee(F_1, \cdots, F_m)) = \cup_{1 \leq i \leq m} \mathsf{BLShare}(s, F_i)$.
- $\mathsf{BLShare}(s, \wedge(F_1, \cdots, F_m)) = \cup_{1 \leq i \leq m} \mathsf{BLShare}(s_i, F_i)$, where s_i's are chosen uniformly from $\mathbb{Z}_q$ such that $s = \sum_{i=1}^{m} s_i \mod q$.

- $\mathsf{BLShare}(s, \mathsf{THRESHOLD}_k(F_1, \cdots, F_m)) = \cup_{1 \le i \le m} \mathsf{BLShare}(s_i, F_i)$, where $(s_1, \cdots, s_m) \leftarrow \mathsf{ShamirShare}^k_m(s)$.

Lemma 11. *For every monotone formula F (with corresponding access structure $\mathcal{A}$), the above Benaloh-Leichter scheme for n parties (described above) is adaptively private.*

Proof. To prove the adaptive privacy of the scheme, according to the definition in section 2.2.1, it is sufficient to prove the following claim.

Claim 2. *For each $A \notin \mathcal{A}$ (unauthorized set), the following distributions are equivalent.*

$\mathcal{X}$:	$\mathcal{Y}$:
• $Sh_1, \cdots, Sh_n \leftarrow \mathsf{BLShare}(s, F)$.	• $Sh_i \in_R \mathbb{Z}_q$, *for each* $i \in A$.
• *Output* $(Sh_i)_{i \in A}$.	• *Output* $(Sh_i)_{i \in A}$.

Proof. We prove the claim by induction on the number of operators in the formula F.

- *Base Case:* A formula with no operators consists of one vaiable v_i, for some $i \in [n]$ and the corresponding access structure will have all subsets of parties containing the i-th one. $\mathsf{BLShare}(s, v_i)$ gives s to P_i alone. For each $A \notin \mathcal{A}$, $i \notin A$, and hence, parties in A do not get anything in this case.

- *Induction Hypothesis:* Suppose the claim holds true for F with $< d$ operators ($d > 0$).

- Now, let F be formula containing d operators, written as, $o(F_1, \cdots, F_m)$, where o is one of $\vee$, $\wedge$ or $\mathsf{THRESHOLD}_k$, and each of $F_1, \cdots, F_m$ is a monotone formula with less than d operators. The following cases arise depending on which operator o is considered.

  – *Case 1:* For $o = \vee$, $\mathsf{BLShare}(s, \vee(F_1, \cdots, F_m)) = \cup_{1 \le i \le m} \mathsf{BLShare}(s, F_i)$. By induction hypothesis, for each $A \notin \mathcal{A}$, for each $1 \le i \le m$, the shares assigned to parties in A by $\mathsf{BLShare}(s, F_i)$ are identical to random field elements. Moreover, for $i \neq j$, $\mathsf{BLShare}(s, F_i)$ and $\mathsf{BLShare}(s, F_j)$ assign shares independently. Hence, for all parties in $\mathcal{A}$, the shares held by them are identical to uniform.

  – *Case 2:* For $o = \wedge$, $\mathsf{BLShare}(s, \wedge(F_1, \cdots, F_m)) = \cup_{1 \le i \le m} \mathsf{BLShare}(s_i, F_i)$, where $s_1, \cdots, s_m$ are chosen at random from $\mathbb{Z}_q$ such that $s = \sum_{i=1}^{m} s_i \mod q$. Now, for any $A \notin \mathcal{A}$, $\exists\, i$, such that, the shares of $\mathsf{BLShare}(s_i, F_i)$ held by parties in A have no information about s_i. Further, by induction hypothesis, since the parties in A corresponding to F_i will be unauthorized, shares of $\mathsf{BLShare}(s_i, F_i)$ corresponding to them will look uniform. Since the following two distributions are identical:

$*$ $s_1, \cdots, s_n \in_R \mathbb{Z}_q | (s = \sum_{i=1}^{m} s_i \mod q)$.	$*$ $s_j \in_R \mathbb{Z}_q$, for each $j \neq i$.
$*$ Output $(s_j)_{j \neq i}$.	$*$ Output $(s_j)_{j \neq i}$.

    therefore, the shares of parties in A corresponding to $\mathsf{BLShare}(s_j, F_j)$, for each $j \neq i$ are also identical to uniform.

– For $o = \mathsf{THRESHOLD}_k$, $\mathsf{BLShare}(s, \mathsf{THRESHOLD}_k(F_1, \cdots, F_m)) = \cup_{1 \le i \le m} \mathsf{BLShare}(s_i, F_i)$, where $(s_1, \cdots, s_m) \leftarrow \mathsf{ShamirShare}^k_m(s)$. For $A \notin \mathcal{A}$, parties in A can only get $< k$ of the s_i's (else A will be in $\mathcal{A}$), say $s_{i_1}, \cdots, s_{i_{k-1}}$ are known to parties in A. By the property of k-threshold Shamir secret sharing (shown in the proof of Lemma A.2.2), $(s_{i_1}, \cdots, s_{i_{k-1}}) \equiv U_{\mathbb{Z}_q^{k-1}}$. Hence, the shares obtained by parties in A corresponding to $\mathsf{BLShare}(s_{i_j}, F_{i_j})$, for each $j \in [k-1]$, are identical to uniform. Further, for each $j \notin \{i_1, \cdots, i_{k-1}\}$, since the parties in A are unauthorised w.r.t. $\mathsf{BLShare}(s_j, F_j)$, by induction hypothesis, their shares are identical to uniform.

Hence, by induction, the claim is proved. □

Given the above claim, clearly each adaptive unauthorized query can be responded with a random field element. Hence, the scheme is adaptively private. □

B Security Proof and Parameters of LRSS scheme in the Joint Leakage and Reveal Model

We give a detailed proof sketch of Theorem 3 below, which is very similar to the proof of Theorem 2.

B.1 Proof Sketch of Theorem 3

Correctness and privacy are already discussed in Theorem 2. We discuss leakage resilience against joint leakage here. The proof idea is similar in spirit to the leakage resilience proof of the adaptive leakage and reveal model. Let $a = \lfloor (t-1)/X \rfloor$ and $h = \lceil \frac{\psi}{a} \rceil$. We group the query sets into sets $\mathcal{JS}_1, \cdots, \mathcal{JS}_h$ as follows. For $i \in [h-1]$, $\mathcal{JS}_i = Q_{(i-1)a+1} \cup \cdots \cup Q_{ia}$. $\mathcal{JS}_h = Q_{(h-1)a+1} \cup \cdots \cup Q_\psi$.

For any message m we define the following the sequence of hybrids (which are similar to hybrids in proof of Thm. 2). Without loss of generality, we assume that $\mathcal{D}$ always asks legitimate queries as per the model. The sequence of hybrids is JLeakB^m_0, $\{\mathsf{JLeakA}^m_q, \mathsf{JLeakB}^m_q\}_{q \in [h]}$ and JLeakC^m. Let JLeakB^m_0 be the distribution $\mathsf{JLeak}^m_{\mathsf{Share}^h}$.

$\underline{\mathsf{JLeakA}^m_q}$: For each $1 \le q \le h$, the only change we make in JLeakA^m_q (in comparison to the previous hybrid JLeakB^m_{q-1}) is that we replace the shares sd^q_j, for each $j \in \mathcal{JS}_q$ (the shares of s^q corresponding to the q-th set joint leakage queries), with shares of a dummy seed $\widetilde{s}^q$. After answering the joint leakage queries corresponding to $\mathcal{S}_q$, $sd^q_{[n]}$ is re-sampled as shares of s^q conditioned on $\widetilde{s}^{\mathcal{S}_q}$. The statistical closeness of hybrids JLeakB^m_{q-1} and JLeakA^m_q follows from adaptive privacy of $\mathsf{SdShare}^q$. This is because $|\mathcal{JS}_q| \le t-1$ and leakage responses to all queries in $\mathcal{JS}_{[q-1]}$ would be independent of s^q and its shares.

$\underline{\mathsf{JLeakB}^m_q}$: For each $1 < q \le h$, the only change we make in JLeakB^m_q (in comparison to the previous hybrid JLeakA^m_q) is that we replace the values x^q_j, for each $j \in \mathcal{JS}_q$ with random, instead of evaluating the $h-(q-1)$ layers of masking to get x^q_j (and hence the query response for any $Q_k \subseteq \mathcal{JS}_q$ is independent of m_{Q_k} and s^i for each $q \le i \le h$). Further, we continue to evaluate $x^{q-1}_j, x^{q-2}_j, \cdots, x^1_j, y^1_j$, for each $j \in \mathcal{JS}_q$ as in the previous hybrid.

The response to each joint leakage query in JLeakA^m_q on any set $Q_k \subseteq \mathcal{JS}_q$ depends on $\{w^q_j, \mathsf{Ext}^q(w^q_j; s^q)\}_{j \in Q_k}$ along with $\{w^i_j, s^i, sd^i_j\}_{i \in [h] \setminus \{q\}, j \in Q_k}$ and shares (corresponding to $j \in Q_k$) of

a dummy seed $\widetilde{s}^q$ (the distribution of the latter two random variables is identical in both hybrids). Now, consider a mega-extractor EXT^q which takes $w^q_{Q_k}$, s^q as input and outputs $\{\mathsf{Ext}^q(w^q_j; s^q)\}_{j\in Q_k}$ (as in Lemma 9 with respect to Ext^q and $u = X$). By Lemma 9, we know EXT^q is $(\mathsf{Full}_\tau, 4X\delta'_q \cdot 2^{Xl_q})$-adaptive[13]. Using adaptivity of this extractor with Lemma 8, we can show that leakage responses for all queries in $\mathcal{S}_q$ are statistically close by $(4(t-1)X\delta'_q \cdot 2^{Xl_q})$ in both hybrids. After answering queries corresponding to $\mathcal{S}_q$, all further joint leakage/reveal queries are answered in identical manner in both hybrids.

$\underline{\mathsf{JLeakC}^m}$**:** In the hybrid JLeakB^m_h, all the shares used in the leakage phase are independent of the shares of the message m. Hence, the only part of the view of $\mathcal{D}$ that depends on the shares of m corresponds to the reveal phase. In the final hybrid JLeakC^m, we replace the $t-1$ shares of m used in the reveal phase by shares of 0^l. This hybrids are close by adaptive privacy of MShare.

B.2 Parameters

For $i \in [h]$, we instantiate $\mathsf{SdShare}^i$ on seeds of length d_i with the (adaptively) private Shamir secret sharing scheme, which results in individual seed share length being d_i. We instantiate MShare on messages of length l_i with the (adaptively) private Shamir secret sharing scheme, which results in individual seed share length being l_i.

Recall, Lemma 7 that states that for any $c > 1$, there exists constants α, β such that $d \le \alpha l$, $\mu \le \beta l$, $\eta \ge \beta l + \tau$, $\epsilon = 2^{-cl}$ and $\delta = 2^{-(c-1)l+2}$ when $l = \omega(\log \eta)$. Fix $c = 2X$, and α, β be values corresponding to this c given by Lemma 7. For each $i \in [h]$, we instantiate $(\eta_i, \mu_i, d_i, l_i, \delta'_i)$-extractor Ext^i as per this lemma as follows[14].

- We set $l_1 = l$, $\delta'_1 = 2^{-cl}$, $d_1 \le \alpha l_1$, $\mu_1 \le \beta l_1$ and $\eta_1 = \beta l_1 + \tau$.
- For $i > 1$, we set $l_i = l_{i-1} + d_{i-1}$, $\delta'_i = 2^{-cl_i}$, $d_i \le \alpha l_i$, $\mu_i \le \beta l_i$ and $\eta_i = \beta l_i + \tau$.

With this setting, individual share length of Share^h is $l_h + d_h + \sum_{i\in[h]} \eta_i = h\tau + \Theta((1+\alpha)^h l) = X^{\Theta(h)} l + h\tau$(as α is specific to $c(= 2X)$). Therefore, when $\tau = \mathcal{O}(l)$ and either $n = \Theta(t)$ or h is a constant, Share^h achieves

- Constant rate and constant leakage rate whenever X is constant.
- Inverse poly logarithmic rate when $X = \log n$.

As our instantiations of $\mathsf{SdShare}^i$'s and MShare are perfectly adaptively private, we have Share^h to be a perfectly adaptively private secret sharing scheme which is $t\sum_{i\in[h]} 2^{Xl_i} \cdot \delta'_i = t2^{-\Omega(l)}$-leakage resilience against the adaptive leakage and reveal model.

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

We begin by defining an LRNMSS and describing our tampering model, for the threshold access structure.

[13] While setting parameters we set $\eta_q - \mu_q \ge \tau$

[14] Adaptivity of Ext^i isn't important and hence we don't mention parameters of adaptivity.

C.1 Tampering Model

The tampering model with adaptive leakage ($\mathcal{F}^{n-t-2,\tau}_{tamper}$), that we consider for our LRNMSS scheme, is as defined below.
Let $\mathsf{Share} : \mathcal{M} \to (\{0,1\}^\gamma)^n$ 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 exactly the LRSS leakage model $\mathcal{J}^{X,\psi,\tau}$ for the t-threshold access structure, where you do not allow full share queries[15] and only allow the leakage queries (with leakage threshold τ as in $\mathcal{J}^{X,\psi,\tau}$) on at most $n-t-2$ shares (i.e., $\psi X \leq n-t-2$). 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 $\mathcal{J}^{X,\psi,\tau}$-leakage queries were made.
- G is a function acting on $\{share_i\}_{i\in\mathcal{L}}$ and follows the leakage model of $\mathcal{J}^{X,\psi,\tau}$ with the added restriction that $|\mathcal{L}| \leq n-t-2$ (i.e., each (adaptive) leakage query can depend on up to $t-1$ shares, disjoint from the prior queries and the total number of shares from which leakage is allowed is $|\mathcal{L}|$).

The threshold τ for leakage is exactly what $\mathcal{J}^{X,\psi,\tau}$ 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 $\mathcal{F}^{n-t-2,\tau}_{tamper}$ as the set of functions $(G, \mathcal{L}, f_1, \cdots, f_n, I)$[16] 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 $\mathcal{F}^{n-t-2,\tau}_{tamper}$ defined above, for the threshold access structure[17].

Definition 5 (Leakage Resilient 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* $\mathcal{F}^{n-t-2,\tau}_{tamper}$ *be the family of tampering*

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

[16] 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.

[17] This definition can be thought of as a special adaptation of the general definition [22] of non-malleable secret sharing against a tampering family $\mathcal{F}$

functions described above. For each $(G, \mathcal{L}, f_1, \cdots, f_n, I) \in \mathcal{F}^{n-t-2,\tau}_{tamper}$, $m \in \mathcal{M}$ *define the tampering experiment*

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

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* $\mathcal{F}^{n-t-2,\tau}_{tamper}$ *if for each* $(G, \mathcal{L}, f_1, \cdots, f_n, I) \in \mathcal{F}^{n-t-2,\tau}_{tamper}$ *there exists a distribution* $\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_n,I}$ *over* $\mathcal{M} \cup \{same^*, \perp\}$ *such that,*
$\forall\ m$, $\mathsf{STamper}^{G,\mathcal{L},f_1,\cdots,f_n,I}_{m} \approx_{\epsilon_{nm}} Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_n,I}, m)$*, where*

$$Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_n,I}, m) = \left\{ \begin{array}{ll} (\mathsf{Leak}, \tilde{m}) & \leftarrow \mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_n,I} \\ Output : & (\mathsf{Leak}, m)\ \textit{if}\ \tilde{m} = same^* \\ & (\mathsf{Leak}, \tilde{m})\ \textit{otherwise} \end{array} \right\}$$

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

C.2 Comparison with Prior Work

We give a comparison of our work with the most relevant works on leakage-resilient non-malleable secret sharing below.

1. In the information theoretic setting, the only known LRNMSS schemes are [28, 13], 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 $n - t - 2$ shares in total before allowing a single independent tampering query, and we achieve a constant rate, for the setting where each query depends on at most a constant number of shares and $t = \alpha n$ (for constant $\alpha < 1$). While our leakage model is incomparable to [28, 13], we get the first *constant rate* scheme for an adaptive leakage model.

2. In the computational setting, there are several works [13, 20, 12] 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 [12], in combination with the compiler from [20] 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[18].

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	$n-t-2$	*constant*	Yes	Single

Table 2: LRNMSS Prior Work

- *[6, 36] and [1] are for 4 and 3-monotone access structures.
- **[1] has a stronger concurrent tampering model than [6].
- n represents the number of parties and t represents the threshold. For our instantiation of constant-rate LRSS, we set $t = \alpha n$ (for a constant $\alpha < 1$).

C.3 Building Blocks

Before we describe our construction, we look at the building blocks needed. Specifically, we need a stronger guarantee of "conditional independence" from the underlying LRSS scheme, and we need non-malleable codes.

C.3.1 Conditional Independence of LRSS

To instantiate the non-malleable secret sharing construction in Section C.4 with the leakage resilient secret sharing of Section 4.5, we need an additional stronger property from the LRSS scheme, which is called conditional independence, defined as below.

Definition 6. *[6] 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 $\mathcal{J}^{X,\psi,\tau}$ (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:*

[18] 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 [30] 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.

- *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* $G_{\mathcal{L},K} \in \mathcal{J}^{X,\psi,\tau}$ *(following the adaptive and joint leakage model of* $\mathcal{J}^{X,\psi,\tau}$*) 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 4.5 satisfies the desired conditional independence property. For completeness, we show this in the following lemma.

Lemma 12. *The LRSS scheme of Section 4.5 is a* t*-threshold leakage resilient secret sharing scheme, that is leakage resilient with respect to* $\mathcal{J}^{X,\psi,\tau}$ *with conditional independence.*

Proof. The proof of correctness and privacy follow directly from Theorem 3. We prove conditional independence and leakage resilience, as in Definition 6.
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 $m \in \mathcal{M}$. We consider the construction in Figure 3.
Define $\mathsf{aux}_{K,S}$ as a function, which, on input m and randomness $rand$, outputs $\mathsf{aux} = s^1, \cdots, s^h$, where $s^1, \cdots, s^h$ are the seed of the extractors ($s^1, \cdots, s^h$ are part of $rand$).
Now, we fix Sh_K, aux, m. Then, it is clear that this fixes all the shares $(m_1, \cdots, m_n)$ of m (since $|K| = t - 1$). The only randomness for sampling Sh_i for any $i \in [n]\setminus K$ is in sampling $w_i^1, \cdots, w_i^h$, which are independent for each i. Hence, conditioned on fixing Sh_K, aux, m, the set of shares Sh_S is independent of Sh_T. Hence, Sh_S and Sh'_S are distributed identically for every fixed $(s^1, \cdots, s^h, m_1, \cdots, m_n)$ (Sh'_S is the re-sampled distribution from the conditional independence definition 6).
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 $G_{\mathcal{L},K} \in \mathcal{J}^{X,\psi,\tau}$ (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 $\mathsf{aux}_{K,S}(m; rand) = s^1, \cdots, s^h$, where $s^1, \cdots, s^h$ are the seeds of the extractors used.
The proof of the above claim follows almost exactly from the proof of Theorem 3 with the small observation that all the hybrids in the proof could also output the seeds $s^1, \cdots, s^h$ at the end of all

the queries. In all the reduction games, observe that the seeds $s^1, \cdots, s^h$ can always be obtained by the reduction game (in the end) and hence it can complete the simulation, by forwarding the seeds 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.

C.3.2 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 7. *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 13. *[3] 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* $(\mathsf{L}^{m_1}, \mathsf{R}^{m_1}) \leftarrow \mathsf{Enc}(m_1)$.

C.3.3 Instantiations of our Building Blocks

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

- A 2-split-state ϵ_1-non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ (as defined in Section C.3.2), 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 [19] $(\mathsf{LRShare}_{(t,n)}^1, \mathsf{LRRec}_{(t,n)}^1)$, with joint and adaptive leakage model $\mathcal{J}^{X,\psi,\tau_1}$ for t-threshold access structure for message space $\{0,1\}^{\beta_1}$ with conditional independence (as in Definition 6). This means that the adversary can make

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

leakage queries on any $n-t+1$ shares adaptively and jointly, with leakage threshold τ_1 (as interpreted in $\mathcal{J}^{X,\psi,\tau_1}$) 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[20] $(\mathsf{LRShare}^2_{(t-1,n)}, \mathsf{LRRec}^2_{(t-1,n)})$, with joint and adaptive leakage model $\mathcal{J}^{X,\psi,\tau_2}$ for message space $\{0,1\}^{\beta_2}$ with conditional independence. This means that the adversary can make leakage queries on any $n-t+2$ shares adaptively and jointly, with leakage threshold τ_2 (as interpreted in $\mathcal{J}^{X,\psi,\tau_2}$) 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.

C.4 Construction

We describe the construction formally in Figure 8. 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:
$$(\mathsf{L}_1, \cdots, \mathsf{L}_n) \leftarrow \mathsf{LRShare}^1_{(t,n)}(\mathsf{L})$$
$$(\mathsf{R}_1, \cdots, \mathsf{R}_n) \leftarrow \mathsf{LRShare}^2_{(t-1,n)}(\mathsf{R})$$
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:
$$\mathsf{L} := \mathsf{LRRec}^1_{(t,n)}(\mathsf{L}_{i_1}, \cdots, \mathsf{L}_{i_t})$$
$$\mathsf{R} := \mathsf{LRRec}^2_{(t-1,n)}(\mathsf{R}_{i_1}, \cdots, \mathsf{R}_{i_{t-1}})$$
3. Output $m := \mathsf{Dec}(\mathsf{L}, \mathsf{R})$

Figure 8: LRNMSS Construction

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

Theorem 4. *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 LRSS schemes as in Section C.3.3, then the construction given in Figure 8 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 $\mathcal{F}^{n-t-2,\tau}_{tamper}$.*

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

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:

- 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 $\mathsf{Hyb}_0 \equiv \mathsf{Share}(m_0)_T \approx_{2\cdot\epsilon'_3+\epsilon'_4} \mathsf{Hyb}_2 \equiv \mathsf{Share}(m_1)_T$

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 $(G, \mathcal{L}, f_1, \cdots, f_n, I) \in \mathcal{F}^{n-t-2,\tau}_{tamper}$.

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

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 $|\mathcal{L}| \leq n - t - 2$
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 6, 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 6)
     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,

$\mathsf{NMTamper}^{m}_{F_h,G_h}$.

Claim 3. $Copy(\mathsf{Sim}^{G,\mathcal{L},f_1,\cdots,f_n,I}, m) \approx_{\epsilon_1} \mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_n,I}$

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 $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_n,I}$.

Claim 4. $\mathsf{Hyb}_1^{G,\mathcal{L},f_1,\cdots,f_n,I} \approx_{\epsilon_3} \mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_n,I}$

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 $(\mathsf{LRShare}^1_{(t,n)}, \mathsf{LRRec}^1_{(t,n)})$ (as in Defintion 6):

1. Generate $(\mathsf{L}, \mathsf{R}) \leftarrow \mathsf{Enc}(m)$ and $(\mathsf{L}^{\$}, \mathsf{R}^{\$}) \leftarrow \mathsf{Enc}(m^{\$})$.
2. Further generate $(\mathsf{R}_1^{\$}, \cdots, \mathsf{R}_n^{\$}) \leftarrow \mathsf{LRShare}^2_{(t-1,n)}(\mathsf{R}^{\$}; r_R)$ and $\mathsf{aux}^2 \leftarrow \mathsf{aux}^2_{\{i_3,\cdots,i_{t-1}\},\{i_1,i_2\}}(\mathsf{R}^{\$}; r_R)$.
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 $\{\mathsf{L}^b_{i_1}, \cdots, \mathsf{L}^b_{i_{t-1}}\}$. Further, it also receives aux^1_b from the leakage resilience challenger. Now, evaluate $(\widetilde{\mathsf{L}^b_{i_j}}, \widetilde{\mathsf{R}^{\$}_{i_j}}) = f_{i_j}(\mathsf{L}^b_{i_j}, \mathsf{R}^{\$}_{i_j}, \mathsf{Leak}_b)$, for each $j = 1, \cdots, t-1$.
7. Set $h := (\{\mathsf{L}^b_{i_j}, \widetilde{\mathsf{L}^b_{i_j}}\}_{j=1,\cdots,t-1}, \{\mathsf{R}^{\$}_{i_j}, \widetilde{\mathsf{R}^{\$}_{i_j}}\}_{j=3,\cdots,t-1}, \mathsf{R}^{\$}_{i_t}, \mathsf{aux}^1_b, \mathsf{aux}^2, \mathsf{Leak}_b)$.
8. Now the reduction outputs $\tilde{m} \leftarrow \mathsf{NMTamper}^{m}_{F_h,G_h}$, 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 $|\mathcal{L}| \leq n-t-2 < n-t+1$ 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 $\mathcal{J}^{X,\psi,\tau_1}$, 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.

Claim 5. $\mathsf{Hyb}_2^{G,\mathcal{L},f_1,\cdots,f_n,I} \equiv \mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_n,I}$

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 $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_n,I}$.

Claim 6. $\mathsf{Hyb}_3^{G,\mathcal{L},f_1,\cdots,f_n,I} \approx_{\epsilon_4} \mathsf{Hyb}_4^{G,\mathcal{L},f_1,\cdots,f_n,I}$

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 6):

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 $\mathsf{aux}^1 \leftarrow \mathsf{aux}^1_{\{i_1,\cdots,i_{t-1}\},\{i_t\}}(\mathsf{L}; r_L)$[21].
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_k's 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: $\{\mathsf{R}_{i_1}^b, \mathsf{R}_{i_2}^b\}$

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

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

[21] 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 $|\mathcal{L}| + 2 \leq (n-t-2)+2 = n-t < n-t+2$ 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 $\mathcal{J}^{X,\psi,\tau_2}$, 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 $\mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_3$ and else, if it uses R, then the reduction output is identical to $\mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_4$. Hence, this breaks the leakage resilience of $(\mathsf{LRShare}^2_{(t-1,n)}, \mathsf{LRRec}^2_{(t-1,n)})$. □

$\underline{\mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_5}$: Finally, we repeat what we did in $\mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_3$ 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.

Claim 7. $\mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_4 \equiv \mathsf{Hyb}^{G,\mathcal{L},f_1,\cdots,f_n,I}_5$

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

C.5 Rate Analysis

We instantiate our leakage resilient non-malleable secret sharing construction for $\mathcal{F}^{n-t-2,\tau}_{tamper}$ with the following underlying primitives:

- We use the constant rate 2-split-state non-malleable code of [5]:

  Theorem 5. *[5] There exists an efficient, information-theoretically secure ϵ-non-malleable code in the 2-split-state model with rate $\mathcal{O}(1)$ and error $\epsilon = 2^{-k^{\Omega(1)}}$, where k is the message length.*

  Hence, for $|m| = k$, we get $\epsilon_1 = 2^{-k^{\Omega(1)}}$, $|\mathsf{L}| = \mathcal{O}(k)$ bits and $|\mathsf{R}| = \mathcal{O}(k)$ 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 4.5, specifically for $n = \Theta(t)$ and a constant X (which gives a rate of $\mathcal{O}(1)$), with leakage thresholds $\tau_1 = \tau$ and $\tau_2 = \tau + \eta_1$ respectively ($\eta_1 = |\mathsf{L}_i|$). This gives us that $\eta_1 = |\mathsf{L}_i| = \mathcal{O}(|\mathsf{L}|) = \mathcal{O}(k)$ and $\eta_2 = |\mathsf{R}_i| = \mathcal{O}(k)$. Further $\epsilon_3 = 2^{-\Omega(k)}$ and $\epsilon_4 = 2^{-\Omega(k)}$ [22].

Combining these two instantiations, we get: $|\mathsf{Sh}_i| = |\mathsf{L}_i| + |\mathsf{R}_i| = \mathcal{O}(k)$ and hence, we get the rate of $\Omega(1)$. The error is $\epsilon_1 + \epsilon_3 + \epsilon_4 = 2^{-k^{\Omega(1)}}$.
We obtain the following corollary:

Corollary 1. *For any $n \in \mathbb{N}$, there exists a leakage resilient non-malleable secret sharing scheme against $\mathcal{F}^{n-t-2,\tau}_{tamper}$ with rate $\Omega(1)$ and simulation error $\epsilon + 2^{-k^{\Omega(1)}}$.*

D Leakage Resilient and Non-malleable Secure Message Transmission

The problem of perfectly secure message transmission (SMT) was introduced in [18], 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 [22], 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 [18, 35, 38, 29, 26] nor the non-malleable SMT [22] 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.

D.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.

[22] 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 $\mathcal{O}(k)$

Definition 8 (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 $m, m' \in \mathcal{M}$,*

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

where $\pi^{view}_{\mathcal{A}}(m, S, R)$ 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 9 (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 $n - t + 1$ wires, jointly and adaptively and then see the messages sent through the remaining $t - 1$ fresh wires (on which leakage queries were not made) in clear, exactly like our LRSS leakage model (section 4.5), $\mathcal{J}^{X,\psi,\tau}$. Clearly, this model is stronger than the standard statistical privacy in definition 8. We denote this leakage family by $\mathcal{F}^{leak}_{t,\tau}$. Formally, this model is defined by taking the joint and adaptive leakage model $\mathcal{J}^{X,\psi,\tau}$ of our LRSS scheme, for the t-threshold access structure, and replacing the role of the shares $share_1, \cdots, share_n$ in the queries in $\mathcal{J}^{X,\psi,\tau}$ 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 $\mathcal{F}^{leak}_{t,\tau}$.

D.1.1 Construction:

Let $(\mathsf{LRShare}_{(t,n)}, \mathsf{LRRec}_{(t,n)})$ be a $(t, n, \epsilon_s, \epsilon_l)$-LRSS against $\mathcal{J}^{X,\psi,\tau}$ (from section 4.5). 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 6. *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 $\mathcal{J}^{X,\psi,\tau}$ (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 $\mathcal{F}^{leak}_{t,\tau}$ 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 $f \in \mathcal{F}^{leak}_{t,\tau}$ 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, $f \in \mathcal{J}^{X,\psi,\tau}$. 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 $f \in \mathcal{F}^{leak}_{t,\tau}$, $\mathbf{SD}\left(f(\pi^{view}(m, S, R)), f(\pi^{view}(m', S, R))\right) \leq \epsilon_l$.

Communication Cost. By theorem 3, taking X to be constant and $n = \Theta(t)$, we get a constant-rate LRSS, and hence the LRSS shares are each of size $O(\log_2(|\mathcal{M}|))$. Thus, we get the desired communication for our message transmission protocol. □

D.2 Leakage Resilient Non-malleable Message Transmission

The communication model is exactly as described above in section D: 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 [22], 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 $n - t - 2$ wires, jointly and adaptively (i.e., queries can be combined leakage on non-overlapping subsets of wires, of size upto $n - t - 2$, 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[23]. Finally, the receiver recovers a modified message

[23]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.

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 $\mathcal{F}_{tamper}^{n-t-2,\tau}$ of our LRNMSS scheme (section C.1), 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 $\mathcal{F}_{tamper}^{n-t-2,\tau}$). 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.

Definition 10 (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

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

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.

D.2.1 Construction:

We consider the same construction of the message transmission protocol as for the leakage resilient case (section D.1.1), with the only difference that we use the $(t,n,\epsilon_s,\epsilon_{nm})$-LRNMSS, $(\mathsf{Share},\mathsf{Rec})$ against $\mathcal{F}_{tamper}^{n-t-2,\tau}$ (from section C.4) to generate the shares $(share_1,\cdots,share_n)\leftarrow\mathsf{Share}(m)$ (instead of the LRSS).

Theorem 7. *Let $n \in \mathbb{N}$, $t \in [n]$ and $\mathcal{M}$ be the message space. If $(\mathsf{Share}, \mathsf{Rec})$ is a $(t, n, \epsilon_s, \epsilon_{nm})$-LRNMSS against $\mathcal{F}_{tamper}^{n-t-2,\tau}$ (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 $(G, \mathcal{L}, \{f_i\}_{i\in[n]\setminus\mathcal{L}}, I) \in \mathcal{F}_{tamper}^{n-t-2,\tau}$). 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 $m \in \mathcal{M}$, $\mathbf{SD}\left(\mathsf{Tamper}_m^{G,\mathcal{L},\{f_i\}_{i\in[n]\setminus\mathcal{L}},I}, Copy(\mathsf{Sim}^{G,\mathcal{L},\{f_i\}_{i\in[n]\setminus\mathcal{L}},I}, n\right.$ ϵ_{nm}.

Communication Cost. By corollary 1, our LRNMSS scheme can be instantiated to have rate $O(1)$, and hence the LRNMSS shares are each of size $O(\log_2(|\mathcal{M}|))$. Thus, we get the desired communication for our message transmission protocol. □