Multi-Source Non-Malleable Extractors and Applications

Vipul Goyal* Akshayaram Srinivasan† Chenzhi Zhu‡

March 5, 2021

Abstract

We introduce a natural generalization of two-source non-malleable extractors (Cheragachi and Guruswami, TCC 2014) called as *multi-source non-malleable extractors*. Multi-source non-malleable extractors are special independent source extractors which satisfy an additional non-malleability property. This property requires that the output of the extractor remains close to uniform even conditioned on its output generated by tampering *several sources together*. We formally define this primitive, give a construction that is secure against a wide class of tampering functions, and provide applications. More specifically, we obtain the following results:

- For any $s \geq 2$, we give an explicit construction of a s-source non-malleable extractor for min-entropy $\Omega(n)$ and error $2^{-n^{\Omega(1)}}$ in the *overlapping joint tampering model*. This means that each tampered source could depend on any strict subset of all the sources and the sets corresponding to each tampered source could be overlapping in a way that we define. Prior to our work, there were no known explicit constructions that were secure even against disjoint tampering (where the sets are required to be disjoint without any overlap).
- We adapt the techniques used in the above construction to give a t-out-of-n non-malleable secret sharing scheme (Goyal and Kumar, STOC 2018) for any $t \leq n$ in the *disjoint tampering model*. This is the first general construction of a threshold non-malleable secret sharing (NMSS) scheme in the disjoint tampering model. All prior constructions had a restriction that the size of the tampered subsets could not be equal.
- We further adapt the techniques used in the above construction to give a t-out-of-n non-malleable secret sharing scheme (Goyal and Kumar, STOC 2018) for any $t \leq n$ in the *overlapping joint tampering model*. This is the first construction of a threshold NMSS in the overlapping joint tampering model.
- We show that a stronger notion of s-source non-malleable extractor that is multi-tamperable against disjoint tampering functions gives a single round network extractor protocol (Kalai

*NTT Research and CMU, Email:vipul@cmu.edu. Supported in part by the NSF award 1916939, DARPA SIEVE program, a gift from Ripple, a DoE NETL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cylab seed funding award.

†Tata Institute of Fundamental Research. Email: akshayaram.srinivasan@tifr.res.in. Work partially done while at UC Berkeley and supported in part from AFOSR Award FA9550-19-1-0200, AFOSR YIP Award, NSF CNS Award 1936826, DARPA and SPAWAR under contract N66001-15-C-4065, a Hellman Award and research grants by the Okawa Foundation, Visa Inc., and Center for Long-Term Cybersecurity (CLTC, UC Berkeley). The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies. Work partially done while visting CMU.

‡Tsinghua University, Email: mrbrtpt@gmail.com. Work partially done while visiting CMU.

et al., FOCS 2008) with attractive features. Plugging in with a new construction of multi-tamperable, 2-source non-malleable extractors provided in our work, we get a network extractor protocol for min-entropy $\Omega(n)$ that tolerates an *optimum* number ($t = p - 2$) of faulty processors and extracts random bits for *every* honest processor. The prior network extractor protocols could only tolerate $t = \Omega(p)$ faulty processors and failed to extract uniform random bits for a fraction of the honest processors.

1 Introduction

Non-Malleable Extractors. Randomness extractors are fundamental objects in the study of computer science and combinatorics. They allow to extract uniform random bits from a source that has "some" randomness which may not necessarily be uniform. The amount of randomness in a source X is captured by the notion of min-entropy defined as $H_\infty(X) = \min_{s \in \mathsf{sup}(X)}\{\log \frac{1}{\Pr[X=s]}\}$. It is well-known that if we only have a single source with min-entropy less than full, then it is impossible to extract uniform random bits out of this source. One way to get around this impossibility result is to assume that we have two or more sources that are independent and the goal is to extract uniform random bits from these independent sources. Such extractors are called as multi-source (or independent source) extractors. A long line of work starting from the seminal work of Chor and Goldreich [CG88] have focused on constructing multi-source extractors for lower min-entropy. This recently resulted in a breakthrough work of Chattopadhyay and Zuckerman showing explicit constructions of two-source extractors for poly logarithmic min-entropy [CZ16]. See also the follow-up works of [Li16, Li17a, BDT17, GKK19].

A natural strengthening of multi-source extractors (that have also been used as a key tool in the recent breakthroughs) is the notion of a non-malleable extractor [CG14]. Roughly speaking, non-malleable extractors require that the output of the extractor (when run on independent sources) to be statistically close to uniform even conditioned on the output of the extractor generated by tampered version of the sources. Formally, we say that a s-source extractor is non-malleable against a tampering function family $\mathcal{F}$ if for any set of s independent sources $X_1, \ldots, X_s$ with sufficient min-entropy and for any tampering function $f \in \mathcal{F}$, there exists a distribution D_f with support in $\{0,1\}^m \cup \{\mathsf{same}^*\}$ that is independent of $X_1, \ldots, X_s$ such that:

$$|\mathsf{MNMExt}(X_1, \ldots, X_s) \circ \mathsf{MNMExt}(f(X_1, \ldots, X_s)) - U_m \circ \mathsf{copy}(D_f, U_m)| \leq \varepsilon$$

Here, $\mathsf{copy}(x, y) = x$ if $x \neq \mathsf{same}^*$; else, it is equal to y and $|X - Y|$ denotes the statistical distance between the random variables X and Y. Such extractors have wide applications in computer science and specifically, in cryptography; in particular, they can be used to construct two-source extractors [CZ16], non-malleable codes [DPW18, CG14, CGL16], non-malleable secret sharing [GK18a], round-optimal non-malleable commitments [GPR16, GKP+18] and cryptography with correlated random tapes [GS19].

Almost all of the prior work in constructing non-malleable multi-source extractors have focused on protecting against tampering functions that tamper each of the sources independently (aka individual tampering family). In this work, we are interested in constructing multi-source extractors that are secure against richer classes of tampering functions that could tamper several sources together. For the case of two sources (that has been the focus of the majority of the prior work), any tampering function that can tamper with both the sources can easily break the non-malleability property and hence, the individual tampering is the best that one could hope for. However, this is not the case for more than two sources.

Non-Malleable Secret Sharing. Non-malleable secret sharing introduced in the work of Goyal-Kumar [GK18a] strengthens the traditional secret sharing with an additional non-malleability property. Specifically, in addition to the standard correctness and privacy properties, a non-malleable secret sharing scheme requires that any tampering attack from a family of allowable tampering functions either preserves the original secret that was shared or completely destroys it. Most of the works in this area [BS19, SV19, ADN+19, KMS18, FV19] focused on constructing non-malleable secret sharing against the individual tampering setting. Specifically, these constructions become insecure even if a tampering function can tamper with two shares together. The work of Goyal and Kumar [GK18a] gave a construction of t-out-of-n non-malleable secret sharing in a restricted version of the disjoint tampering model. Here, the tampering function first chooses a set of t shares, then partitions this share into two sets of unequal sizes and then tampers each partition independently. *It was crucial to their security analysis that the partitions are of unequal size and this construction does not work for equal size partitions.* In [GK18b], this assumption was removed for the specific case of $t = n$ and a construction that was secure in the overlapping joint tampering model with cover-free subsets (the exact description of this model can be found in Section 1.1) was given. However, the construction and the analysis crucially rely on the fact that $t = n$ and does not work for any $t < n$. Despite a number of follow up works, overcoming this restriction for threshold NMSS has remained an open problem. This brings us to the following questions.

Can we construct a threshold non-malleable secret sharing scheme secure in the disjoint tampering model (without restriction on the size of tampering sets)?

Can we construct a threshold non-malleable secret sharing scheme in the overlapping joint tampering model?

Network Extractor Protocols. Network extractor [DO03, GSV05, KLRZ08, KLR09] is a protocol between p processors, each starting with an independent source X_i of length n with min-entropy k. The processors exchange some messages during the protocol and these messages are sent over public channels. At the end of the protocol, we require each (honest) processor to end up with an independent (statistically close to) uniform string. We require this guarantee to hold even in the face of a centralized adversary who can corrupt a set of processors and instruct these processors to arbitrarily deviate from the protocol specification (byzantine corruptions). Such network extractor protocols can be run prior to any secure multiparty computation protocol or distributed computation protocols where the honest parties necessarily require private uniform random bits but they only start with independent sources with some min-entropy.

Formally, if B is the random variable denoting all the messages exchanged during the protocol and Z_i is the random variable denoting the output of the i-th processor, then the definition of a network extractor protocol is as follows.

Definition 1.1 (Network Extractor Protocol [KLRZ08]) *A protocol for p processors is a (t, g, ε) network extractor for min-entropy k if for any (n, k) independent sources $X_1, \ldots, X_p$ and any choice T of t faulty processors, after running the protocol, there exists a set $G \in [p] \setminus T$ of size at least g such that*

$$|B, \{X_i\}_{i \notin G}, \{Z_i\}_{i \in G} - B, \{X_i\}_{i \notin G}, U_{gm}| < \varepsilon$$

Here U_{gm} is the uniform distribution on gm bits, independent of B, and $\{X_i\}_{i \notin G}$.

It is easy to see that if we allow the adversary to corrupt $p-1$ processors then this task is impossible as it amounts to extracting random bits from a single source. Kalai et al. [KLRZ08] gave a $(t=\Omega(p), p-(1+O(1))t, 2^{-n^{\Omega(1)}})$-network extractor protocol for min-entropy $k=(1/2+O(1))n$. This protocol required a single round of interaction. They also showed another multi-round protocol for lower min-entropy (specifically, $k=2^{\log^{\beta} n}$ for some $\beta<1$) but in this protocol, a smaller number of honest processors end up with a uniform string. Li [Li13] further improved this result and gave a 2-round network extractor protocol for $k \geq \log^{c} n$. However, all these protocols only allow an adversary to corrupt $\Omega(p)$ processors and additionally, there exists a fraction of the honest processors whose output is not statistically close to uniform. This brings us to the next question.

Can we construct a network extractor protocol where the adversary can corrupt upto $p-2$ processors and the protocol ensures that every honest processor ends up with a uniform output?

We note that in the computational setting, the work of Kalai et al. [KLR09] gave a protocol satisfying both the properties assuming sub-exponential hardness of one-way permutations.

Our work. In this work, we provide positive answers to the question on non-malleable secret sharing as well as the network extractor protocols by viewing them through the lens of multi-source non-malleable extractors. The details follow.

1.1 Our Results

In this work, we initiate the systematic study of multi-source non-malleable extractors and give constructions that are secure against a wide class of tampering function families. We also show applications of this primitive in constructing non-malleable codes [DPW18], non-malleable secret sharing [GK18a], and network extractor protocols [DO03, GSV05, KLRZ08, KLR09]. Before we state the formal theorem statements, we first describe the tampering functions of interest.

Overlapping Joint Tampering. For any $s \in \mathbb{N}$, the overlapping joint tampering family is given by a sequence of sets $(T_1, \ldots, T_s)$ where $T_s \subset [s]$ and the associated tampering functions $(f_{T_1}, \ldots, f_{T_s})$. The i-th tampered source $\widetilde{X}_i$ is generated by applying f_{T_i} on the sources $\{X_j\}_{j \in T_i}$. In other words, the tampered source $\widetilde{X}_i$ is generated by tampering all the sources indexed by the set T_i using the function f_{T_i}.

We say that $(T_1, \ldots, T_s)$ are *cover-free*, if for every $i \in [s]$, the union of all T_j such that $i \in T_j$ has size at most $s-1$. Some examples of cover-free subsets are:

- **Individual Tampering:** This is the setting where $T_i = \{i\}$.
- **Disjoint Tampering:** Here, $(T_1, \ldots, T_s)$ are such that for each $i, j \in [s]$, either $T_i = T_j$ or $T_i \cap T_j = \emptyset$.
- **Cycles of size at most $\lfloor s/2 \rfloor$:** Here, $T_i = \{i, i+1 \mod s, \ldots, i+\lfloor s/2 \rfloor - 1 \mod s\}$.

Cover-free subsets include a rich class of joint tampering functions and it strictly generalizes the individual tampering functions considered in the previous works. In this work, we focus on constructing multi-source non-malleable extractors in the overlapping joint tampering model with cover-free subsets (cover-free tampering, in short). We note that prior to our work, no construction of non-malleable extractors was known even in the disjoint tampering model.

Multi-source Non-malleable Extractors. Our first result in this paper is a construction of multi-source non-malleable extractors that are secure against cover-free tampering. The formal theorem statement appears below.

Theorem 1.2 *For any $s \geq 2$, there exists a constants $\gamma > 0$ and n_0 such that for any $n > n_0$, there exists an efficient construction of a s-source, non-malleable extractor* $\mathsf{MNMExt} : (\{0,1\}^n)^s \to \{0,1\}^m$ *against cover-free tampering at min-entropy $n(1-\gamma)$ and error $2^{-n^{\Omega(1)}}$ with output length $m = n^{\Omega(1)}$.*

We note that extending the class of tampering functions beyond cover-free tampering requires a new set of tools as there are sources which are tampered together with every other source. We leave open the fascinating problem of constructing explicit extractors that are secure against a generalization of cover-free tampering.

Split-state Non-malleable codes. We show that (a variant of) our multi-source extractor is efficiently pre-image sampleable, meaning that there exists an efficient algorithm such that given any string of length m, the algorithm outputs (except with negligible probability) an uniform pre-image of this string. This feature combined with a straightforward generalization of the result of Cheraghchi and Guruswami [CG14] gives the following theorem.

Theorem 1.3 *For any $s \geq 2$ and $m \in \mathbb{N}$, there exists an efficient construction of s-split-state non-malleable code for messages of length m that is secure against cover-free tampering with error $2^{-m^{\Omega(1)}}$.*

This result is a conceptual contribution as we already know constructions of s-split state non-malleable codes against cover-free tampering from the work of [GK18b]. However, as we will see below this construction leads to a t-out-of-n non-malleable secret sharing in the overlapping joint tampering model.

1.1.1 Non-malleable Secret Sharing

An interesting aspect of our construction of multi-source non-malleable extractor is that a minor modification to this construction gives a t-out-of-n non-malleable secret sharing against t-cover-free tampering. t-cover free tampering is the same as cover-free tampering defined above except that we require that for every i, the union of all T_j's such that $i \in T_j$ has size at most $t-1$. As before, t-cover-free tampering includes disjoint tampering where each partition is of size at most $t-1$. We note if any set of t or more shares are tampered together, then the tampering function can trivially reconstruct the secret and hence, obtaining non-malleability is impossible. The formal statement about our construction is given below.

Theorem 1.4 *For every $t \geq 2$, $n \geq t$ and $m \in \mathbb{N}$, there exists an efficient construction of t-out-of-n non-malleable secret sharing for secrets of length m against t-cover-free tampering with error $2^{-m^{\Omega(1)}}$.*

As a corollary, we get a construction of t-out-of-n non-malleable secret sharing in the disjoint tampering model.

Corollary 1.5 *For every $t \geq 2$, $n \geq t$ and $m \in \mathbb{N}$, there exists an efficient construction of t-out-of-n non-malleable secret sharing for secrets of length m in the disjoint tampering model with error $2^{-m^{\Omega(1)}}$.*

As mentioned before, this is the first construction of threshold NMSS in the disjoint tampering model without restriction on the size of the tampering sets. This answers an explicit open problem from the work of Goyal and Kumar [GK18a]. In addition, ours is also the first construction of threshold NMSS in the overlapping joint tampering model. The only previous construction of NMSS in the overlapping joint tampering model was for n-of-n secret sharing [GK18b].

1.1.2 Network Extractor Protocols

For any $s \geq 2$, we show that a stronger notion of s-source non-malleable extractor that is multi-tamperable and whose non-malleability property holds even conditioned on all but one of the sources implies a single round network extractor protocol with at least s honest processors. It is sufficient for such multi-source non-malleable extractors to be resilient against a weaker form of disjoint tampering. For the case of 2 sources, we give a compiler that transforms a single tamperable non-malleable extractor to a multi-tamperable non-malleable extractor by building on the ideas of Cohen [Coh16a] who gave such a compiler for seeded non-malleable extractors. This result might be of independent interest. We show that the resultant extractor is sufficient to instantiate the network extractor protocol. This leads to a single round network extractor protocol that is resilient against an optimum number of byzantine corruptions of $p - 2$ (where p is the total number of processors) and ensures that all the honest processors end up with a string that is statistically close to uniform. Specifcially, we show the following result.

Theorem 1.6 *For any $p \geq 2$, there exists constants $\gamma > 0$ and n_0 such that for all $n > n_0$ and for any $t \leq p - 2$, there exists a single-round, $(t, p - t, 2^{-n^{\Omega(1)}})$-network extractor protocol for p processors and $(n, n(1 - \gamma))$ sources.*

We note that all the prior information-theoretic network extractor protocols could only tolerate $\Omega(p)$ number of byzantine corruptions and furthermore, these protocols could not extract uniform randomness for a $\Omega(t)$ number of honest processors. Our protocol tolerates an optimum number of corruptions and ensures that every honest processor outputs a string that is statistically close to uniform. This matches the best protocols known in the computational setting [KLR09] that relied on sub-exponential hardness assumptions but has weaker min-entropy requirements.

2 Technical Overview

In this section, we give a high-level overview of the techniques used in obtaining our main results. We start our overview with the construction of multi-source non-malleable extractors. Then, we will extend this result to obtain a non-malleable secret sharing. Finally, we give the description of our network extractor protocol.

2.1 Multi-source Non-malleable Extractor

An s-source non-malleable extractor $\mathsf{MNMExt} : (\{0,1\}^n)^s \to \{0,1\}^m$ is just like any other independent source extractor with an additional non-malleability property. Recall that an s-source

extractor is said to be non-malleable against the tampering function family $\mathcal{F}$ if for any set of s independent sources $X_1, \ldots, X_s$ with sufficient min-entropy and for any tampering function $f \in \mathcal{F}$, there exists a distribution D_f with support in $\{0,1\}^m \cup \{\mathsf{same}^*\}$ that is independent of $X_1, \ldots, X_s$ such that:

$$|\mathsf{MNMExt}(X_1, \ldots, X_s) \circ \mathsf{MNMExt}(f(X_1, \ldots, X_s)) - U_m \circ \mathsf{copy}(D_f, U_m)| \leq \varepsilon$$

Here, $\mathsf{copy}(x, y) = x$ if $x \neq \mathsf{same}^*$; else, it is equal to y. A standard two-source non-malleable extractor is a special case of a multi-source extractor that is secure against the independent tampering family. Furthermore, it can be shown that any two-source non-malleable extractor implies an s-source non-malleable extractor for any $s \geq 2$ where each of the s-sources are tampered independently. However, in this work, we are interested in designing multi-source non-malleable extractors that are secure against richer forms of tampering where several sources can potentially be tampered together. In such a scenario, the trivial construction of extending any two-source extractor to an s-source extractor is insecure.

To explain the key ideas behind our construction without getting bogged down with the details, let us make the following simplifying assumptions. We stress that our actual construction does not make any of the following assumptions.

- Let us assume that there are only 3 sources X_1, X_2 and X_3 and each of the sources have full min-entropy. Even when the sources have full entropy, non-malleable extractors are known to imply non-malleable codes [CG14].
- We are interested in protecting against tampering functions that tamper two sources together and tampers the other source independently. The identity of the two sources that are tampered together is not fixed apriori. Specifically, we assume that the tampering functions are given by (f_{ij}, g_k) for distinct $i, j, k \in [3]$ where f_{ij} takes in sources X_i, X_j and outputs $\widetilde{X}_i, \widetilde{X}_j$. Similarly, g_k takes in X_k and outputs $\widetilde{X}_k$.

A Simple construction. A natural attempt at constructing a multi-source non-malleable extractor is to take any 2 source non-malleable extractor $\mathsf{2NMExt}$ and output $\mathsf{2NMExt}(X_1 \circ 1, X_2 \circ 2) \oplus \mathsf{2NMExt}(X_2 \circ 2, X_3 \circ 3) \oplus \mathsf{2NMExt}(X_3 \circ 3, X_1 \circ 1)$ where $\circ$ denotes concatenation. Recall that our tampering functions satisfy the property that for every source there exists at least one other source that is not tampered together with this source. Since the above construction applies a non-malleable extractor for every pair of sources, we can hope to reduce the security of this construction to the security of the underlying non-malleable extractor. However, proving this is not straightforward as the tampering function may not modify these two sources and thus, proving independence between the tampered output and the untampered output is tricky. Nevertheless, with some non-trivial work, we can show using the techniques developed in [CGGL19] (for completeness, we provide a full proof in Appendix B) that this construction is indeed secure against cover-free tampering if the underlying non-malleable extractor is multi-tamperable[1] and is symmetric (meaning that $\mathsf{2NMExt}(x, y) = \mathsf{2NMExt}(y, x)$ for every x, y). However, a major problem

[1] A multi-tamperable non-malleable extractor introduced in [CGL16] considers several sets of split-state tampering functions and requires the output of the extractor to be random even conditioned on all the tampered outputs generated by each split-state tampering function. An equivalent way to view the multi tamperable (or, t tamperable) non-malleable extractor is to allow the split-state tampering functions to have t sets of outputs and we require the real output to be close to random even conditioned on joint distribution of the t tampered outputs.

with this simple construction is that it is *not efficiently* pre-image sampleable. Recall that for a non-malleable extractor to be efficiently pre-image sampleable, we need an efficient algorithm that given any output of the non-malleable extractor, samples an uniform pre-image of this output. This property is crucially needed to construct a s-split state non-malleable code from non-malleable extractors using the approach of Cheraghchi and Guruswami [CG14]. To see why this construction is not efficiently pre-image sampleable, consider any output $s \in \{0,1\}^m$ of the extractor. Now, we need to sample three sources, X_1, X_2, X_3 such that $\mathsf{2NMExt}(X_1 \circ 1, X_2 \circ 2) \oplus \mathsf{2NMExt}(X_2 \circ 2, X_3 \circ 3) \oplus \mathsf{2NMExt}(X_3 \circ 3, X_1 \circ 1) = s$. Even if we assume that $\mathsf{2NMExt}$ is efficiently pre-image sampleable, fixing any two sources, say X_1, X_2, requires the third source to satisfy the equation $\mathsf{2NMExt}(X_2 \circ 2, X_3 \circ 3) \oplus \mathsf{2NMExt}(X_3 \circ 3, X_1 \circ 1) = s \oplus \mathsf{2NMExt}(X_1 \circ 1, X_2 \circ 2)$. Efficiently sampling from the set of such X_3's seems highly non-trivial. This seems to be a major roadblock with this simple construction (and is crucial to obtain our main application in constructing non-malleable secret sharing) and hence, it calls for a more sophisticated construction that is efficiently pre-image sampleable.

A Starting Point. In order to construct a multi-source non-malleable extractor with efficient pre-image sampling, we could try to make the following generalization. We can parse the sources X_1 as $(X^{(1)}, Y^{(3)})$, X_2 as $(X^{(2)}, Y^{(1)})$, X_3 as $(X^{(3)}, Y^{(2)})$ and output $\oplus_i \mathsf{2NMExt}(X^{(i)}, Y^{(i)})$. This construction is efficiently pre-image sampleable since the inputs to each invocation of the underlying $\mathsf{2NMExt}$ is "non-overlapping". Specifically, given any output $s \in \{0,1\}^m$, we can sample $X^{(1)}$, $Y^{(1)}, X^{(2)}, Y^{(2)}$ uniformly at random and sample $X^{(3)}, Y^{(3)}$ such that $\mathsf{2NMExt}(X^{(3)}, Y^{(3)}) = s \oplus \mathsf{2NMExt}(X^{(2)}, Y^{(2)}) \oplus \mathsf{2NMExt}(X^{(2)}, Y^{(2)})$. This process is efficient if the underlying $\mathsf{2NMExt}$ has efficient pre-image sampling. This seems like progress but unfortunately, we prove this construction is insecure. In particular, consider any tampering function that tampers X_1, X_2 together. Such a tampering function takes as input $(X^{(1)}, Y^{(3)})$ and $(X^{(2)}, Y^{(1)})$, leaves $X^{(2)}, Y^{(3)}$ untampered, but tampers $X^{(1)}, Y^{(1)}$ to $\widetilde{X}^{(1)}, \widetilde{Y}^{(1)}$ such that $\mathsf{2NMExt}(\widetilde{X}^{(1)}, \widetilde{Y}^{(1)}) = \overline{\mathsf{2NMExt}(X^{(1)}, Y^{(1)})}$ (where $\overline{z}$ denotes flipping each bit of z). If the tampering function against X_3 is the identity function, then we infer that the real output XORed with the tampered output will be the all 1s string.

Our Construction. If we look a little bit closely into the analysis of the above construction, we realize that the main reason for the attack is that $X^{(1)}, Y^{(1)}$ was available in the clear to one of the tampering functions. However, this attack could have been avoided if every tampering function does not get hold of both $X^{(i)}, Y^{(i)}$ together. With this intuition, we are ready to describe our extractor with efficient pre-image sampleability.

1. Parse X_i as $(X_i^{(1)}, X_i^{(2)}, X_i^{(3)}, Y^{(i)})$.
2. Compute $X^{(i)} = X_1^{(i)} \oplus X_2^{(i)} \oplus X_3^{(i)}$ for each $i \in [3]$.
3. Output $\mathsf{2NMExt}(X^{(1)}, Y^{(1)}) \oplus \mathsf{2NMExt}(X^{(2)}, Y^{(2)}) \oplus \mathsf{2NMExt}(X^{(3)}, Y^{(3)})$.

Notice that any tampering function that looks at any two sources X_i, X_j cannot determine $X^{(i)}$ and $X^{(j)}$ since these are "secret shared" between all the three sources. Furthermore, we observe that this construction has efficient pre-image sampling if the underlying $\mathsf{2NMExt}$ is efficiently pre-image sampleable. This is because for any image $s \in \{0,1\}^m$, we can sample $X^{(2)}, Y^{(2)}$ and $X^{(3)}, Y^{(3)}$ uniformly at random and we sample $X^{(1)}, Y^{(1)}$ conditioned on its output being equal to

$\mathsf{2NMExt}(X^{(2)}, Y^{(2)}) \oplus \mathsf{2NMExt}(X^{(3)}, Y^{(3)}) \oplus s$. Then, for every $i \in [3]$, we sample $X_1^{(1)}, X_2^{(i)}, X_3^{(i)}$ uniformly at random conditioned on its XOR being equal to $X^{(i)}$. This allows to efficiently find the sources X_1, X_2, X_3 such that applying the extractor on these sources yields s. Below, we give the main ideas behind proving the non-malleability of this construction.

Proof Idea. The key technical component of our security proof is a way to reduce the tampering of our extractor to a multi-tampering of the underlying non-malleable extractor $\mathsf{2NMExt}$. However, unlike the simple construction, this reduction is highly non-trivial and it requires the underlying extractor to satisfy a strong leakage-resilience property. The details follow.

Recall that in the tampering functions of our interest, for every source j, there exists at least one other source j^* that is not tampered together with this source. The main trick in the reduction is that we view $X_i^{(j)}$ for every i as a *secret share* of the source $X^{(j)}$. Viewing $X_i^{(j)}$ as a secret share of $X^{(j)}$ allows us to fix all the shares except $X_{j^*}^{(j)}$. Hence, $X_{j^*}^{(j)}$ is completely determined by the source $X^{(j)}$ and the fixed shares. Now, since j and j^* are not tampered together, we infer that $Y^{(j)}$ and $X^{(j)}$ are tampered independently! This allows us to reduce any tampering attack on our extractor to a split-state tampering attack on $\mathsf{2NMExt}$. Thus, relying on this reduction, we can hope to make the tampered output of our extractor to be "independent" of $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$ and thus, conclude that the real output is independent of the tampered output. However, arguing independence is not as straightforward as it seems. Notice that nothing prevents a tampering function from leaving $X^{(j)}, Y^{(j)}$ untampered. In this case, $\mathsf{2NMExt}(\widetilde{X}^{(j)}, \widetilde{Y}^{(j)}) = \mathsf{2NMExt}(X^{(j)}, Y^{(j)})$ and hence, it is impossible to argue that the tampered output is independent of $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$.

To get around this problem, we prove a *weaker property* about our reduction to split-state multi-tampering of $\mathsf{2NMExt}$. Specifically, we show that for every $i, j \in [3]$, the tampered output $\mathsf{2NMExt}(\widetilde{X}^{(i)}, \widetilde{Y}^{(i)})$ is either independent of $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$ (meaning that a non-trivial tampering attack has taken place) or is the same as $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$ (meaning that the tampering function has just copied). This in fact allows us to argue (via a hybrid argument going over every $j \in [\lambda]$)[2] that the tampered tuple $(\mathsf{2NMExt}(\widetilde{X}^{(1)}, \widetilde{Y}^{(1)}),$
$\mathsf{2NMExt}(\widetilde{X}^{(2)}, \widetilde{Y}^{(2)}), \mathsf{2NMExt}(\widetilde{X}^{(3)}, \widetilde{Y}^{(3)}))$ is either a permutation of $(\mathsf{2NMExt}(X^{(1)}, Y^{(1)}), \mathsf{2NMExt}(X^{(2)},$ $Y^{(2)}), \mathsf{2NMExt}(X^{(3)}, Y^{(3)}))$ in which case the adversarial tampering functions have not changed the output of the extractor or there exists at least one j such that the tampered tuple is independent of $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$. This allows us to argue that the real output is independent of the tampered output and it is in fact, close to uniform since $\mathsf{2NMExt}(X^{(j)}, Y^{(j)})$ is close to uniform.

Below, we show a sketch of a proof of this property. This is shown via a reduction to the multi-tampering of the underlying 2-source non-malleable extractor. As mentioned before, for this reduction to go through, we need the underlying non-malleable extractor to satisfy an additional strong leakage resilience property.

The Main Reduction. Let us try to sketch the above reduction for $j = 1$ by considering specific tampering functions f_{12}, g_3. Recall that f_{12} takes X_1, X_2 as input and outputs $\widetilde{X}_1, \widetilde{X}_2$ and g_3 takes X_3 as input and outputs $\widetilde{X}_3$. The goal here is to show that each entry of the tampered tuple $(\mathsf{2NMExt}(\widetilde{X}^{(1)}, \widetilde{Y}^{(1)}), \mathsf{2NMExt}(\widetilde{X}^{(2)}, \widetilde{Y}^{(2)}), \mathsf{2NMExt}(\widetilde{X}^{(3)}, \widetilde{Y}^{(3)}))$ is either equal to $\mathsf{2NMExt}(X^{(1)}, Y^{(1)})$ or independent of this value. As mentioned before, we prove this via a reduction

[2]This is where we need the stronger property that for every source j there exists at least one other source that is not tampered together with this source.

from any tampering attack against our extractor to a split-state tampering attack (f', g') against $X^{(1)}, Y^{(1)}$.

Towards this goal, we will fix $X^{(2)}, Y^{(2)}$, $X^{(3)}, Y^{(3)}$ and all the shares of $X^{(2)}$ and $X^{(3)}$. In addition to this, we will fix the shares $X_1^{(1)}$ and $X_2^{(1)}$. Notice that by the choice of our tampering functions, X_1 and X_3 are tampered independently and thus, by fixing $X_1^{(1)}, X_2^{(1)}$, we have ensured that $X^{(1)}$ and $Y^{(1)}$ are tampered independently. Let us additionally assume that there exists a special string Y^* such that for every $s \in \{0,1\}^m$, there exists an $x \in \{0,1\}^m$ such that $\mathsf{2NMExt}(x, Y^*) = s$ (it will be clear on why this property is needed when we explain our reduction). We show that for any non-malleable extractor with sufficiently low-error, there exists such an Y^*.

Given the fixed values and the string Y^*, designing the multi-tampering function g' against $Y^{(1)}$ is straightforward. On input $Y^{(1)}$, g' uses the fixed values and the input $Y^{(1)}$ to reconstruct the sources X_1, X_2. It then applies f_{12} on these two sources and obtains $\widetilde{X}_1, \widetilde{X}_2$. It now outputs $(\widetilde{Y}^{(1)}, \widetilde{Y}^{(2)}, Y^*)$ (where $\widetilde{Y}^{(1)}, \widetilde{Y}^{(2)}$ are obtained from $\widetilde{X}_1, \widetilde{X}_2$) as the three tampered outputs. However, constructing a tampering function against $X^{(1)}$ is not as straightforward. Notice that the tampering function against $X^{(1)}$ must somehow get $\{\widetilde{X}_1^{(i)}, \widetilde{X}_2^{(i)}, \widetilde{X}_3^{(i)}\}_{i\in[3]}$, XOR them together and finally output the XORed value as the tampered source $\widetilde{X}^{(i)}$. However, $\{\widetilde{X}_1^{(i)}, \widetilde{X}_2^{(i)}\}_{i\in[3]}$ are generated by the tampering function f_{12} that depends on $Y^{(1)}$. Hence, we cannot directly invoke the security of $\mathsf{2NMExt}$ since the tampering against $X^{(1)}$ and $Y^{(1)}$ are not independent of each other. To solve this issue, we rely on a "strong leakage-resilience" property of $\mathsf{2NMExt}$. Under this stronger property, one of the tampering functions can get a leakage about the other source such that the amount of leakage is an arbitrary polynomial in the length of the tampered source. If we have such an extractor, we can view $\{\widetilde{X}_1^{(i)}, \widetilde{X}_2^{(i)}\}_{i\in[3]}$ as leakage from the source $Y^{(1)}$ given to the tampering function f' against $X^{(1)}$. Given this leakage and the input $X^{(3)}$, f' reconstructs the source X_3 from the fixed values and the input $X^{(3)}$ and applies $g_3(X_3)$ to obtain $\widetilde{X}_3$. Now, it can use the leakage $\{\widetilde{X}_1^{(i)}, \widetilde{X}_2^{(i)}\}_{i\in[3]}$ and $\{\widetilde{X}_3^{(i)}\}_{i\in[3]}$ (obtained from $\widetilde{X}_3$) to obtain $\widetilde{X}^{(i)}$ for every $i \in [3]$. Furthermore, f' also has $\widetilde{Y}^{(3)}$. It computes $\mathsf{2NMExt}(\widetilde{X}^{(3)}, \widetilde{Y}^{(3)})$ and samples a string x such that $\mathsf{2NMExt}(x, Y^*) = \mathsf{2NMExt}(\widetilde{X}^{(3)}, \widetilde{Y}^{(3)})$. It outputs $(\widetilde{X}^{(1)}, \widetilde{X}^{(2)}, x)$ as the tampered sources. Notice that applying $\mathsf{2NMExt}$ on the outputs of f', g' precisely yields $(\mathsf{2NMExt}(\widetilde{X}^{(1)}, \widetilde{Y}^{(1)}), \mathsf{2NMExt}(\widetilde{X}^{(2)}, \widetilde{Y}^{(2)}), \mathsf{2NMExt}(\widetilde{X}^{(3)}, \widetilde{Y}^{(3)}))$. Further, it now follows from the split-state non-malleability of $\mathsf{2NMExt}$ that each of these outputs is either independent of $\mathsf{2NMExt}(X^{(1)}, Y^{(1)})$ or is exactly the same as $\mathsf{2NMExt}(X^{(1)}, Y^{(1)})$. This shows the main claim of the proof.

In the next subsection, we show how to construct such a strong leakage-resilient non-malleable extractor.

2.2 Strong Leakage-resilient Non-malleable Extractor

Recall that a $(2,t)$-non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ (introduced in [CG14,CGL16]) satisfies the following property: for any t split-state tampering functions $F = (f_1, g_1) \ldots, (f_t, g_t)$ and independent sources X, Y with sufficient min-entropy, there exists a distribution D_F with support on $\{0,1\}^m \cup \{\mathsf{same}^*\}$ that is independent of X, Y such that

$$|\mathsf{2NMExt}(X,Y), \mathsf{2NMExt}(f_1(X), g_1(Y)), \ldots, \mathsf{2NMExt}(f_t(X), g_t(Y)) - U_m, \mathsf{copy}^{(t)}(D_F, U_m)| < \varepsilon \tag{2.1}$$

where both U_m's refer to the same uniform m-bit string. Here, $\mathsf{copy}^{(t)}((x_1,\ldots,x_t),y) = (z_1,\ldots,z_t)$ where $z_i = \begin{cases} x_i & \text{if } x_i \neq \mathsf{same}^* \\ y & \text{if } x_i = \mathsf{same}^* \end{cases}$.

A leakage-resilient variant of such an extractor requires that even when one half of these tampering functions, say $\{f_i\}_{i\in[t]}$ gets some bounded leakage on the other source Y, the non-malleability property still holds. Specifically, for any leakage function $h : \{0,1\}^n \to \{0,1\}^\mu$, we require that

$$|\mathsf{2NMExt}(X,Y), \{\mathsf{2NMExt}(f_i(X,h(Y)), g_i(Y))\}_{i\in[t]} - U_m, \mathsf{copy}^{(t)}(D_{F,h}, U_m)| < \varepsilon \tag{2.2}$$

It is not hard to see that if the underlying non-malleable extractor tolerates a min-entropy loss of roughly μ, then such a non-malleable extractor can be shown to be leakage-resilient. Notice that for this approach to work, the length of the source must be far greater than the amount of leakage tolerated. However, for our application to constructing multi-source non-malleable extractor, we require the amount of leakage from one of the sources to be an arbitrary polynomial in the length of the other source. Of course, if we insist on both the sources to be of same length then it is easy to see that such a primitive does not exist. Hence, this primitive necessarily requires uneven length sources. We call such a non-malleable extractor as $(2,t)$-strong leakage-resilient non-malleable extractor where we require the output length of h in Eqn 2.2 to be an arbitrary polynomial in the length of X.

A similar primitive for the case of non-malleable codes was studied in the work of Goyal and Kumar [GK18a]. They showed that the CGL construction [CGL16] of non-malleable code satisfies this property. Unfortunately, they neither give a construction of a non-malleable extractor for sufficiently low min-entropy nor do they give a multi-tamperable version of the result. Both of these properties are crucial in obtaining our main results.

In this work, we show that any $(2,t)$-leakage-resilient non-malleable extractor (where the leakage tolerated is only a fraction of the source length) can be bootstrapped to a $(2,t)$-strong leakage-resilient non-malleable extractors (where the leakage tolerated is an arbitrary polynomial in the length of the other source). This gives a modular approach of constructing such primitives and additionally, simplifies the construction of strong leakage resilient non-malleable codes in the work of [GK18a].

2.2.1 Our Compiler

To illustrate the main ideas behind our compiler, let us simplify the problem and assume that X and Y are independent full entropy sources with length n_1 and n_2 respectively. Further, assume that $n_2 >> p(n_1)$ where $p(\cdot)$ is a polynomial denoting the amount of leakage tolerated.

Our compiler under these assumptions is extremely simple. We view the source X as (S, X') where S is the seed of a strong extractor Ext. We apply $\mathrm{Ext}(Y,S)$ to obtain Y' where the length of Y' is equal to the length of X'. We finally apply $\mathsf{2NMExt}(X',Y')$ and output the result. The main intuition behind the compiler is that conditioned on the output of the leakage function, it can be shown (via standard approaches [MW97, DORS08]) that Y has sufficient min-entropy. Hence, if we apply a seeded extractor on this Y, the output is close to uniform.

While the main intuition is relatively straightforward, proving the non-malleability of this construction requires new tricks. Notice that to prove the non-malleability of the compiled construction, we need to invoke the non-malleability of the underlying $\mathsf{2NMExt}$. However, if we closely notice the compiler, we see that the tampered version of the source $\widetilde{Y}'$ that is fed as the second input to

$\mathsf{2NMExt}$ is not only a function of Y but also a function of the other source X' via the tampered seed $\widetilde{S}$. In particular, $\widetilde{S}$ could be a function of the source X' and hence, $\widetilde{Y}'$ is a function of both X' and Y. This means that the tampering of the second source is not independent of the first source and hence, we cannot directly invoke the security of $\mathsf{2NMExt}$. To solve this issue, we recall that $\mathsf{2NMExt}$ is in fact, a leakage-resilient non-malleable extractor. In particular, we can fix the length of the seed S to be small enough so that it is only a fraction of the length of X'. We now view the tampered seed $\widetilde{S}$ as leakage from the source X' to the tampering function of Y. This allows us to reduce the non-malleability of the compiled construction to the leakage-resilient, non-malleability of $\mathsf{2NMExt}$.

Lower min-entropy case. Recall that the above construction crucially relied on the fact that X is a full entropy source to make sure that the seed S has full-entropy. This compiler completely breaks down if X didn't have full entropy as otherwise, we cannot rely on the pseudorandomness of Ext. Thus, we require a new approach to deal with the case where the entropy of the sources are not full. In this setting, we modify our compiler as follows. We view X as (X', X_1) and Y as (Y_1, Y_2). We first apply a strong two-source extractor $\mathsf{2Ext}(X_1, Y_1)$ to get a short seed S. We later apply a strong seeded extractor $\mathrm{Ext}(Y_2, S)$ to obtain Y'. Finally, we output $\mathsf{2NMExt}(X', Y')$.

As in the previous construction, we can show that conditioned on the leakage $h(Y)$, the source Y has sufficient min-entropy. Now, since X_1, Y_1 are independent sources, it follows from the pseudorandomness of $\mathsf{2Ext}$ that the output S is close to uniform. Now, we can rely on the pseudorandomness of Ext to show that Y' is close to uniform. Again, as in the previous case, we can rely on the leakage-resilience property of the underlying $\mathsf{2NMExt}$ extractor to leak the tampered version $\widetilde{X}_1$ to the tampering function of Y and this allows us to argue non-malleability of the compiled construction. However, one subtlety that arises here is that we necessarily require the length of Y_1 to be much larger than the length of the other source X_1 that is fed as input to the strong two-source extractor. This is because we require Y_1 to have sufficient min-entropy even conditioned on the output of the leakage function h and the output of the leakage function is a polynomial in the length of the other source. This means that the length of X_1 is much smaller than the length of Y_1 and hence, we have to rely on the uneven length two-source extractor given by Raz [Raz05].

2.3 Non-Malleable Secret Sharing

A significant advantage of our construction of multi-source non-malleable extractor is its generality to give other primitives. In particular, we show that a minor modification to our construction gives a t-out-of-n non-malleable secret sharing scheme for every t and n against a family of t-cover-free tampering functions. Roughly speaking, t-cover-free family requires that every share is tampered with at most $t-2$ other shares. This family includes disjoint tampering (as defined in [GK18a]) as a special case and gives the first construction of threshold non-malleable secret sharing scheme that is secure against a strict super class of disjoint tampering.[3]

Our Construction. The construction we give for t-out-of-n non-malleable secret closely resembles the construction of our n-source non-malleable extractor. Specifically, the i-th share of our non-malleable secret sharing scheme is viewed as $(X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(n)}, Y^{(i)})$. The only difference

[3]We note that even for the case of disjoint tampering, the work of Goyal and Kumar [GK18a] assumes that the partitioned subsets must be of unequal length.

in the semantics is that instead of viewing $(X_1^{(i)}, \ldots, X_n^{(i)})$ as an XOR (or equivalently, n-out-of-n) secret sharing of the value $X^{(i)}$, we consider them to be a t-out-of-n secret sharing of $X^{(i)}$. Now, given any t-shares, say corresponding to $i_1, \ldots, i_t$, we would be able to reconstruct $X^{(1)}, \ldots, X^{(n)}$ and compute $\mathsf{2NMExt}(X^{(i_j)}, Y^{(i_j)})$ for every $j \in [t]$. We now interpret $\mathsf{2NMExt}(X^{(i_j)}, Y^{(i_j)})$ as the i_j-th Shamir share of a secret message $s \in \{0,1\}^m$ and these t Shamir shares can be put together to reconstruct the secret s. Recall that in the case of multi-source non-malleable extractors, we interpreted $\mathsf{2NMExt}(X^{(i_j)}, Y^{(i_j)})$ as an n-out-of-n secret sharing of the output. Below, we give the description of our sharing algorithm assuming that $\mathsf{2NMExt}$ is efficiently pre-image sampleable. Here, we use a t-out-of-n secret sharing scheme Share with perfect privacy.

To share a secret $s \in \{0,1\}^m$, we do the following:

1. $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n) \leftarrow \mathsf{Share}(s)$.
2. For each $i \in [n]$, compute $(X^{(i)}, Y^{(i)}) \leftarrow \mathsf{2NMExt}^{-1}(\mathsf{Sh}_i)$.
3. For each $i \in [n]$, $(X_1^{(i)}, \ldots, X_n^{(i)}) \leftarrow \mathsf{Share}(X^{(i)})$.
4. Set $\mathsf{share}_i = (X_i^{(1)}, \ldots, X_i^{(n)}, Y^{(i)})$.
5. Output $(\mathsf{share}_1, \ldots, \mathsf{share}_n)$.

We show via a similar argument to the proof of our multi-source non-malleable extractor that if the underlying $\mathsf{2NMExt}$ is strong leakage-resilient then the above non-malleable secret sharing is secure against t-cover-free tampering. The complete analysis of the construction appears in Section 8.

2.4 Network Extractor Protocol

Another application of our multi-source non-malleable extractors is to get improved results for network extractor protocols [DO03, GSV05, KLRZ08, KLR09]. In the setting of network extractors, there are p processors, each with an independent source X_i having some min-entropy. The processors exchange some messages and at the end of the protocol, we require that every honest processor end up with an uniform random string independent of outputs of the other processors and the transcript of the protocol. This property must hold even if a subset of the processors are corrupted by a centralized adversary who can instruct the corrupted processors to deviate arbitrarily from the protocol. It is easy to see that if the adversary controls $p-1$ processors then this task is impossible as it amounts to extracting random bits from a single source with min-entropy less than full. However, if the adversary corrupts at most $p-s$ processors, we show that a s-source non-malleable extractor that is multi-tamperable can give a one-round protocol for this task. Additionally, unlike the other prior works (except in the computational setting), this approach allows every honest party to extract uniform random bits.

For simplicity, let us show a variant of our protocol from a multi-tamperable 2-source non-malleable extractor $\mathsf{2NMExt}$. This allows us to obtain optimal results for the case of $p-2$ corruptions. We give the description of the protocol below.

1. Each processor parses X_i as $X_1^{(i)}, \ldots, X_p^{(i)}$.
2. It broadcast $\{X_j^{(i)}\}_{j \neq i}$.

3. It receive $\{X_i^{(j)}\}_{j \neq i}$ from all the processors. If some processor j does not send any message, it replaces $X_i^{(j)}$ with a default value.

4. For every $j \subseteq [p] \setminus \{i\}$, processor P_i

   (a) Computes $y_j = \mathsf{2NMExt}(X_i^{(i)}, X_i^{(j)})$.

5. It removes the duplicates from the sequence $(y_j)_{j \neq i}$ to get $y'_1, \ldots, y'_k$.

6. It outputs $z_i = y'_1 \oplus \ldots \oplus y'_k$.

The main intuition behind the proof of this network extractor protocol is that for every honest processor i, the message $X_i^{(j)}$ sent by every adversarial processor j can be viewed as a tampering of the message $X_i^{(i^*)}$ of one another honest processor i^*. Thus, it now follows from the multi-tamperability of $\mathsf{2NMExt}$ that the tampered output $\mathsf{2NMExt}(X_i^{(i)}, X_i^{(j)})$ is independent of the real output $\mathsf{2NMExt}(X_i^{(i)}, X_i^{(i^*)})$ which in turn is close to uniform. However, for this argument to hold, we require the non-malleability property to hold even conditioned on $X_i^{(i^*)}$, in other words, we require $\mathsf{2NMExt}$ to be a strong non-malleable extractor. Fortunately, Li [Li17a] showed that every non-malleable extractor with sufficiently low min-entropy is also a strong non-malleable extractor and this allows us to complete the proof.

The new constructions of multi-source extractors for $s \geq 3$ given in this paper have the same min-entropy requirement as that of the two source extractors and hence, do not provide any further improvements over the above result. We leave open the fascinating problem of constructing multi-source extractors for $s \geq 3$ for lower min-entropy requirements.

3 Preliminaries

Notation. We use capital letters to denote distributions and their support, and the corresponding lowercase letters to denote a sample from the same. $x \sim X$ is used to denote a sample x from a distribution X. We will slightly abuse the notation and use X to denote a random variable as well as a distribution. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$, and U_r denote the uniform distribution over $\{0,1\}^r$. For any finite set S, we use $s \leftarrow S$ to denote the process of sampling s uniformly at random from S. For any $i \in [n]$, let x_i denote the symbol at the i-th co-ordinate of x, and for any $T \subseteq [n]$, let $x_T \in \{0,1\}^{|T|}$ denote the projection of x to the co-ordinates indexed by T. We write $\circ$ to denote concatenation.

Standard Definitions and Results. Standard definitions of min-entropy and statistical distance are given below. We also recall some standard results about these notions.

Definition 3.1 (Min-entropy) *The min-entropy of a source X is defined to be*

$$H_\infty(X) = \min_{s \in \text{support}(X)} \{\log(1/\Pr[X = s])\}$$

A (n,k)-source is a distribution on $\{0,1\}^n$ with min-entropy k. The entropy loss is given by $(n-k)$.

Lemma 3.2 ([MW97]) *Let X, Y be random variables such that the random variable Y takes at ℓ values. Then*

$$\Pr_{y\sim Y}\left[H_\infty(X|Y=y) \geq H_\infty(X) - \log \ell - \log\left(\frac{1}{\varepsilon}\right)\right] > 1-\varepsilon.$$

Definition 3.3 ([DORS08]) *The average conditional min-entropy is defined as*

$$\widetilde{H}_\infty(X|W) = \log\left(E_{w\leftarrow W}\left[\max_x \Pr[X=x|W=w]\right]\right) = -\log E\left[2^{-H_\infty(X|W=w)}\right]$$

We recall some results on conditional min-entropy from [DORS08].

Lemma 3.4 ([DORS08]) *If a random variable B can take at most ℓ values, then $\widetilde{H}_\infty(A|B) \geq H_\infty(A) - \log \ell$.*

Definition 3.5 (Statistical distance) *Let D_1 and D_2 be two distributions on a set S. The statistical distance between D_1 and D_2 is defined to be:*

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

D_1 is ε-close to D_2 if $|D_1 - D_2| \leq \varepsilon$.

We will use the notation $D_1 \approx_\varepsilon D_2$ to denote that the statistical distance between D_1 and D_2 is at most ε.

Lemma 3.6 (Triangle Inequality) *If $D_1 \approx_{\varepsilon_1} D_2$ and $D_2 \approx_{\varepsilon_2} D_3$ then $D_1 \approx_{\varepsilon_1+\varepsilon_2} D_3$.*

Lemma 3.7 ([CG14]) *Let D_1 and D_2 be two distribution on a finite set S and suppose they are ε-close to each other. Let E be any event such that $\Pr(E) = p$. Then, the condition distributions $D_1|E$ and $D_2|E$ are (ε/p)-close.*

Seeded Extractors. We now recall the definition of a strong seeded extractor.

Definition 3.8 (Strong seeded extractor) *A function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a strong seeded extractor for min-entropy k and error ε if for any (n,k)-source X and an independent uniformly random string U_d, we have*

$$|\mathrm{Ext}(X, U_d) \circ U_d - U_m \circ U_d| < \varepsilon,$$

where U_m is independent of U_d. Further if the function $\mathrm{Ext}(\cdot, u)$ is a linear function over $\mathbb{F}_2$ for every $u \in \{0,1\}^d$, then Ext is called a linear seeded extractor.

Theorem 3.9 ([GUV09]) *For every constant $\alpha > 0$, and any $n, k, \varepsilon > 0$, there exists a strong seeded extractor $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ for min-entropy k and error ε with $d = O(\log n + \log 1/\varepsilon)$ and $m = (1-\alpha)k$.*

We will also use the explicit constructions of strong linear seeded extractors [Tre01] [RRV02].

Theorem 3.10 ([Tre01] [RRV02]) *For every $n, k, m \in \mathbb{N}$ and $\varepsilon > 0$ such that $m \leq k \leq n$, there exists an explicit linear strong seeded extractor* $\mathrm{LSExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy k, error ε, and $d = O\left(\frac{\log^2(n/\varepsilon)}{\log(k/m)}\right)$.*

An average case seeded extractor requires that if a source X has average case conditional min-entropy $\widetilde{H}_\infty(X|Z) \geq k$ then the output of the extractor is uniform even when Z is given. We recall the following lemma from [DORS08] which states that every strong seeded extractor is also an average-case strong extractor.

Lemma 3.11 ([DORS08]) *For any $\delta > 0$, if* Ext *is a (k, ε)-strong seeded extractor then it is also a $(k + \log\left(\frac{1}{\delta}\right), \varepsilon + \delta)$ average case strong extractor.*

3.1 Strong Seedless Extractors

We now recall the definition of 2-source extractors and strong 2-source extractors.

Definition 3.12 (2-source extractor) *A function* $\mathsf{2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *is called a two sources extractor for min-entropy (k_1, k_2) and error ε if for any independent (n_1, k_1)-source X and (n_2, k_2)-source Y, we have*

$$|\mathsf{NMExt}(X, Y) - U_m| \leq \varepsilon.$$

where U_m is independent of X, Y.

Theorem 3.13 ([CG85]) *For all integers $n, m, k_1, k_2 > 0$, there exists an efficient 2-source extractor* $\mathrm{IP} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *with min-entropy (k_1, k_2) and error $\varepsilon = 2^{-(k_1+k_2-n-m)/2}$.*

Definition 3.14 (Strong extractor) *A function* $\mathsf{2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *is called a strong two sources extractor for min-entropy (k_1, k_2) and error ε if for any independent (n_1, k_1)-source X and (n_2, k_2)-source Y, we have*

$$|X \circ \mathsf{2Ext}(X, Y) - X \circ U_m| < \varepsilon \text{ and } |Y \circ \mathsf{2Ext}(X, Y) - Y \circ U_m| < \varepsilon,$$

where U_m is independent of X, Y.

Theorem 3.15 ([Raz05]) *For all integers n_1, n_2, k_1, m and for any $\varepsilon > 0$ such that $n_2 = \Omega(\log(n_1/\varepsilon))$, $k_1 = \Omega(n_2)$ and $m = O(n_2)$, there exists an efficient-computable strong 2-source extractor* $\mathrm{Raz} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *with min-entropy $(k_1, 0.6n_2)$ and error ε.*

3.2 Seedless Non-Malleable Extractors

We now give the definition of 2-source, non-malleable extractors that are tamperable t times [CGL16]. Such an extractor is called as $(2, t)$-non-malleable extractors.

Definition 3.16 ((2,t)-Non-Malleable Extractor) *A function* $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a $(2, t)$-non-malleable extractor at min-entropy k and error ε if it satisfies the following property: if X and Y are independent (n, k)-sources and $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$*

are t arbitrary 2-split-state tampering functions, then there exists a random variable $D_{\overrightarrow{f},\overrightarrow{g}}$ on $(\{0,1\}^m \cup \{\mathsf{same}^*\})^t$ which is independent of the random variables X and Y, such that

$$|\mathsf{2NMExt}(X,Y), \mathsf{2NMExt}(f_1(X),g_1(Y)), \ldots, \mathsf{2NMExt}(f_t(X),g_t(Y)) - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g}}, U_m)| < \varepsilon$$

where both U_m's refer to the same uniform m-bit string. Here, $\mathsf{copy}^{(t)}((x_1,\ldots,x_t),y) = (z_1,\ldots,z_t)$ where $z_i = \begin{cases} x_i & \text{if } x_i \neq \mathsf{same}^* \\ y & \text{if } x_i = \mathsf{same}^* \end{cases}$.

For $t = 1$, we call $\mathsf{2NMExt}$ a non-malleable 2-source extractor.

Theorem 3.17 ([CGL16]) *There exists a constant $\gamma > 0$ such that for all $n > 0$ and $t < n^\gamma$, there exists a $(2,t)$-non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{n^{\Omega(1)}}$ at min-entropy $n - n^\gamma$ with error 2^{-n^γ}.*

Theorem 3.18 ([Li17b]) *For any $n > 0$, there exists a constant γ such that there exists a non-malleable 2-source extractor $\mathsf{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ with min-entropy $(1-\gamma)n$, $m = \Omega(k)$ and error $\varepsilon = 2^{-\Omega(n/\log(n))}$.*

3.3 Non-Malleable Codes

We start with the definition of a coding scheme.

Definition 3.19 (Coding scheme) *Let $\mathsf{Enc} : \{0,1\}^m \to \{0,1\}^n$ be a randomized algorithm and $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^m \cup \{\perp\}$ be a deterministic function. We say that $(\mathsf{Enc}, \mathsf{Dec})$ is a* coding scheme with code length n and message length m *if for all $s \in \{0,1\}^m$, $\Pr[\mathsf{Dec}(\mathsf{Enc}(s)) = s] = 1$, where the probability is taken over the randomness of Enc. The* rate *of the coding scheme is $\frac{m}{n}$.*

Dziembowski, Pietrzak and Wichs [DPW18] introduced the notion of non-malleable codes which generalizes the usual notion of error correction. In particular, it guarantees that when a codeword is subject to tampering attack, the reconstructed message is either the original one or something that is independent of the original message.

Definition 3.20 (Non-Malleable Codes [DPW18]) *Let $\mathsf{Enc} : \{0,1\}^m \to \{0,1\}^n$ and $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^m \cup \{\perp\}$ be (possibly randomized) functions, such that $\mathsf{Dec}\big(\mathsf{Enc}(s)\big) = s$ with probability 1 for all $s \in \{0,1\}^m$. Let $\mathcal{F}$ be a family of tampering functions and fix $\varepsilon > 0$. We say that $(\mathsf{Enc}, \mathsf{Dec})$ is* $\varepsilon-$non-malleable *w.r.t. $\mathcal{F}$ if for every $f \in \mathcal{F}$, there exists a random variable D_f on $\{0,1\}^m \cup \{\mathsf{same}\}$, such that for all $s \in \{0,1\}^m$,*

$$|\mathsf{Dec}(f(X_s)) - \mathsf{copy}(D_f, s)| \leq \varepsilon$$

where $X_s \leftarrow \mathsf{Enc}(s)$ and copy is defined by $\mathsf{copy}(x,y) = \begin{cases} x & \text{if } x \neq \mathsf{same} \\ y & \text{if } x = \mathsf{same} \end{cases}$. We call n the length *of the code and m/n the* rate.

Chattopadhyay, Goyal and Li [CGL16] defined a stronger notion of non-malleability against multiple tampering and we now recall this definition.

Definition 3.21 (Non-Malleable Codes against Multiple Tampering [CGL16]) *A coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with code length* n *and message length* m *is a non-malleable code with tampering degree* t *w.r.t. a family of tampering functions* $\mathcal{F} \subset (\mathcal{F}_n)^t$ *and error* ε *if for every* $(f_1, \ldots, f_t) \in \mathcal{F}$*, there exists a random variable* $D_{\vec{f}}$ *on* $(\{0,1\}^m \cup \{\mathsf{same}\})^t$ *such that for all messages* $s \in \{0,1\}^m$*, it holds that*

$$|(\mathsf{Dec}(f_1(X)), \ldots, \mathsf{Dec}(f_t(X))) - \mathsf{copy}^{(t)}(D_{\vec{f}}, s)| \leq \varepsilon$$

where $X = \mathsf{Enc}(s)$*. We refer to* t *as the* tampering degree *of the code.*

4 $(2,t)$-Non-Malleable Randomness Extractors

In this section, we give a construction of $(2,t)$-Non-malleable extractors for min-entropy $\Omega(n)$. We achieve this by giving a generic transformation from $(2,1)$-non-malleable extractor to $(2,t)$-non-malleable randomness extractor. This follows a similar approach given in [Coh16b] for the case of seeded non-malleable extractors. We start with a slightly stronger definition of non-malleable extractors given in [CGL16] that is shown to imply the standard definition.

Definition 4.1 (t-non-malleable 2-source extractor) *For an integer* $t \geq 1$*, a* t*-non-malleable 2-source extractor for min-entropy* k *and error* ε *is a function* $\mathsf{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *with the following property. Let* $f_1, g_1, \ldots, f_n, g_n$ *be arbitrary function from* $\{0,1\}^n$ *to* $\{0,1\}^n$ *such that at least one of* f_i, g_i *has no fixed point for all* $i \in [t]$*. Let* X, Y *be independent* (n,k)*-sources. Let* $X^{(i)} = f_i(X)$ *and* $Y^{(i)} = g_i(Y)$ *for* $i \in [t]$*. Then, it holds that*

$$|\mathsf{NMExt}(X,Y), \{\mathsf{NMExt}(X^{(i)}, Y^{(i)})\}_{i\in[t]} - U_m, \{\mathsf{NMExt}(X^{(i)}, Y^{(i)})\}_{i\in[t]}| \leq \varepsilon.$$

For $t = 1$*, we call* NMExt *a non-malleable 2-source extractor.*

One of the main tools used in this transformation is a correlation breaker with advice and we start by recalling this definition.

Definition 4.2 (t-correlation-breaker with advice [Coh16a]) *For an integer* $t \geq 1$ *a* t*-correlation-breaker with advice for min-entropy* k *and error* ε *is a function* $\mathsf{AdvBC} : \{0,1\}^w \times \{0,1\}^l \times \{0,1\}^a \to \{0,1\}^m$ *with the following property. Let* $X, X^{(1)}, \ldots, X^{(t)}$ *be random variables distributed over* $\{0,1\}^w$ *such that* X *has min-entropy* k*. Let* $Y, Y^{(1)}, \ldots, Y^{(t)}$ *be random variables distributed over* $\{0,1\}^l$ *that are jointly independent of* $(X, X^{(1)}, \ldots, X^{(t)})$ *such that* Y *is uniform. Then, for any string* $s, s^{(1)}, \ldots, s^{(t)} \in \{0,1\}^a$ *such that* $s \notin \{s^{(1)}, \ldots, s^{(t)}\}$*, it holds that*

$$|\mathsf{AdvBC}(X,Y,s), \{\mathsf{AdvBC}(X^{(i)}, Y^{(i)}, s^{(i)})\}_{i\in[t]} - U_m, \{\mathsf{AdvBC}(X^{(i)}, Y^{(i)}, s^{(i)})\}_{i\in[t]}| \leq \varepsilon.$$

Theorem 4.3 ([CGL16]) *For all integers* ℓ, w, a, t *and for any* $\varepsilon \in (0,1)$ *such that*

$$\ell = \Omega(at \cdot \log(aw/\varepsilon)),$$

there exists a poly(ℓ, w)*-time computable* t*-correlation-breaker with advice* $\mathsf{AdvBC} : \{0,1\}^w \times \{0,1\}^\ell \times \{0,1\}^a \to \{0,1\}^m$*, for entropy*

$$k = \Omega(at \cdot \log(a\ell/\varepsilon)),$$

with error ε *and* $m = \Omega(\ell/(at))$ *output bits.*

4.1 Transformation

Building blocks and parameters

1. Let $\mathsf{NMExt} : \{0,1\}^{d_1} \times \{0,1\}^{d_1} \to \{0,1\}^{l_1}$ be a non-malleable 2-source extractor with min-entropy $d_1 - \Delta$ and error ε, where $l_1 = \Omega(\log(1/\varepsilon))$.
2. Let $\mathsf{ECC} : \{0,1\}^{d_2} \to \{0,1\}^{D_2}$ be an error correcting code with $D_2 = O(d_2)$ and relative distance $1/4$.
3. Let $\mathsf{IP} : \{0,1\}^{d_1} \times \{0,1\}^{d_1} \to \{0,1\}^{l_2'}$ be a strong 2-source extractor with error ε and min-entropy $d_1 - \Delta$, where $l_2' = l_2 \log(D_2)$ and $l_2 = \Omega(\log(1/\varepsilon))$.
4. Let $\mathsf{Raz} : \{0,1\}^{n} \times \{0,1\}^{d_2} \to \{0,1\}^{l_3}$ be a strong 2-source extractor with error ε, where the min-entropy requirement for the first source is $n - \Delta - (1+t)(d_1 + l_2) - \log(1/\varepsilon)$ and that for the second source is $d_2 - \Delta - (1+t)(d_1 + l_2) - \log(1/\varepsilon)$.
5. Let $\mathsf{AdvBC} : \{0,1\}^{d_3} \times \{0,1\}^{l_3} \times \{0,1\}^{a} \to \{0,1\}^{m}$ be an efficient t-correlation-breaker with advice for error ε and min-entropy $d_3 - \Delta - (1+t)(d_1 + l_2 + d_2) - \log(1/\varepsilon)$, where $a = l_1 + 2l_2$

Construction On the input sources X, Y, NMExt' is computed as follows.

1. Let $X = X_1 \circ X_2$, $Y = Y_1 \circ Y_2$, where $|X_1| = |Y_1| = d_1$.
2. Let $\mathsf{AdvGen}(X, Y) = \mathsf{NMExt}(X_1, Y_1) \circ \mathsf{ECC}(X_2)_{\mathsf{IP}(X_1,Y_1)} \circ \mathsf{ECC}(Y_2)_{\mathsf{IP}(X_1,Y_1)}$, where $S_{\mathsf{IP}(X_1,Y_1)}$ means to take the bits from S with indexes represented by $\mathsf{IP}(X_1, Y_1)$.
3. Let $Y_2 = Y_3 \circ Y_4$, where $|Y_3| = d_2$ and $|Y_4| = d_3$.
4. Return $\mathsf{AdvBC}(Y_4, \mathsf{Raz}(X, Y_3), \mathsf{AdvGen}(X, Y))$.

Theorem 4.4 *In the above construction,* NMExt' *is a* t*-non-malleable 2-source extractor with min-entropy* $n - \Delta$ *and error* $O(t\sqrt{\varepsilon})$.

Proof Denote the tampering function as $f_1, g_1, \ldots, f_t, g_t$ such that at least one of f_i, g_i has no fixed point for all $i \in [t]$. Let $X^{(i)} = f_i(X)$ and $Y^{(i)} = g_i(Y)$. Let $Z = \mathsf{AdvGen}(X, Y)$ and $Z^{(i)} = \mathsf{AdvGen}(X^{(i)}, Y^{(i)})$.

Claim 4.5 *With high probability,* Z *is not equal to all* $Z^{(1)}, \ldots, Z^{(t)}$.

Proof Denote a tampered version of X, Y as X', Y'. Assume without loss of generality that $X \neq X'$ always holds. Let $\tilde{X}_1 = X_1'$ if $X_1 \neq X_1'$ and $\tilde{X}_1$ be an arbitrary value not equal to X_1 if $X_1 = X_1'$. Also, since X_1 and Y_1 are independent $(d_1, d_1 - \Delta)$ source, it holds that

$$|\mathsf{NMExt}(X_1, Y_1), \mathsf{NMExt}(\tilde{X}_1, Y_1') - U_{l_1}, \mathsf{NMExt}(\tilde{X}_1, Y_1')| \leq \varepsilon,$$

$$\Pr[\mathsf{NMExt}(X_1, Y_1) = \mathsf{NMExt}(\tilde{X}_1, Y_1')] \leq \Pr[U_{l_1} = \mathsf{NMExt}(\tilde{X}_1, Y_1')] + \varepsilon \leq 2^{-l_1} + \varepsilon \leq 2\varepsilon.$$

Since X_1 and Y_1 are independent $(d_1, d_1 - \Delta)$ source, it holds that

$$|\mathsf{IP}(X_1, Y_1), X_1 - U_{l_2}, X_1| \leq \varepsilon.$$

Denote x_1 is good if $|\mathsf{IP}(x_1, Y_1), x_1 - U_{l_2'}, x_1| \leq \sqrt{\varepsilon}$ and we have

$$\Pr[X_1 \in \mathsf{good}] \geq 1 - \sqrt{\varepsilon}.$$

Now consider a fixed $x = (x_1, x_2)$ and x_2' such that x_1 is good and $x_2 \neq x_2'$. It holds that

$$\Pr[\mathsf{ECC}(x_2)_{\mathsf{IP}(x_1,Y_1)} = \mathsf{ECC}(x_2')_{\mathsf{IP}(x_1,Y_1)}] \leq \Pr[\mathsf{ECC}(x_2)_{U_{l_2'}} = \mathsf{ECC}(x_2')_{U_{l_2'}}] + \sqrt{\varepsilon} \leq (3/4)^{l_2} + \sqrt{\varepsilon} = \varepsilon + \sqrt{\varepsilon}.$$

Thus, it holds that

$$\begin{aligned}
&\Pr[\mathsf{AdvGen}(X, Y) = \mathsf{AdvGen}(X', Y')] \\
&\leq \sum_{(x_1,x_2)\in\{0,1\}^n} \Pr[X = (x_1, x_2) \wedge X_1' = x_1] \Pr[Y_1 = Y_1' \wedge \mathsf{ECC}(x_2)_{\mathsf{IP}(x_1,Y_1)} = \mathsf{ECC}(x_2')_{\mathsf{IP}(x_1,Y_1)}] \\
&\qquad + \Pr[X = (x_1, x_2) \wedge (X_1', Y_1') \neq (x_1, Y_1) \wedge \mathsf{NMExt}(X_1, Y_1) = \mathsf{NMExt}(X_1', Y_1')] \\
&\leq \Pr[\mathsf{NMExt}(X_1, Y_1) = \mathsf{NMExt}(\tilde{X}_1, Y_1')] \\
&\quad + \sum_{(x_1,x_2)\in\{0,1\}^n} \Pr[X = (x_1, x_2) \wedge X_1' = x_1 \wedge x_1 \in \mathsf{good}](\varepsilon + \sqrt{\varepsilon}) \\
&\quad + \Pr[X_1 \notin \mathsf{good}] \\
&\leq 3\varepsilon + 2\sqrt{\varepsilon}.
\end{aligned}$$

By union bound, the probability that one of $Z^{(1)}, \ldots, Z^{(t)}$ is equal to Z is $O(t\sqrt{\varepsilon})$. ■

Now we fix $Z, X_1, Y_1, Z^{(1)}, X_1^{(1)}, Y_1^{(1)}, \ldots, Z^{(t)}, X_1^{(t)}, Y_1^{(t)}$ and then $X, (Y_3, Y_4)$ are independent. By Theorem 3.2, with probability $1-\varepsilon$, X has min-entropy at least $n-\Delta-(1+t)(d_1+l_2)-\log(1/\varepsilon)$. Similarly, with probability $1-\varepsilon$, Y_3 has min-entropy at least $d_2 - \Delta - (1+t)(d_1+l_2) - \log(1/\varepsilon)$. Assume X has min-entropy $n-\Delta-(1+t)(d_1+l_2)-\log(1/\varepsilon)$ and Y_3 has min-entropy $d_2-\Delta-(1+t)(d_1+l_2)-\log(1/\varepsilon)$. Let $S = \mathsf{Raz}(X, Y_3)$. Since Raz is a strong 2-source non-malleable extractor and (X, Y_3) satisfies the min-entropy requirement, we have

$$|S, Y_3 - U_m, Y_3| \leq \varepsilon.$$

Since S and Y is independent given Y_3, we have

$$|S, Y - U_m, Y| \leq \varepsilon,$$

$$|S, Y_3, Y_3^{(1)}, \ldots, Y_3^{(t)} - U_m, Y_3, Y_3^{(1)}, \ldots, Y_3^{(t)}| \leq \varepsilon.$$

Now fix $Y_3, Y_3^{(1)}, \ldots, Y_3^{(t)}$. With probability at least $1-\sqrt{\varepsilon}$, $|S - U_{l_3}| \leq \sqrt{\varepsilon}$ and with probability at least $1-\varepsilon$, Y_4 has min-entropy $d_3 - \Delta - (1+t)(d_1+l_2+d_2) - \log(1/\varepsilon)$. Also, $(Y_4, Y_4^{(1)}, \ldots, Y_4^{(t)})$ is independent of $(S, S^{(1)}, \ldots, S^{(t)})$.

To summarize, denote the value we so far by $\tau = (Z, Z^{(1)}, \ldots, Z^{(t)}, X_1, X_1^{(1)}, \ldots, X_1^{(t)}, Y_1, Y_1^{(1)}, \ldots, Y_1^{(t)}, Y_3, Y_3^{(1)}, \ldots, Y_3^{(t)})$. Define τ is good if given τ the following holds.

1. Z is not equal to $Z^{(i)}$ for all i.
2. $|S - U_{l_3}| \leq \sqrt{\varepsilon}$.

3. Y_4 has min-entropy $d_3 - \Delta - (1+t)(d_1 + l_2 + d_2) - \log(1/\varepsilon)$.

4. $(S, S^{(1)}, \ldots, S^{(t)})$ is independent of $(Y_4, Y_4^{(1)}, \ldots, Y_4^{(t)})$.

From above, the probability that τ is not good is $O(t\sqrt{\varepsilon})$. For a good τ, consider the following hybrids.
$\underline{D_0}$: Given τ, sample X and Y. Output $\mathsf{AdvBC}(Y_4, S, Z), \mathsf{AdvBC}(Y_4^{(1)}, S^{(1)}, Z^{(1)}), \ldots, \mathsf{AdvBC}(Y_4^{(t)}, S^{(t)}, Z^{(t)})$.
$\underline{D_1}$: Given τ, sample Y and sample $\overline{X}$ as the source X. Compute S with τ and $\overline{X}$ and then sample X given τ and S. Output $\mathsf{AdvBC}(Y_4, S, Z), \mathsf{AdvBC}(Y_4^{(1)}, S^{(1)}, Z^{(1)}), \ldots, \mathsf{AdvBC}(Y_4^{(t)}, S^{(t)}, Z^{(t)})$.
$\underline{D_2}$: Given τ, sample Y and sample S uniformly from $\{0,1\}^{l_3}$. Sample X given τ and S. Output

$$\mathsf{AdvBC}(Y_4, S, Z), \mathsf{AdvBC}(Y_4^{(1)}, S^{(1)}, Z^{(1)}), \ldots, \mathsf{AdvBC}(Y_4^{(t)}, S^{(t)}, Z^{(t)}).$$

$\underline{D_3}$: Given τ, sample Y and sample S uniformly from $\{0,1\}^{l_3}$. Sample X given τ and S. Output

$$U_m, \mathsf{AdvBC}(Y_4^{(1)}, S^{(1)}, Z^{(1)}), \ldots, \mathsf{AdvBC}(Y_4^{(t)}, S^{(t)}, Z^{(t)}).$$

$\underline{D_4}$: Given τ, sample X and Y. Output $U_m, \mathsf{AdvBC}(Y_4^{(1)}, S^{(1)}, Z^{(1)}), \ldots, \mathsf{AdvBC}(Y_4^{(t)}, S^{(t)}, Z^{(t)})$.

The distribution D_0 should be the same as D_1. The distribution D_1 is $\sqrt{\varepsilon}$ close to D_2 since S in D_1 is $\sqrt{\varepsilon}$ close to U_{l_3}. For distribution D_2 and D_3, since τ is good, the followings hold.

1. Z is not equal to $Z^{(i)}$ for all i.

2. S is uniform.

3. Y_4 has min-entropy $d_3 - \Delta - (1+t)(d_1 + l_2 + d_2) - \log(1/\varepsilon)$.

4. $(S, S^{(1)}, \ldots, S^{(t)})$ is independent of $(Y_4, Y_4^{(1)}, \ldots, Y_4^t)$.

Therefore, by the security of AdvBC, the distribution D_2 is ε close to D_3. Similar to the argument for the transformation from D_0 to D_2, it holds that $|D_3 - D_4| \leq \sqrt{\varepsilon}$. Thus, we have

$$|\mathsf{NMExt}(X,Y), \mathsf{NMExt}(X^{(1)}, Y^{(1)}), \ldots, \mathsf{NMExt}(X^{(t)}, Y^{(t)}) - U_m, \mathsf{NMExt}(X^{(1)}, Y^{(1)}), \ldots, \mathsf{NMExt}(X^{(t)}, Y^{(t)})|$$
$$\leq \Pr[\tau \notin \mathsf{good}] + \Pr[\tau \in \mathsf{good}] \cdot |D_0 - D_4| \leq O(t\sqrt{\varepsilon}).$$

■

Instantiation

1. Let $d_1 = C_1 n/t^2, d_2 = C_2 n/t, l_2 = C_3 n^\beta/t^2, l_1 = \log(1/\varepsilon) = C_4 n^\beta/t^2, \Delta = C_5 n/t^2$ and $l_3 = C_6 n/t$, where $0 \leq C_1, C_2, C_3, C_4, C_5, C_6, \beta < 1$.

2. Also, $a = l_1 + 2l_2 \leq (C_4 + 2C_3) n^\beta/t^2$.

3. Instantiate NMExt from Theorem 3.18 with min-entropy $(1-\gamma)d_1$ and error $2^{-\Omega(d_1/\log(d_1))}$.

4. Instantiate IP from Theorem 3.13 with error $2^{-\frac{d_1 - 2\Delta - l_2' - 1}{2}}$.

5. Instantiate Raz from Theorem 3.15 with $d_2 = \Omega(\log(n/\varepsilon))$, the min-entropy requirement for the first source is $k_R = \Omega(d_2)$, the min-entropy requirement for the second source is $0.6d_2$ and $l_3 = O(d_2)$.

6. Instantiate AdvBC from Theorem 4.3 with $l_3 = \Omega\left(at\log(ad_3/\varepsilon)\right)$, min-entropy $\Omega(at\log(al_3/\varepsilon))$ and $m = \Omega(l_3/(at))$.

7. Set $C_1 \geq 1/\gamma C_5$ and for n such that $n^{1-\beta}C_1 \geq C_4\log(C_1 n/t^2)$, the error and min-entropy requirements for NMExt are satisfied.

8. Set $C_1 \geq 2C_5 + 3C_4$ so that the error requirement for IP is satisfied.

9. Set $0.4C_2 \geq C_5 + 2C_1 + 2C_3 + C_4$ so that the min-entropy requirement for the second source of Raz is satisfied.

10. Set $1 \geq C_5 + 2C_1 + 2C_3 + C_4$ so that the min-entropy requirements for the first source of Raz are satisfied.

11. Set $C_6 \geq 2C_3 + C_4$ and $1 \geq 2C_1 + 2C_2 + 3C_3 + 4C_4 + C_5$ so that the requirements for AdvBC are satisfied.

For any constant $t \geq 1$, the error of NMExt' is $O(t\sqrt{\varepsilon}) = 2^{-\Omega(n^{\beta/2})} = 2^{-n^{\Omega(1)}}$, the output length m is $\Omega(l_3/(at)) = \Omega(n^{1-\beta}) = n^{\Omega(1)}$ and $\Delta = C_5 n/t^2 = O(n)$. Thus, we have the following corollary.

Corollary 4.6 *For any $t \geq 1$, there exists constant $n_0, \gamma > 0$ such that for any $n > n_0$ there exists a t-non-malleable 2-source extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ satisfying definition 4.1 with error $2^{-n^{\Omega(1)}}$, min-entropy $(1-\gamma)n$ and output length $m = n^{\Omega(1)}$.*

We now show that $(2,t)$-non-malleable extractors satisfying Definition 4.1 implies standard definition (see Definition 3.16) using the ideas developed in [CGL16]. Large parts of the following proof is taken verbatim from [CGL16].

Lemma 4.7 *Let $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be a $(2,t)$ non-malleable extractor for (n,k) sources with error ε satisfying definition 4.1. Then, for any $k' \geq k$, $\mathsf{2NMExt}$ is $(2,t)$ non-malleable extractor for (n,k') sources with error $2^{2t}(\varepsilon + 2^{k-k'})$*

Proof Let $\mathsf{adv}_1 = (f_1, g_1), \ldots, \mathsf{adv}_t = (f_t, g_t)$ be arbitrary 2-split-state tampering functions. We partition $\{0,1\}^n$ in two different ways based on the fixed points of the tampering functions.

For any $R \subseteq [t]$, define

$$W^{(R)} = \{x \in \{0,1\}^n : f_i(x) = x \text{ for } i \in R, \text{ and } f_i(x) \neq x \text{ for } i \in [t] \setminus R\}.$$

Similarly, for any $S \subseteq [t]$, define

$$V^{(S)} = \{x \in \{0,1\}^n : g_i(x) = x \text{ for } i \in S, \text{ and } g_i(x) \neq x \text{ for } i \in [t] \setminus S\}.$$

For any $R \subseteq [t]$, let $X^{(R)}$ be the conditional distribution of the random variable X conditioned on $X \in W^{(R)}$. Similarly, define $Y^{(S)}$ to be the conditional distribution of the random variable Y conditioned on $Y \in V^{(S)}$. Define

$$D^{(R,S)}_{\mathsf{adv}} = (U_m, \mathsf{copy}^{(t)}(Z_1^{(R,S)}, \ldots, Z_t^{(R,S)}, U_m))$$

where $Z^{(R,S)_i} = \mathsf{2NMExt}(X^{(R)}, Y^{(S)})$ if $i \in [t] \setminus (R \cap S)$; else, it is equal to same^*. We define the distribution

$$D_{\mathsf{adv}} = \sum_{R,S} \alpha_{(R,S)} D^{(R,S)}$$

where $\alpha_{(R,S)} = \Pr[X \in W^{(R)}] \Pr[Y \in V^{(S)}]$.

Claim 4.8 *Let*

$$\Delta_{(R,S)} = \alpha_{(R,S)} |\mathsf{2NMExt}(X^{(R)}, Y^{(S)}), \mathsf{2NMExt}(\mathsf{adv}_1(X^{(R)}, Y^{(S)})), \ldots, \mathsf{2NMExt}(\mathsf{adv}_t(X^{(R)}, Y^{(S)})) - D^{(R,S)}_{\mathsf{adv}}|$$

For any $R, S \subseteq [t]$, $\Delta_{(R,S)} \leq 2^{k-k'} + \varepsilon$.

Proof If either $\Pr[X \in W^{(R)}] \leq 2^{k-k'}$ or $\Pr[Y \in V^{(S)}] \leq 2^{k-k'}$ then $\Delta_{(R,S)} \leq 2^{k-k'}$ and we are done. So, let us assume that $\Pr[X \in W^{(R)}] \geq 2^{k-k'}$ and $\Pr[Y \in V^{(S)}] \geq 2^{k-k'}$. This implies that $H_\infty(X^{(R)})$ and $H_\infty(Y^{(S)})$ is at least k.

Now, let $T = [t] \setminus (R \cap S)$. Then for every element $i \in T$, either f_i has no fixed points or g_i has no fixed points. Then, from definition 4.1, we infer that:

$$|\mathsf{2NMExt}(X^{(R)}, Y^{(S)}), \mathsf{2NMExt}(\mathsf{adv}_T(X^{(R)}, Y^{(S)})) - U_m, \mathsf{2NMExt}(\mathsf{adv}_T(X^{(R)}, Y^{(S)}))| \leq \varepsilon$$

Now, the claim follows from the observation that for every $i \in R \cap S$, both f_i and g_i are identity functions on $W^{(R)}$ and $V^{(S)}$ respectively. ■

Now, if X and Y are sources with min-entropy at least k', we have

$$\begin{aligned} &|\mathsf{2NMExt}(X, Y), \mathsf{2NMExt}(\mathsf{adv}_1(X, Y)), \ldots, \mathsf{2NMExt}(\mathsf{adv}_t(X, Y)) \\ &- U_m, \mathsf{copy}^{(t)}(D_{\mathsf{adv}}, U_m)| \leq \sum_{R,S} \Delta_{(R,S)} \leq 2^{2t}(\varepsilon + 2^{k-k'}). \end{aligned}$$

■

Instantiating with the extractor from Corollary 4.6 and setting $n'_0 = \max(n_0, t^{O(1)})$ and $k' = (1-\gamma')n$ for some $\gamma' < \gamma$, we get the following corollary.

Corollary 4.9 *For any $t \geq 1$, there exists constant $n'_0, \gamma' > 0$ such that for any $n > n'_0$ there exists a t-non-malleable 2-source extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ satisfying definition 3.16 with error $2^{-n^{\Omega(1)}}$, min-entropy $(1-\gamma')n$ and output length $m = n^{\Omega(1)}$.*

5 Strong Leakage-Resilient Non-Malleable Extractor

In this section, we give a construction of a $(2,t)$-non-malleable extractor where one of the tampering functions, say g, that is tampering the source Y, can get leakage about the other source X. The crucial property we will need is that the amount of leakage can be an arbitrary polynomial in the length of the source Y. We call such non-malleable extractors as *strong leakage-resilient non-malleable extractors*. This, in particular would require that the length of the source X to be much larger than the length of the other source Y.

Definition. We now define a strong leakage-resilient non-malleable extractor.

Definition 5.1 (Strong Leakage-Resilient Non-Malleable Extractor) *For any polynomial $p(\cdot)$, a $(2,t)$ non-malleable extractor $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ is said to be p-strong leakage resilient if it satisfies the following property: if X and Y are independent (n_1,k_1) and (n_2,k_2) sources, $\mathcal{A}_1 = (f_1,g_1),\ldots,\mathcal{A}_t = (f_t,g_t)$ are t arbitrary 2-split-state tampering functions and $h : \{0,1\}^{n_1} \to \{0,1\}^{p(n_2)}$ is an arbitrary leakage function, then there exists a random variable $D_{\overrightarrow{f},\overrightarrow{g},h}$ on $(\{0,1\}^m \cup \{\mathsf{same}^*\})^t$ which is independent of the random variables X and Y, such that*

$$|\mathsf{2SLNMExt}(X,Y), \mathsf{2SLNMExt}(f_1(X), g_1(h(X),Y)), \ldots, \mathsf{2SLNMExt}(f_t(X), g_t(h(X),Y)) - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, U_m)| < \varepsilon$$

where both U_m's refer to the same uniform m-bit string.

Organization. This section is organized as follows. In Section 5.1, we define a weaker variant called as leakage resilient non-malleable extractor. The main difference between this variant and our strong leakage-resilience is that here, the sources are of same length but one of the tampering functions can get some fractional leakage about the other source. We show that any non-malleable extractors that works for sufficiently small min-entropy already satisfies this property. Next, in Section 5.2, we show how to bootstrap leakage-resilience to strong leakage-resilience with the help of a strong seeded extractor and strong two-source extractors. In Section 5.4, we give a variant of our extractor that is additionally preimage sampleable. Finally, in section 5.5, we give a couple of useful lemmas that will be used in the subsequent sections.

5.1 Leakage-Resilient Non-Malleable Extractors

We now give the definition of a $(2,t)$-leakage resilient non-malleable extractor.

Definition 5.2 (Leakage-Resilient Non-Malleable Extractor) *For some $\mu \in \mathbb{N}$, a $(2,t)$ non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is said to be μ-leakage resilient if it satisfies the following property: if X and Y are independent (n,k)-sources, $\mathcal{A}_1 = (f_1,g_1),\ldots,\mathcal{A}_t = (f_t,g_t)$ are t arbitrary 2-split-state tampering functions and $h : \{0,1\}^n \to \{0,1\}^\mu$ is an arbitrary leakage function, then there exists a random variable $D_{\overrightarrow{f},\overrightarrow{g},h}$ on $(\{0,1\}^m \cup \{\mathsf{same}^*\})^t$ which is independent of the random variables X and Y, such that*

$$|\mathsf{2NMExt}(X,Y), \mathsf{2NMExt}(f_1(X,h(Y)), g_1(Y)), \ldots, \mathsf{2NMExt}(f_t(X,h(Y)), g_t(Y)) - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, U_m)| < \varepsilon$$

where both U_m's refer to the same uniform m-bit string.

We now prove the following lemma which states that any $(2,t)$-non-malleable extractor is also a leakage-resilient non-malleable extractor. A similar result was also shown in [GKP+18] and we include it here for the sake of completeness.

Lemma 5.3 ([GKP+18]) *Let $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be a $(2,t)$-non-malleable extractor at min-entropy k and error ε. For any function $h : \{0,1\}^n \to \{0,1\}^\mu$, $\mathsf{2NMExt}$ is μ-leakage resilient at min-entropy k' and error 2ε for any $n \geq k' \geq k + \mu + \log 1/\varepsilon$.*

Proof Let us fix the t tampering functions $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ and the leakage function h. For any $\tau \in \{0,1\}^\mu$, let $h^{-1}(\tau)$ be the set of all $y \in \{0,1\}^n$ such that $h(y) = \tau$. Let X, Y be independent (n, k') sources. Consider the following random variable $\mathcal{D}_0$.

$\underline{\mathcal{D}_0}$: Sample $x \sim X$ and $y \sim Y$ and compute $\tau = h(y)$. Output $\mathsf{2NMExt}(x,y), \mathsf{2NMExt}(f_1(x, \tau), g_1(y)), \ldots, \mathsf{2NMExt}(f_t(x,\tau), g_t(y))$.

Now, we will define another random variable $\mathcal{D}_1$ and argue that it is identically distributed to $\mathcal{D}_0$.

$\underline{\mathcal{D}_1}$: Sample $x \sim X$, $y' \sim Y$ and compute $\tau = h(y')$. Sample $y \sim Y | h(Y) = \tau$ and output $\mathsf{2NMExt}(x,y), \mathsf{2NMExt}(f_1(x,\tau), g_1(y)), \ldots, \mathsf{2NMExt}(f_t(x,\tau), g_t(y))$.

$\mathcal{D}_1$ is identical to $\mathcal{D}_0$ since sampling from Y and computing τ is equivalent to first sampling τ randomly (from the correct distribution) and then sampling Y conditioned on $h(Y) = \tau$.

We define a $\tau \in \{0,1\}^\mu$ to be good if $H_\infty(Y|h(Y) = \tau) \geq k$ and bad otherwise. Now, from Lemma 3.2, we infer that $\Pr_\tau[\tau \text{ is } \mathsf{bad}] \leq \varepsilon$ and hence, $\Pr_\tau[\tau \text{ is } \mathsf{good}] \geq 1 - \varepsilon$. Conditioned on τ being good, the random variables X and $Y|h(Y) = \tau$ (denoted by the distribution $\overline{Y}$) in $\mathcal{D}_1$ are independent (n,k)-sources. Further, if we define the left source tampering functions $f_1^\tau, \ldots, f_t^\tau$ as $f_i^\tau(X) = f(X, \tau)$ then $(f_1^\tau, g_1), \ldots, (f_t^\tau, g_t)$ are split-state tampering functions. This means that there exists a random variable $D_{\overrightarrow{f^\tau}, \overrightarrow{g}}$ such that

$$|\mathsf{2NMExt}(X, \overline{Y}), \mathsf{2NMExt}(f_1^\tau(X), g_1(\overline{Y})), \ldots, \mathsf{2NMExt}(f_t^\tau(X), g_t(\overline{Y})) - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f^\tau}, \overrightarrow{g}}, U_m)| < \varepsilon$$

We now define the random variable $D_{\overrightarrow{f}, \overrightarrow{g}, h}$ as follows. Sample an independent $y' \sim Y$ and compute $\tau = h(y')$. Now, output $D_{\overrightarrow{f^\tau}, \overrightarrow{g}}$.

$$\begin{aligned}
|\mathcal{D}_1 - D_{\overrightarrow{f}, \overrightarrow{g}, h}| &= \sum_\tau \Pr[h(Y) = \tau] \Big| \mathcal{D}_1 | \tau - D_{\overrightarrow{f^\tau}, \overrightarrow{g}} \Big| \\
&\leq \sum_\tau \Pr[h(Y) = \tau \wedge \tau \in \mathsf{good}] \Big| \mathcal{D}_1 | \tau - D_{\overrightarrow{f^\tau}, \overrightarrow{g}} \Big| + \sum_\tau \Pr[h(Y) = \tau \wedge \tau \in \mathsf{bad}] \\
&\leq 2\varepsilon
\end{aligned}$$

■

5.2 Bootstrapping

We will now show how to bootstrap a leakage-resilient non-malleable extractor to a strong leakage-resilient non-malleable extractor.

Building Blocks and Parameters. Let $n_1, n_2 \in \mathbb{N}$ and let Δ be another parameter that will denote the entropy loss. Let ε denote the error parameter and $p(\cdot)$ be any polynomial. In our construction, we will use the following building blocks and set the parameters as shown below.

- Let $\mathsf{2Ext} : \{0,1\}^{n_1'} \times \{0,1\}^{n_2'} \to \{0,1\}^d$ be a a strong two sources extractor at min-entropy $(n_1' - \Delta - p(n_2) - \log(1/\varepsilon), n_2' - \Delta - \log(1/\varepsilon))$ and error ε.

- Let $\mathsf{Ext} : \{0,1\}^{n_1-n_1'} \times \{0,1\}^d \to \{0,1\}^{n_2-n_2'}$ be a strong seeded extractor at min-entropy $n_1 - n_1' - \Delta - p(n_2) - \log(1/\varepsilon)$ and error ε.
- Fix $\mu = n_2' t$. Let $\mathsf{2NMExt} : \{0,1\}^{n_2-n_2'} \times \{0,1\}^{n_2-n_2'} \to \{0,1\}^m$ be a $(2,t)$-non-malleable extractor at min-entropy $n_2 - n_2' - \Delta - \mu - 2\log(1/\varepsilon)$ and error ε. By Lemma 5.3, we infer that $\mathsf{2NMExt}$ is μ-leakage-resilient for min-entropy $n_2 - n_2' - \Delta - \log(1/\varepsilon)$ and error 2ε.
- We set $n_1' = n_2 + p(n_2)$, $n_2' = 3\Delta$, and $n_1 \geq 4n_2 + 2p(n_2)$.

Construction 1. On input $((x_1, x_2), (y_1, y))$ where $x_1 \in \{0,1\}^{n_1'}, y_1 \in \{0,1\}^{n_2'}$, $x_2 \in \{0,1\}^{n_1-n_1'}$, and $y \in \{0,1\}^{n_2-n_2'}$, the function $\mathsf{2SLNMExt}$ is computed as follows:

1. Compute $s = \mathsf{2Ext}(x_1, y_1)$.
2. Compute $x = \mathsf{Ext}(x_2, s)$.
3. Output $\mathsf{2NMExt}(x, y)$.

Theorem 5.4 *For any polynomial $p(\cdot)$, $\mathsf{2SLNMExt}$ described in construction 1 is a p-strong leakage-resilient, $(2,t)$-non-malleable extractor at min-entropy $(n_1 - \Delta, n_2 - \Delta)$ with error 8ε.*

Proof Let us fix the t tampering functions $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ and the leakage function h. For any $\tau \in \{0,1\}^\mu$, let $h^{-1}(\tau)$ be the set of all $y \in \{0,1\}^n$ such that $h(y) = \tau$. Let (X_1, X_2) and (Y_1, Y) be two independent $(n_1, n_1 - \Delta)$ and $(n_2, n_2 - \Delta)$ sources. Consider the following random variable $\mathcal{D}_0$.

$\underline{\mathcal{D}_0}$: Sample $(x_1, x_2) \sim (X_1, X_2)$, $(y_1, y) \sim (Y_1, Y)$, and compute $\tau = h(x_1, x_2)$. Output $\mathsf{2SLNMExt}((x_1, x_2), (y_1, y)), \mathsf{2SLNMExt}(f_1(x_1, x_2), g_1((y_1, y), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x_1, x_2), g_t((y_1, y), \tau))$.

Now, we will define another random variable $\mathcal{D}_1$ and argue that it is identically distributed to $\mathcal{D}_0$.

$\underline{\mathcal{D}_1}$: Sample $(x_1', x_2') \sim (X_1, X_2)$, $(y_1, y) \sim (Y_1, Y)$ and compute $\tau = h(x_1', x_2')$, $s = \mathsf{2Ext}(x_1', y_1)$, $x = \mathsf{Ext}(x_2, s)$. Sample $(x_1, x_2) \sim (X_1, X_2) | (h(X_1, X_2) = \tau \wedge \mathsf{2Ext}(X_1, y_1) = s \wedge \mathsf{Ext}(X_2, s) = x)$. Output $\mathsf{2SLNMExt}((x_1, x_2), (y_1, y)), \mathsf{2SLNMExt}(f_1(x_1, x_2), g_1((y_1, y), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x_1, x_2), g_t((y_1, y), \tau))$.

The only difference between $\mathcal{D}_0$ and $\mathcal{D}_1$ is the way we sample (x_1, x_2). Notice that in $\mathcal{D}_1$, for any y_1, y, the samples (x_1, x_2) and (x_1', x_2') are from the same distribution. Also, notice that the sample (x_1', x_2') in $\mathcal{D}_1$ and the sample (x_1, x_2) in $\mathcal{D}_0$ are from identical distributions. Therefore, $\mathcal{D}_0$ and $\mathcal{D}_1$ are identical. We now define another distribution $\mathcal{D}_2$.

$\underline{\mathcal{D}_2}$: Sample $(x_1', x_2') \sim (X_1, X_2)$, $(y_1, y) \sim (Y_1, Y)$ and compute $\tau = h(x_1', x_2')$. Sample $s \leftarrow \{0,1\}^d$ and compute $x = \mathsf{Ext}(x_2', s)$. Sample $(x_1, x_2) \sim (X_1, X_2) | (h(X_1, X_2) = \tau \wedge \mathsf{2Ext}(X_1, y_1) = s \wedge \mathsf{Ext}(X_2, s) = x)$. Output $\mathsf{2NMExt}((x_1, x_2), (y_1, y)), \mathsf{2SLNMExt}(f_1(x_1, x_2), g_1((y_1, y), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x_1, x_2), g_t((y_1, y), \tau))$.

The only difference between $\mathcal{D}_1$ and $\mathcal{D}_2$ is that in $\mathcal{D}_1$, s is computed as $\mathsf{2Ext}(x'_1, y_1)$ whereas in $\mathcal{D}_2$ it is sampled uniformly at random. Now, from Lemma 3.2, we infer the following:

$$\Pr_{h(X_1,X_2),X_2}[H_\infty(X_1|h(X_1,X_2),X_2) \geq n'_1 - \Delta - p(n_2) - \log(1/\varepsilon)] \geq 1 - \varepsilon \tag{5.1}$$

and,

$$\Pr_{Y}[H_\infty(Y_1|Y) \geq n'_2 - \Delta - \log(1/\varepsilon)] \geq 1 - \varepsilon \tag{5.2}$$

It now follows that with probability at least $1 - 2\varepsilon$ over the randomness of sampling τ, y, and x_2, that $X_1|\tau, x_2$ and $Y_1|y$ are independent random variables with min-entropy at least $(n'_1 - \Delta - p(n_2) - \log(1/\varepsilon), n'_2 - \Delta - \log(1/\varepsilon))$. It now follows that since $\mathsf{2Ext}$ is a strong two-source extractor that $\mathcal{D}_1$ and $\mathcal{D}_2$ are ε-close when τ, y, x_2 satisfy the above property. Hence, $\mathcal{D}_1, \mathcal{D}_2$ are 3ε-close.

$\underline{\mathcal{D}_3}$: Sample $(x'_1, x'_2) \sim (X_1, X_2)$, $(y_1, y) \sim (Y_1, Y)$ and compute $\tau = h(x'_1, x'_2)$. Sample $s \leftarrow \{0,1\}^d$ and sample $x \sim U_{n_2 - n'_2}$. Sample $(x_1, x_2) \sim (X_1, X_2)|(h(X_1, X_2) = \tau \wedge \mathsf{2Ext}(X_1, y_1) = s \wedge \mathsf{Ext}(X_2, s) = x)$. Output $\mathsf{2SLNMExt}((x_1, x_2), (y_1, y)), \mathsf{2SLNMExt}(f_1(x_1, x_2), g_1((y_1, y), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x_1, x_2), g_t((y_1, y), \tau))$.

The only difference between $\mathcal{D}_2$ and $\mathcal{D}_3$ is in the way we sample x. In $\mathcal{D}_2$, x is set as $\mathsf{Ext}(x_2, s)$ but in $\mathcal{D}_2$ it sampled uniformly at random from $U_{n_2 - n'_2}$. We define a $(\tau, x_1) \in \{0,1\}^{p(n_2)} \times \{0,1\}^{n'_1}$ to be good if $H_\infty(X_2|h(x_1, X_2) = \tau, X_1 = x_1) \geq n_1 - n'_1 - \Delta - p(n_2) - \log 1/\varepsilon$. Otherwise, we call a (τ, x_1) to be bad. Now, from Lemma 3.2, we infer that $\Pr_{\tau,x_1}[(\tau, x_1) \text{ is } \mathsf{bad}] \leq \varepsilon$. Conditioned on (τ, x_1) being good, $X_2|h(x_1, X_2) = \tau, X_1 = x_1)$ is a $(n_1 - n'_1, n_1 - n'_1 - \Delta - p(n_2) - \log 1/\varepsilon)$-source independent of the seed s. Since Ext is a strong seeded extractor at min-entropy $n_1 - n'_1 - \Delta - p(n_2) - \log(1/\varepsilon)$, we have the sample $\mathsf{Ext}(x_2, s)$ in $\mathcal{D}_2$ is from a distribution that is ε-close to the uniform distribution even given the seed s. Thus, when (τ, x_1) is good, $\mathcal{D}_2$ is ε-close to $\mathcal{D}_3$. Hence,

$$\begin{aligned}
|\mathcal{D}_2 - \mathcal{D}_3| &= \sum_{\tau,x_1} \Pr[h(X_1,X_2) = \tau, X_1 = x_1] \Big| \mathcal{D}_2|(\tau,x_1) - \mathcal{D}_3|(\tau,x_1) \Big| \\
&\leq \sum_{\tau,x_1} \Pr[(h(X_1,X_2) = \tau, X_1 = x_1) \wedge (\tau,x_1) \in \mathsf{good}] \Big| \mathcal{D}_2|(\tau,x_1) - \mathcal{D}_3|(\tau,x_1) \Big| + \\
&\qquad \sum_{\tau,x_1} \Pr[(h(X_1,X_2) = \tau, X_1 = x_1) \wedge (\tau,x_1) \in \mathsf{bad}] \\
&< 2\varepsilon.
\end{aligned}$$

We now show that $\mathcal{D}_3$ is 3ε-close to the simulated distribution. Towards this, we define a split-state tampering function family $\overrightarrow{f}', \overrightarrow{g}'$ and a leakage function L' against the underlying $\mathsf{2NMExt}$.

- **Shared Randomness.** Sample $(x'_1, x'_2) \sim (X_1, X_2)$ and compute $\tau = h(x'_1, x'_2)$. Sample $y_1 \leftarrow Y_1$, $s \leftarrow \{0,1\}^d$. Sample an uniform random tape r for sampling from the distribution $(X_1, X_2)|h(X_1, X_2) = \tau \wedge \mathsf{2Ext}(X_1, y_1) = s \wedge \mathsf{Ext}(X_2, s) = z$ for any $z \in \{0,1\}^{n_2 - n'_2}$. The shared randomness includes (s, y_1, τ, r).

- L'**.** On input $y \in \{0,1\}^{n_2-n_2'}$, L computes $g_i(y_1, y)$ for every $i \in [t]$ to obtain $(\widetilde{y}_1^i, \widetilde{y}^i)$. It outputs $(\widetilde{y}_1^1, \ldots, \widetilde{y}_1^t)$.
- g_i'**:** g_i' on input $y \in \{0,1\}^{n_2-n_2'}$ computes $g_i(y_1, y)$ to obtain $(\widetilde{y}_1^i, \widetilde{y}^i)$. It outputs $\widetilde{y}^i$.
- f_i'**:** On input $x \in \{0,1\}^{n_2-n_2'}$ and the leakage $(\widetilde{y}_1^i, \ldots, \widetilde{y}_1^t)$, f_i' uses the random tape r to sample (x_1, x_2) from the distribution $(X_1, X_2)|h(X_1, X_2) = \tau \wedge \mathsf{2Ext}(X_1, y_1) = s \wedge \mathsf{Ext}(X_2, s) = x$.[4] It computes $f_i(x_1, x_2) = (\widetilde{x}_1^i, \widetilde{x}_2^i)$, and outputs $\mathsf{Ext}(\widetilde{x}_2^i, \mathsf{2Ext}(\widetilde{x}_1^i, \widetilde{y}_1^i))$.

We now define $\mathcal{D}_4$ as follows.

$\underline{\mathcal{D}_4}$: Sample the shared randomness (s, y_1, τ, r) as described above. Use (s, y_1, τ, r) as the shared randomness and define $\overrightarrow{f}', \overrightarrow{g}', L'$ as above. Sample $x \sim U_{n_2-n_2'}$ and $y \sim Y|Y_1 = y_1$ and output $\mathsf{2NMExt}(x, y), \mathsf{2NMExt}(f_1'(x, L'(y)), g_1'(y)), \ldots, \mathsf{2NMExt}(f_t'(x, L'(y)), g_t'(y))$.

It can be easily seen that $\mathcal{D}_4$ is identical to $\mathcal{D}_3$. Further from Lemma 3.2, we infer that with probability at least $1-\varepsilon$ over the randomness of y_1, $Y|Y_1 = y_1$ (denoted by $\overline{Y}$) is a $(n_2 - n_2', n_2 - n_2' - \Delta - \log(1/\varepsilon))$ source. Since $\mathsf{2NMExt}$ is a μ-leakage-resilient $(2,t)$-non-malleable extractor for min-entropy $n_2 - n_2' - \Delta - \log(1/\varepsilon)$, it follows that with probability at least $1-\varepsilon$ over the shared randomness (y_1, s, τ, r), there exists a distribution $D_{\overrightarrow{f}', \overrightarrow{g}', L'}$ such that

$$|\mathsf{2NMExt}(X, \overline{Y}), \mathsf{2NMExt}(f_1'(X, L'(\overline{Y})), g_1'(\overline{Y})), \ldots, \mathsf{2NMExt}(f_t'(X, L'(\overline{Y})), g_t'(\overline{Y})) \\ -U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}', \overrightarrow{g}', L'}, U_m)| < 2\varepsilon.$$

We now define the random variable $D_{\overrightarrow{f}, \overrightarrow{g}, h}$ as follows. Sample (s, y_1, τ, r) as above and define $\overrightarrow{f}', \overrightarrow{g}', L'$ as described. Now, output $D_{\overrightarrow{f}', \overrightarrow{g}', L'}$. It is easy to see from the above equation that $D_{\overrightarrow{f}, \overrightarrow{g}, h}$ and $\mathcal{D}_4$ are 3ε-close. ■

5.3 Instantiation

We now instantiate the above result with explicit protocols mentioned in section 3.

Parameters For any polynomial p and tampering degree t, the parameters can be set as follows.

1. Let $\Delta = \gamma_1 n_2$ and $\log(1/\varepsilon) = C_1 n_2^{\gamma_2}$ where $0 < C_1, \gamma_1, \gamma_2 < 1$.
2. Let $n_1' = n_2 + p(n_2)$, $n_2' = 3\Delta$, $n_1 = 4n_2 + 2p(n_2)$ and $n_2'' = n_2 - n_2'$.
3. Since $n_2' = \Omega(n_2) = \Omega(\log(n_1/\varepsilon))$, we can instantiate $\mathsf{2Ext}$ from Theorem 3.15 at min-entropy $(C_2 n_2', 0.6n_2')$ and error ε with output length $d = C_3 n_2'$ for some constant C_2, C_3.
4. Instantiate Ext from Theorem 3.9 at min-entropy $2n_2$ and error ε.
5. Instantiate $\mathsf{2NMExt}$ from Corollary 4.9 at min-entropy $n_2''(1-\gamma)$ and error ε and $m = n_2''^{\Omega(1)}$.

[4]Since each f_i' uses the same random tape r, it follows that each f_i' will compute the same (x_1, x_2) on the same input x.

6. Set γ_1, C_1 such that $\gamma_1(1+3C_2)+C_1 \leq 1$ and $\gamma_1 + C_1 \leq 0.4$. Then, the min-entropy requirements for $\mathsf{2Ext}$ and Ext are satisfied.

7. Set γ_1, C_1 such that $(3t+3\gamma+1)\gamma_1+2C_1 \leq \gamma$. Then, the min-entropy requirement for $\mathsf{2NMExt}$ is satisfied.

8. Set $\gamma_1 \leq \frac{1}{6}$ so that $n_2'' \geq \frac{1}{2}n_2$ and $m = n_2^{\Omega(1)}$.

We summarize the instantiation with the following corollary.

Corollary 5.5 *For any polynomial p and constant t, there exists constants $\gamma, n_0 > 0$ such that for any $n_2 > n_0$, there exists an p-strong leakage-resilient $(2,t)$-non-malleable extractor $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ with min-entropy $(n_1 - \gamma n_2, n_2 - \gamma n_2)$ and error $2^{-n_2^{\Omega(1)}}$, where $n_1 = 4n_2 + 2p(n_2)$.*

5.4 Efficient Pre-image Sampleability

In this subsection, we give a construction of strong leakage-resilient non-malleable extractor with *efficient pre-image sampleability*. However, if we rely on the construction in section 5, the instantiation discussed in section 5.3 is not efficiently preimage-sampleable since the underlying strong two sources extractor from [Raz05] is not known to be efficiently preimage-sampleable. To solve this problem, we modify the construction of the strong leakage-resilient $(2,t)$-non-malleable extractors, which has efficient preimage sampleability but only works for the case when the sources have full min-entropy.

Building Blocks and Parameters. In our construction, we will use the following building blocks and set the parameters as shown below.

- Let $\mathsf{Ext} : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^{n_2-d}$ be a strong seeded extractor at min-entropy $2n_2$ and error ε.
- Fix $\mu = dt$. Let $\mathsf{2NMExt} : \{0,1\}^{n_2-d} \times \{0,1\}^{n_2-d} \to \{0,1\}^m$ be a $(2,t)$-non-malleable extractor at min-entropy $n_2 - d - dt - \log(1/\varepsilon)$ and error ε. By Lemma 5.3, we infer that $\mathsf{2NMExt}$ is μ-leakage-resilient at min-entropy $n_2 - d$ and error 2ε.
- We set $n_1 > p(n_2) + 2n_2 + \log(1/\varepsilon)$.

Construction 2. On input $(x', (y,s)) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$, the function $\mathsf{2SLNMExt}$ is computed as follows:

1. Compute $x = \mathsf{Ext}(x', s)$.
2. Output $\mathsf{2NMExt}(x, y)$.

Theorem 5.6 *In the above construction, $\mathsf{2SLNMExt}$ is a p-strong leakage-resilient, $(2,t)$-non-malleable extractor at full with error 4ε.*

Proof Let us fix the t tampering functions $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ and the leakage function h. For any $\tau \in \{0,1\}^\mu$, let $h^{-1}(\tau)$ be the set of all $x \in \{0,1\}_1^n$ such that $h(x) = \tau$. Let X', Y be two independent sources with full entropy. Consider the following random variable $\mathcal{D}_0$.

$\underline{\mathcal{D}_0}$: Sample $x' \sim X'$, $y \sim Y$, $s \leftarrow \{0,1\}^d$ and compute $\tau = h(x')$. Output $\mathsf{2SLNMExt}(x', (y, s)), \mathsf{2SLNMExt}(f_1(x'), g_1((y,s),\tau)), \ldots, \mathsf{2SLNMExt}(f_t(x'), g_t((y,s),\tau))$.

Now, we will define another random variable $\mathcal{D}_1$ and argue that it is identically distributed to $\mathcal{D}_0$.

$\underline{\mathcal{D}_1}$: Sample $\overline{x} \sim X'$, $y \sim Y$, $s \leftarrow \{0,1\}^d$ and compute $\tau = h(\overline{x})$, $x = \mathsf{Ext}(\overline{x}, s)$. Sample $x' \sim X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = x$. Output $\mathsf{2SLNMExt}(x', (y,s)), \mathsf{2SLNMExt}(f_1(x'), g_1((y,s), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x'), g_t((y,s),\tau))$.

The only difference between $\mathcal{D}_0$ and $\mathcal{D}_1$ is the way we sample x'. Notice that in $\mathcal{D}_1$, for any y, s, the samples x' and $\overline{x}$ are from the same distribution. Also, notice that the sample $\overline{x}$ in $\mathcal{D}_1$ and the sample x' in $\mathcal{D}_0$ are from identical distributions. Therefore, $\mathcal{D}_0$ and $\mathcal{D}_1$ are identical. We now define another distribution $\mathcal{D}_2$.

$\underline{\mathcal{D}_2}$: Sample $\overline{x} \sim X'$, $x \sim U_{n_2-d}$, $y \sim Y$, $s \leftarrow \{0,1\}^d$ and compute $\tau = h(\overline{x})$. Sample $x' \sim X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = x$. Output $\mathsf{2SLNMExt}(x', (y,s)), \mathsf{2SLNMExt}(f_1(x'), g_1((y,s), \tau)), \ldots, \mathsf{2SLNMExt}(f_t(x'), g_t((y,s),\tau))$.

The only difference between $\mathcal{D}_1$ and $\mathcal{D}_2$ is in the way we sample x. In $\mathcal{D}_1$, x is set as $\mathsf{Ext}(\overline{x}, s)$ but in $\mathcal{D}_2$ it sampled uniformly at random from U_{n_2}. We define a $\tau \in \{0,1\}^{p(n)}$ to be good if $H_\infty(X'|h(X') = \tau) \geq 2n_2$. Otherwise, we call a τ to be bad. Now, from Lemma 3.2, we infer that $\Pr_\tau[\tau \text{ is } \mathsf{bad}] \leq \varepsilon$. Conditioned on τ being good, $X'|h(X') = \tau$ is a $(n_1, 2n_2)$-source independent of the seed s. Since Ext is a strong seeded extractor at min-entropy $2n_2$, we have the sample $\mathsf{Ext}(\overline{x}, s)$ in $\mathcal{D}_1$ is from a distribution that is ε-close to the uniform distribution even given the seed s. Thus, when τ is good, $\mathcal{D}_1$ is ε-close to $\mathcal{D}_2$. Hence,

$$\begin{aligned}
|\mathcal{D}_1 - \mathcal{D}_2| &= \sum_\tau \Pr[h(X') = \tau] \Big| \mathcal{D}_1|\tau - \mathcal{D}_2|\tau \Big| \\
&\leq \sum_\tau \Pr[h(X') = \tau \wedge \tau \in \mathsf{good}] \Big| \mathcal{D}_1|\tau - \mathcal{D}_2|\tau \Big| + \sum_\tau \Pr[h(X') = \tau \wedge \tau \in \mathsf{bad}] \\
&< 2\varepsilon.
\end{aligned}$$

We now show that $\mathcal{D}_2$ is ε-close to the simulated distribution. Towards this, we define a split-state tampering function family $\overrightarrow{f}', \overrightarrow{g}'$ and a leakage function L' against the underlying $\mathsf{2NMExt}$.

- **Shared Randomness.** Sample $\overline{x} \sim X'$ and compute $\tau = h(\overline{x})$. Sample $s \leftarrow \{0,1\}^d$. Sample an uniform random tape r for sampling from the distribution $X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = z$ for any $z \in \{0,1\}^{n_2-d}$. The shared randomness includes (s, τ, r).
- L'**.** On input $y \in \{0,1\}^{n_2-d}$, L computes $g_i(y, s)$ for every $i \in [t]$ to obtain $(\widetilde{y}_i, \widetilde{s}_i)$. It outputs $(\widetilde{s}_1, \ldots, \widetilde{s}_t)$.

- g_i'**:** For every $i \in [t]$, g_i' on input $y \in \{0,1\}^{n_2-d}$ computes $g_i(y,s)$ to obtain $(\widetilde{y}_i, \widetilde{s}_i)$. It outputs $\widetilde{y}_i$.
- f_i'**:** For every $i \in [t]$, on input $x \in \{0,1\}^{n_2-d}$ and the leakage $(\widetilde{s}_1, \ldots, \widetilde{s}_t)$, f_i' uses the random tape r to sample x' from the distribution $X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = x$.[5] It outputs $\mathsf{Ext}(f_i(x'), \widetilde{s}_i)$.

We now define $\mathcal{D}_3$ as follows.

$\underline{\mathcal{D}_3}$: Sample $\overline{x} \leftarrow X'$, $x \sim U_{n_2-d}$, $y \sim Y$, $s \leftarrow \{0,1\}^d$ and compute $\tau = h(\overline{x})$. Sample an uniform random tape r for sampling from the distribution $X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = z$ for any $z \in \{0,1\}^{n_2-d}$. Use (s, τ, r) as the shared randomness and define $\overrightarrow{f}', \overrightarrow{g}', L'$ as above. Output $\mathsf{2NMExt}(x,y), \mathsf{2NMExt}(f_1'(x, L'(y)), g_1'(y)), \ldots, \mathsf{2NMExt}(f_t'(x, L'(y)), g_t'(y))$.

It can be easily seen that $\mathcal{D}_2$ is identical to $\mathcal{D}_3$. Since $\mathsf{2NMExt}$ is a dt-leakage-resilient $(2,t)$-non-malleable extractor at full min-entropy, it follows that for any choice of the shared randomness (s, τ, r), there exists a distribution $D_{\overrightarrow{f}', \overrightarrow{g}', L'}$ such that

$$|\mathsf{2NMExt}(X,Y), \mathsf{2NMExt}(f_1'(X, L'(Y)), g_1'(Y)), \ldots, \mathsf{2NMExt}(f_t'(X, L'(Y)), g_t'(Y)) \\ -U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}', \overrightarrow{g}', L'}, U_m)| < \varepsilon.$$

We now define the random variable $D_{\overrightarrow{f}, \overrightarrow{g}, h}$ as follows. Sample independent $\overline{x} \sim X'$, $s \leftarrow \{0,1\}^d$ and compute $\tau = h(\overline{x})$. Sample an uniform random tape r for sampling from the distribution $X'|h(X') = \tau \wedge \mathsf{Ext}(X', s) = z$ for any $z \in \{0,1\}^{n_2-d}$. Use (s, τ, r) as the shared randomness and define $\overrightarrow{f}', \overrightarrow{g}', L'$ as above. Now, output $D_{\overrightarrow{f}', \overrightarrow{g}', L'}$. Then, we have

$$\begin{aligned} |\mathcal{D}_3 \;&-\; U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}, h}, U_m)| \\ &= \sum_{s,\tau,r} \Pr[h(X') = \tau \wedge U_d = s \wedge R = r] \Big| \mathcal{D}_3|_{s,\tau,r} - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}', \overrightarrow{g}', L'}, U_m) \Big| \\ &< \varepsilon. \end{aligned}$$

■

5.4.1 Instantiation

We now instantiate the above construction in a similar way as section 5.3 and show that the instantiation is efficiently preimage-sampleable.

Parameters

1. Let $n_1 = p(n_2) + 4n_2$.
2. Let $\mathsf{Ext} : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^{n_2-d}$ be the linear strong seeded extractor at min-entropy $2n_2$ with $d = O(\log^2(n/\varepsilon))$ from Theorem 3.10.

[5] Since each f_i' uses the same random tape r, it follows that each f_i' will compute the same x' on the same input x.

3. Let γ be the constant in Theorem 3.17. Let $0 < \gamma_1, \gamma_2, \gamma_3 < 1$ be small constants such that $\gamma_2 + 2\gamma_3 < \gamma$. Let $t = C_2 n_2^{\gamma_2}$ and $\log(1/\varepsilon) = C_3 n_2^{\gamma_3}$ and $d = O(\log^2(n/\varepsilon)) = C_1 n_2^{\gamma_1}$ for some constants $0 < C_1, C_2, C_3 < 1/2$ such that $(C_1 C_2 + C_3)^{1/\gamma} < 1/2$.

4. Let $n_2' = n_2 - d > n_2/2$. Let $\mathsf{2NMExt} : \{0,1\}^{n_2'} \times \{0,1\}^{n_2'} \to \{0,1\}^{n_2'^{\Omega(1)}}$ be a $(2,t)$-non-malleable extractor at min-entropy $n_2' - n_2'^{\gamma}$ with error ε from the Theorem 3.17, where we use the fact that $t < C_2 n_2^{\gamma_2} < n_2'^{\gamma}$ and $\varepsilon > 2^{-C_3 n_2^{\gamma}} > 2^{-n_2'^{\gamma}}$.

Since $(dt + \log(1/\varepsilon))^{1/\gamma} \leq (C_1 C_2 + C_3)^{1/\gamma} n_2 \leq n_2/2 < n_2'$ and thus

$$n_2 - d - dt - \log(1/\varepsilon) > n_2' - n_2'^{\gamma},$$

2NMExt satisfies the min-entropy requirement. We now show that the above instantiation also make 2SLNMExt efficiently pre-image sampleable. Given $s \in \{0,1\}^m$, pre-image sampling procedure is stated as follows:

1. Sample (x, y) uniformly from the preimage of s in 2NMExt.
2. Sample (x', s) uniformly from the preimage of x in Ext. Output $(x', (y, s))$.

Since Ext from [Tre01, RRV02] are linear and hence efficiently pre-image sampleable and 2NMExt from [CGL16] is efficiently pre-image sampleable, our construction is also efficiently pre-image sampleable.

Corollary 5.7 *For any polynomial p and n_2, there exists an efficiently pre-image sampleable p-strong leakage-resilient $(2, n_2^{\Omega(1)})$-non-malleable extractor* $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *with min-entropy (n_1, n_2) and error $2^{-n_2^{\Omega(1)}}$, where $n_1 = 4n_2 + p(n_2)$ and $m = n_2^{\Omega(1)}$.*

5.5 Some Useful Lemmas

In this subsection, we will give a couple of useful lemmas about p-strong leakage resilient non-malleable extractor. These lemmas will be used in the subsequent sections for constructing multi-source non-malleable extractors and non-malleable secret sharing. Before we give the lemmas, we define a function Sanitize below.

Definition 5.8 *The function* Sanitize *on input $\alpha, (x_1, \ldots, x_s)$ outputs $(y_1, \ldots, y_s)$ where $y_i = \mathsf{same}^*$ if $x_i = \alpha$; else, $y_i = x_i$.*

Lemma 5.9 *For some polynomial $p(\cdot)$, let* $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *is a p-strong leakage-resilient, $(2,t)$-non-malleable extractor at min-entropy $(n_1 - \Delta, n_2 - \Delta)$ and error ε. Then, $(n_1, n_1 - \Delta)$ and $(n_2, n_2 - \Delta)$ independent sources X, Y and for any set of tampering functions $\overrightarrow{f}$, $\overrightarrow{g}$ and a leakage function $h : \{0,1\}^{n_1} \to \{0,1\}^{p(n_2)}$, the following two distributions are $O(\varepsilon + t2^{-m})$.*

$\mathcal{D}_0$: Sample independent $x, x' \sim X$ and $y, y' \sim Y$. Output $\mathsf{2SLNMExt}(x,y) \circ \mathsf{2SLNMExt}(x',y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x,y), \{\mathsf{2SLNMExt}(f_i(x), g_i(y, h(x)))\}_{i \in [s]})$.

$\mathcal{D}_1$: Sample independent $x, x' \sim X$ and $y, y' \sim Y$. Output $\mathsf{2SLNMExt}(x,y) \circ \mathsf{2SLNMExt}(x',y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x',y'), \{\mathsf{2SLNMExt}(f_i(x'), g_i(y', h(x')))\}_{i \in [s]})$.

Proof

$$\begin{aligned}
\mathcal{D}_0 \quad &\approx_{\varepsilon} \quad U_m \circ \mathsf{2SLNMExt}(x', y') \circ \mathsf{Sanitize}(U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, U_m)) \\
&\approx_{t2^{-m}} \quad U_m \circ \mathsf{2SLNMExt}(x', y') \circ D_{\overrightarrow{f},\overrightarrow{g},h} \\
&\approx_{\varepsilon} \quad \mathsf{2SLNMExt}(x, y) \circ \mathsf{2SLNMExt}(x', y') \circ D_{\overrightarrow{f},\overrightarrow{g},h} \\
&\approx_{\varepsilon} \quad \mathsf{2SLNMExt}(x, y) \circ U_m \circ D_{\overrightarrow{f},\overrightarrow{g},h} \\
&\approx_{t2^{-m}} \quad \mathsf{2SLNMExt}(x, y) \circ U_m \circ \mathsf{Sanitize}(U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, U_m)) \\
&\approx_{\varepsilon} \quad \mathcal{D}_1
\end{aligned}$$

Here, the first and the last equations follow from the security of 2SLNMExt. The second and the fifth equations follow from the fact that $D_{\overrightarrow{f},\overrightarrow{g},h} = \mathsf{Sanitize}(U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, U_m))$ with probability $1 - t2^{-m}$. The third and fourth equations follow from the fact that 2SLNMExt is a two-source extractor. ■

Lemma 5.10 ([ADKO15]) *Let* $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *be a strong p-leakage-resilient* $(2,t)$*-non-malleable extractors with error* ε*. Construct* $(\mathsf{SEnc}, \mathsf{SDec})$ *as following: For any pair* $s_0, s_1 \in \{0,1\}^m$*, let* $(X_0, Y_0) \leftarrow \mathsf{2SLNMExt}^{-1}(s_0)$ *and* $(X_1, Y_1) \leftarrow \mathsf{2SLNMExt}^{-1}(s_1)$*. Then,* $|X_0 - X_1| < 2\varepsilon(2^{m+1} + 1)$ *and* $|Y - Y'| < 2\varepsilon(2^{m+1} + 1)$*.*

Proof We show that a 2SLNMExt implies a non-malleable code (via a reduction given by [CG14]) and this is sufficient to show the above lemma using [ADKO15] who showed that for any 2-split state non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ with error ε' and for any two messages $s_0, s_1 \in \{0,1\}^m$, it holds that,

$$|L_0 - L_1| < 2\varepsilon', |R_0 - R_1| < 2\varepsilon'$$

where $(L_0, R_0) \leftarrow \mathsf{Enc}(s_0)$ and $(L_1, R_1) \leftarrow \mathsf{Enc}(s_1)$.

Towards this goal, we construct $(\mathsf{SEnc}, \mathsf{SDec})$ as following:

- $\mathsf{SEnc} : \{0,1\}^m \to \{0,1\}^{n_1} \times \{0,1\}^{n_2}$ such that $\mathsf{SEnc}(s)$ outputs a uniform sample from $\mathsf{2SLNMExt}^{-1}(s)$.
- $\mathsf{SDec} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ such that $\mathsf{SDec}(x, y) = \mathsf{2SLNMExt}(x, y)$ for any $(x, y) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2}$.

It is clear that for any $s \in \{0,1\}^m$, $\mathsf{SDec}(\mathsf{SEnc}(s)) = s$ with probability 1, so $(\mathsf{SEnc}, \mathsf{SDec})$ is a valid coding scheme. To prove its non-malleability, let us fix the t tampering functions $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ and the leakage function h (we show that such an extractor implies a strong form of leakage-resilient non-malleable codes). Since the 2SLNMExt is a strong p-leakage-resilient $(2,t)$-non-malleable extractor with error ε, by definition 5.1, there exists a distribution $D_{\overrightarrow{f},\overrightarrow{g},h}$ such that for independent uniform random variables $X \in \{0,1\}^{n_1}$ and $Y \in \{0,1\}^{n_2}$, it holds that

$$\begin{aligned}
|\mathsf{2SLNMExt}(X, Y), \mathsf{2SLNMExt}(f_1(X), g_1(h(X), Y)), \ldots, \mathsf{2SLNMExt}(f_t(X), g_t(h(X), Y)) \\
-\mathsf{2SLNMExt}(X, Y), \mathsf{copy}^{(t)}(D_{\overrightarrow{f},\overrightarrow{g},h}, \mathsf{2SLNMExt}(X, Y))| < 2\varepsilon.
\end{aligned}$$

For $s \sim U_m$, we note that $\mathsf{2SLNMExt}(x, y)$ where $(x, y) \leftarrow \mathsf{SEnc}(s)$ is ε-close to $\mathsf{2SLNMExt}(x', y')$ where $x' \sim U_{n_1}$ and $y' \sim U_{n_2}$. Thus, for any $s \in \{0,1\}^m$, by Lemma 3.7, it holds that

$$|\mathsf{2SLNMExt}(f_1(X), g_1(h(X), Y)), \ldots, \mathsf{2SLNMExt}(f_t(X), g_t(h(X), Y)) - \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}, h}, s)| < \varepsilon(2^{m+1} + 1),$$

where X, Y is uniformly sample from the preimage of s in $\mathsf{2SLNMExt}$. Therefore, by the definition of SEnc and SDec we have

$$|\mathsf{SDec}(f_1(X), g_1(h(X), Y)), \ldots, \mathsf{SDec}(f_t(X), g_t(h(X), Y)) - \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}, h}, s)| < \varepsilon(2^{m+1} + 1),$$

where $X, Y = Enc(s)$. ■

6 Multi-Source Non-Malleable Extractors

In this section, we will define and construct multi-source non-malleable extractors against a wide class of tampering function families.

6.1 Definition

Definition 6.1 (Multi-Source Non-Malleable Extractors) *A function* $\mathsf{MNMExt} : \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n \to \{0,1\}^m$ *is a s-source non-malleable extractor against a tampering family* $\mathcal{F}$ *at min-entropy* k *and error* ε *if it satisfies the following property: If* $X_1, \ldots, X_s$ *are independent* (n, k)*-sources and for any* $f \in \mathcal{F}$*, there exists a random variable* D_f *with support on* $\{0,1\}^m \cup \{\mathsf{same}^*\}$ *that is independent of* $(X_1, \ldots, X_s)$ *such that*

$$|\mathsf{MNMExt}(X_1, \ldots, X_s) \circ \mathsf{MNMExt}(f(X_1, \ldots, X_s)) - U_m \circ \mathsf{copy}(D_f, U_m)| \leq \varepsilon$$

where both U_m*'s refer to the same uniform* m*-bit string and* $\mathsf{copy}(x, y) = \begin{cases} x & \text{if } x \neq \mathsf{same}^* \\ y & \text{if } x = \mathsf{same}^* \end{cases}$.

Tampering Function Family. We are interested in constructing multi-source non-malleable extractors that are secure against the tampering function families of the following form. Let $T_1, \ldots, T_s \subset [s]$. The tampering family $\mathcal{F}_{T_1,\ldots,T_s}$ consists of the set of all functions $f = (f_{T_1}, \ldots, f_{T_s})$ such that on input $(X_1, \ldots, X_s)$, f outputs $(\widetilde{X}_1, \ldots, \widetilde{X}_s)$ where for every $i \in [s]$, $f_{T_i}(\{X_j\}_{j \in T_i}) = \widetilde{X}_i$. In other words, $\widetilde{X}_i$ is generated by applying f_{T_i} on the set of sources $\{X_j\}_{j \in T_i}$. Depending on the properties required from the sets $\{T_1, \ldots, T_s\}$, we get two interesting classes of tampering functions.

- **Disjoint Tampering Family.** The disjoint tampering family $\mathcal{F}_{\mathsf{dis}}$ is the set of all $\mathcal{F}_{T_1,\ldots,T_s}$ for every possible $T_1, \ldots, T_s$ such that each T_i is non-empty, $|T_i| \leq s - 1$, and if $x \in T_i, T_j$ then $T_i = T_j$.
- **Cover-free Tampering Family.** For every $i \in [s]$, let us define $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_s$ to be the union of all the sets T_j where $i \in T_j$. The cover-free tampering family $\mathcal{F}_{\mathsf{cover-free}}$ is the set of all $\mathcal{F}_{T_1,\ldots,T_s}$ for all possible $T_1, \ldots, T_s \subset [s]$ such that for every $i \in [s]$, the size of $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_s$ is at most $s - 1$.

Observe that $\mathcal{F}_{\mathsf{dis}} \subset \mathcal{F}_{\mathsf{cover-free}}$ and hence in the rest of the section, we will focus on constructing non-malleable extractors that are secure against $\mathcal{F}_{\mathsf{cover-free}}$.

6.2 Construction

In this subsection, we will give a construction of s-source non-malleable extractor that is secure against $\mathcal{F}_{\mathsf{cover-free}}$.

Building Blocks and Parameters. In our construction, we will use the following building blocks and set the parameters as shown below. Let $n_1, n_2 \in \mathbb{N}$ and let ε denote the error and Δ denote the entropy loss parameter.

- Define the polynomial $p(\cdot)$ as $p(x) = xs^2$. Let $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ be a p-strong leakage-resilient, $(2, s)$-non-malleable extractor (see Definition 5.1). Let the min-entropy requirement of the extractor be $(n_1 - \Delta, n_2 - \Delta)$ and error be ε.
- We set $n = n_1 + sn_2$.
- We set $\varepsilon < 1/2^m$.

Construction 3. On input strings $(x_1, \ldots, x_s)$ where each $x_i \in \{0,1\}^n$, the function MNMExt is computed as follows:

1. For each $i \in [s]$, partition x_i into $(s+1)$ blocks $(x^{(i)}, y_i^{(1)}, \ldots, y_i^{(s)})$ where $x^{(i)}$ has length n_1 and each $y_i^{(j)}$ has length n_2.
2. For each $i \in [s]$, compute $y^{(i)} = y_1^{(i)} \oplus y_2^{(i)} \ldots \oplus y_s^{(i)}$.
3. Output $\mathsf{2SLNMExt}(x^{(1)}, y^{(1)}) \oplus \mathsf{2SLNMExt}(x^{(2)}, y^{(2)}) \ldots \oplus \mathsf{2SLNMExt}(x^{(s)}, y^{(s)})$.

Theorem 6.2 *Assume that* $\mathsf{2SLNMExt}$ *is a p-strong leakage resilient $(2, s)$-non-malleable extractor with error ε. Then, construction 2 is a s-source, non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy $n - \Delta + \log(1/\varepsilon)$ and error $O(s(\varepsilon + s2^{-m}))$.*

Proof Let us fix a tampering function $f = (f_{T_1}, \ldots, f_{T_s}) \in \mathcal{F}_{\mathsf{cover-free}}$. Recall that by definition, for every $i \in [s]$, the size of $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_s$ is at most $s-1$. To prove the non-malleability of construction 2, we need to show the existence of a distribution D_f such that:

$$|\mathsf{MNMExt}(X_1, \ldots, X_s) \circ \mathsf{MNMExt}(f(X_1, \ldots, X_s)) - U_m \circ \mathsf{copy}(D_f, U_m)| \leq O(s \cdot \varepsilon + s^2 2^{-m}) \quad (6.1)$$

where $X_1, \ldots, X_s$ are independent $(n, n - \Delta + \log(1/\varepsilon))$. We will show equation 6.1 through a hybrid argument where the final hybrid will give the description of the distribution D_f. Before we go to the hybrid argument, we define the following useful function.

Definition 6.3 *The function* split *takes as input $x_1, \ldots, x_s$ and does the following:*

- *Parses x_i as $(x^{(i)}, y_i^{(1)}, \ldots, y_i^{(s)})$ where $x^{(i)}$ has length n_1 and each $y_i^{(j)}$ has length n_2.*
- *For each $i \in [s]$, computes $y^{(i)} = y_1^{(i)} \oplus y_2^{(i)} \ldots \oplus y_s^{(i)}$.*
- *It outputs $\mathsf{2SLNMExt}(x^{(1)}, y^{(1)}), \mathsf{2SLNMExt}(x^{(2)}, y^{(2)}) \ldots, \mathsf{2SLNMExt}(x^{(s)}, y^{(s)})$.*

Notice that the only difference between split and MNMExt is in the last step where instead of XORing $\mathsf{2SLNMExt}(x^{(1)}, y^{(1)}), \mathsf{2SLNMExt}(x^{(2)}, y^{(2)}) \ldots, \mathsf{2SLNMExt}(x^{(s)}, y^{(s)})$, split outputs these s values.
We are ready to give the description of Hyb_j.

$\underline{\mathsf{Hyb}_j}$:

1. Uniformly sample $b_1, \ldots, b_j$ from $\{0,1\}^m$.
2. Sample $(x_1, \ldots, x_s) \sim (X_1, \ldots, X_s)$.
3. Parse x_i as $(x^{(i)}, y_i^{(1)}, \ldots, y_i^{(s)})$ where $x^{(i)}$ has length n_1 and each $y_i^{(j)}$ has length n_2.
4. For each $i \in [s]$, define $y^{(i)} = y_1^{(i)} \oplus y_2^{(i)} \ldots \oplus y_s^{(i)}$.
5. Run $f(x_1, \ldots, x_n)$ to obtain $(\widetilde{x}_1, \ldots, \widetilde{x}_s)$.
6. Let $(a_1, \ldots, a_s) := \mathsf{split}(\widetilde{x}_1, \ldots, \widetilde{x}_s)$.
7. For $i > j$, let $b_i = \mathsf{2SLNMExt}(x^{(i)}, y^{(i)})$.
8. If for any $i \in [s]$, $a_i = \mathsf{2SLNMExt}(x^{(k)}, y^{(k)})$ for some $k \in [s]$, then replace a_i with b_k.
9. Output $(b_1 \oplus \ldots \oplus b_s) \circ (a_1 \oplus \ldots \oplus a_s)$.

Notice that the output of Hyb_0 is identically distributed to the first distribution in Equation 6.1. We now show that for every $j \in [s]$, $\mathsf{Hyb}_j \approx_{O(\varepsilon + s2^{-m})} \mathsf{Hyb}_{j-1}$.

Claim 6.4 *For every $j \in [s]$, $\mathsf{Hyb}_j \approx_{O(\varepsilon + s2^{-m})} \mathsf{Hyb}_{j-1}$.*

Proof We will prove this claim via a reduction to the security of underlying $\mathsf{2SLNMExt}$. Towards this goal, we will define tampering functions $\overrightarrow{f}', \overrightarrow{g}'$ and a leakage function h'. But before we give the description of these functions, we introduce some notation. Let $T_{j_1}, \ldots, T_{j_k}$ be all the sets among $T_1, \ldots, T_s$ that contain j. We notice that by the cover-freeness property, $|\mathsf{Cover}(j)| = |T_{j_1} \cup T_{j_2} \cup \ldots T_{j_k}| \leq s - 1$. In other words, there exists some j^* such that $j^* \notin \mathsf{Cover}(j)$.

Intuition. To give the main intuition behind the proof, assume for the sake of simplicity that the sources $X_1, \ldots, X_s$ have full min-entropy. Notice that the only difference between Hyb_j and Hyb_{j-1} is that in Hyb_{j-1}, b_j is set to $\mathsf{2SLNMExt}(x^{(j)}, y^{(j)})$ whereas in Hyb_j, is sampled from $\{0,1\}^m$ uniformly. Also, notice that conditioned on fixing all the values in $(x_1, x_2, \ldots, x_s)$ in the description of Hyb_j and Hyb_{j-1} except $x^{(j)}$ and $y_{j^*}^{(j)}$, $X^{(j)}$ and $Y_{j^*}^{(j)}$ are independent sources with full min-entropy and further, these two sources have the same distribution in both Hyb_j and Hyb_{j-1}. Thus, we use the Lemma 5.9 and argue that Hyb_j is statistically close to Hyb_{j-1} by designing suitable tampering and leakage functions.

Description of $\overrightarrow{f}', \overrightarrow{g}', h'$.

- **Shared Randomness.** Sample $(x_1, \ldots, x_s) \sim (X_1, \ldots, X_s)$ and fix all the values except $x^{(j)}$ and $y^{(j)}_{j^*}$. Let x^* be a string of length n_1 bits such that for every $a \in \{0,1\}^m$, there exists some y_a such that $\mathsf{2SLNMExt}(x^*, y_a) = a$.[6]
- **Description of f'_i.** On input $\overline{x}^{(j)}$, f'_i does the following:
  1. If $i \notin \{j_1, \ldots, j_k\}$, output x^*.
  2. Else, set $x_j = (\overline{x}^{(j)}, y^{(1)}_j, \ldots, y^{(s)}_j)$.
  3. Apply $f_{T_i}(x_{T_i})$ to obtain $\widetilde{x}_i$.
  4. Parse $\widetilde{x}_i$ as $\widetilde{x}^{(i)}, \widetilde{y}^{(1)}_i \ldots, \widetilde{y}^{(s)}_i$ and output $\widetilde{x}^{(i)}$.
- **Description of h'.** On input $\overline{x}^{(j)}$, h' does the following.
  1. Set $x_j = (\overline{x}^{(j)}, y^{(1)}_j, \ldots, y^{(s)}_j)$.
  2. For every $i \in \{j_1, \ldots, j_k\}$, apply $f_{T_i}(x_{T_i})$ to obtain $\widetilde{x}_i$.
  3. Parse every $\widetilde{x}_i$ as $\widetilde{x}^{(i)}, \widetilde{y}^{(1)}_i \ldots, \widetilde{y}^{(s)}_i$ and output $\{\widetilde{y}^{(1)}_i \ldots, \widetilde{y}^{(s)}_i\}_{i \in \{j_1, \ldots, j_k\}}$.
- **Description of g'_i.** On input $\overline{y}^{(j)}$ and $\{\widetilde{y}^{(1)}_i \ldots, \widetilde{y}^{(s)}_i\}_{i \in \{j_1, \ldots, j_k\}}$, g'_i does the following.
  1. Set $y^{(j)}_{j^*} = \overline{y}^{(j)} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell)$.
  2. For every $\ell \notin \{T_{j_1}, \ldots, T_{j_k}\}$, run $f_{T_\ell}(x_{T_\ell})$ to obtain $\widetilde{x}_\ell$.
  3. Parse every such $\widetilde{x}_\ell$ as $\widetilde{x}^{(\ell)}, \widetilde{y}^{(1)}_\ell \ldots, \widetilde{y}^{(s)}_\ell$.
  4. Using the above obtained values and the output of the leakage function, compute $\widetilde{y}^{(\ell)} = \widetilde{y}^{(\ell)}_1 \oplus \ldots \oplus \widetilde{y}^{(\ell)}_s$ for every $\ell \in [s]$.
  5. If $i \in \{j_1, \ldots, j_k\}$, output $\widetilde{y}^{(i)}$.
  6. Else, compute $z_i = \mathsf{2SLNMExt}(\widetilde{x}^{(i)}, \widetilde{y}^{(i)})$. Let y^*_i be the value such that $\mathsf{2NMExt}(x^*, y^*_i) = z_i$. Output y^*_i.

Analysis. Notice that the only difference between Hyb_j and Hyb_{j-1} is that in Hyb_{j-1}, b_j is set to $\mathsf{2SLNMExt}(x^{(i)}, y^{(j)})$ whereas in Hyb_j, this value is sampled from $\{0,1\}^m$ uniformly. Let us fix all the values in $x_1, \ldots, x_s$ except $x^{(j)}$ and $y^{(j)}_{j^*}$ in two distributions. Let us collectively call the fixed values as τ. Note that with probability at least $1 - 2\varepsilon$ over sampling τ, the random variables $X^{(j)}$ and $Y^{(j)}_{j^*} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell)$ conditioned on fixing τ are independent and have min-entropy $n_1 - \Delta$ and $n_2 - \Delta$ respectively (follows from Lemma 3.2). To show that, Hyb_{j-1} and Hyb_j are $O(\varepsilon + s2^{-m})$-close to each other, we consider the following algorithm $\mathcal{B}$ that on input the fixed values τ and $(z, z', a_1, \ldots, a_s) \in \{0,1\}^m \times \{0,1\}^m \times (\{0,1\}^m \cup \{\mathsf{same}^*\})^s$ that does the following.

[6] We now argue that since $\varepsilon < 1/2^m$ such an x^* always exists. Assume for the sake of contradiction that for every $x \in \{0,1\}^{n_1}$, there exists some a_x such that for every $y \in \{0,1\}^{n_2}$, $\mathsf{2SLNMExt}(x, y) \neq a_x$. Then, for an uniform choice of X and Y, $\Pr[\mathsf{2SLNMExt}(X, Y) = a_X] = 0$ and hence the statistical distance between $\mathsf{2SLNMExt}(X, Y)$ and U_m is at least $1/2^m$.

1. Uniformly sample $b_1, \ldots, b_{j-1}$ from $\{0,1\}^m$.
2. For $i > j$, set $b_i = \mathsf{2SLNMExt}(x^{(i)}, y^{(i)})$.
3. If for any $i \in [s]$, $a_i = \mathsf{same}^*$, replace a_i with z'.
4. If for any $i \in [s]$, $a_i = z'$, replace a_i with z.
5. If for any $i \in [s]$, $a_i = \mathsf{2SLNMExt}(x^{(k)}, y^{(k)})$ for some $k \in [s] \setminus \{j\}$ (obtained from τ), then replace a_i with b_k.
6. Output $(b_1 \oplus \ldots \oplus b_s) \circ (a_1 \oplus \ldots \oplus a_s)$.

We now consider a sequence of distributions $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2$ that serve as inputs to $\mathcal{B}$.

$\underline{\mathcal{D}_0}$: Let τ denote the set of all fixed values. Sample $(x,y),(x',y') \sim (X^{(j)}, Y^{(j)}_{j^*} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell))|\tau$ independently. Output $\mathsf{2SLNMExt}(x,y) \circ \mathsf{2SLNMExt}(x',y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x,y), \{\mathsf{2SLNMExt}(f'_i(x), g'_i(y, h(x)))\}_{i\in[s]})$

$\underline{\mathcal{D}_1}$: Let τ denote the set of all fixed values. Sample $(x,y),(x',y') \sim (X^{(j)}, Y^{(j)}_{j^*} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell))|\tau$ independently. Output $\mathsf{2SLNMExt}(x,y) \circ \mathsf{2SLNMExt}(x',y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x',y'), \{\mathsf{2SLNMExt}(f'_i(x'), g'_i(y', h(x')))\}_{i\in[s]})$.

$\underline{\mathcal{D}_2}$: Let τ denote the set of all fixed values. Sample $(x',y') \sim (X^{(j)}, Y^{(j)}_{j^*} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell))|\tau$ independently. Output $U_m \circ \mathsf{2SLNMExt}(x',y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x',y'), \{\mathsf{2SLNMExt}(f'_i(x'), g'_i(y', h(x')))\}_{i\in[s]})$.

Notice that if $\mathcal{B}$ was given a sample from $\mathcal{D}_0$ then the output of $\mathcal{B}$ is $O(\varepsilon + s2^{-m})$-close to Hyb_{j-1} since $\mathsf{2SLNMExt}(x'^{(j)}, y'^{(j)})$ is ε-close to U_m and the probability in that any $a_j = U_m$ is at most 2^{-m}.
Since, the random variables $X^{(j)}$ and $Y^{(j)}_{j^*} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell)$ conditioned on fixing τ are independent and have min-entropy $n_1 - \Delta$ and $n_2 - \Delta$ with probability $1 - 2\varepsilon$ (from Lemma 3.2), it follows from Lemma 5.9 that $\mathcal{B}(\mathcal{D}_0)$ and $\mathcal{B}(\mathcal{D}_1)$ defined above are $O(\varepsilon + s2^{-m})$-close.

Further, notice that if $\mathcal{B}$ was given a sample from $\mathcal{D}_2$, its output is identical to Hyb_j. Now, since $(X^{(j)}, Y^{(j)} \oplus (\oplus_{\ell \neq j^*} y^{(j)}_\ell))|\tau$ is a source with min-entropy $(n_1 - \Delta, n_2 - \Delta)$ (with probability at least $1 - 2\varepsilon$), the distribution $\mathsf{2SLNMExt}(x,y)$ in $\mathcal{D}_1$ is ε close to U_m and thus $\mathcal{D}_1$ is ε close to $\mathcal{D}_2$. Therefore, $\mathcal{B}(\mathcal{D}_1)$ is $O(\varepsilon)$ close to $\mathcal{B}(\mathcal{D}_2)$. This completes the proof of the claim. ■

We are now ready to define our simulator D_f.

$\underline{D_f}$: The input to the simulator is $b_1, \ldots, b_s$ where each b_i is m-bits long and chosen uniformly at random. The output of this hybrid is generated as follows.

1. Sample $(x_1, \ldots, x_s) \sim (X_1, \ldots, X_s)$
2. Run $f(x_1, \ldots, x_n)$ to obtain $(\widetilde{x}_1, \ldots, \widetilde{x}_s)$.
3. Let $(a_1, \ldots, a_s) := \mathsf{split}(\widetilde{x}_1, \ldots, \widetilde{x}_s)$.

4. If for any $i \in [s]$, $a_i = \mathsf{2SLNMExt}(x^{(k)}, y^{(k)})$ where $y^{(k)} = y_1^{(k)} \oplus y_2^{(k)} \ldots \oplus y_s^{(k)}$ for some $k \in [s]$, then replace a_i with b_k.

5. If $(a_1, \ldots, a_s)$ is a permutation of $(b_1, \ldots, b_s)$ output same^*.

6. Else, output $(a_1 \oplus \ldots \oplus a_s)$.

It now follows that since D_f uses at most $s-1$ of the values $b_1, \ldots, b_s$, we get

$$\mathsf{Hyb}_s \equiv (b_1 \oplus \ldots \oplus b_s) \circ \mathsf{copy}(D_f, b_1 \oplus \ldots \oplus b_s) \equiv (U_m, \mathsf{copy}(D_f, U_m)).$$

This completes the proof of the theorem. ■

6.3 Instantiation

We now instantiate construction 3 with the strong leakage-resilient non-malleable extractors from section 5.3.

From Theorem **??** (see also Corollary 5.5), by setting $p(n_2) = s^2 n_2$, there exists n_0 such that for any $n_2 > n_0$, we get could a p-strong leakage-resilient $(2, s)$-non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ with min-entropy $(n_1 - \Delta, n_2 - \Delta)$ and error ε, where $n_1 = 4n_2 + 2p(n_2)$, $m = n_2^{\Omega(1)}$, $\Delta = \gamma n_2$, $\varepsilon = 2^{-n_2^{\Omega(1)}}$ for some constant γ. We can assume $m < \log 1/\varepsilon$ since we can cut any number of bits from the output of $\mathsf{2SLNMExt}$ while the error bound ε still holds. We can also let $\Delta > 2 \log 1/\varepsilon$ by enlarging ε.

Let $n = (2s^2 + s + 4)n_2$ and $\gamma' = \gamma/(2s^2 + s + 4)$. From theorem 6.2, we get a s-source, non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy $(1-\gamma')n$ and error $2^{-n^{\Omega(1)}}$ with output length $n^{\Omega(1)}$. We summarize the instantiation with the following corollary.

Corollary 6.5 *For any $s \geq 2$, there exists a constant n_0 and γ such that for any $n > n_0$, there exists a s-source, non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy $(1-\gamma)n$ and error $2^{-n^{\Omega(1)}}$ with output length $n^{\Omega(1)}$.*

6.4 Efficient Pre-image sampleability

We now show that if the underlying $\mathsf{2SLNMExt}$ is efficiently pre-image sampeable, then our construction of multi-source non-malleable extractor is also efficiently pre-image sampleable.

Pre-image Sampling Procedure Given any $\mathsf{msg} \in \{0,1\}^m$, the pre-image sampling procedure does the following:

1. Sample $\mathsf{msg}_1, \ldots, \mathsf{msg}_{s-1}$ uniformly from $\{0,1\}^m$.

2. Set $\mathsf{msg}_s = \mathsf{msg} \oplus \mathsf{msg}_1 \oplus \mathsf{msg}_2 \ldots \oplus \mathsf{msg}_{s-1}$.

3. Sample $(x^{(i)}, y^{(i)}) \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{msg}_i)$ for all $1 \leq i \leq s$.

4. Sample $y_1^{(i)}, \ldots, y_{s-1}^{(i)}$ from $\{0,1\}^{n_2}$ for all $1 \leq i \leq s$.

5. Set $y_s^{(i)} = y^{(i)} \oplus y_1^{(i)} \oplus y_2^{(i)} \ldots \oplus y_{s-1}^{(i)}$ for all $1 \leq i \leq s$.

6. Output $(x_1, y_1^{(1)}, \ldots, y_1^{(s)}) \circ (x_2, y_2^{(1)}, \ldots, y_2^{(s)}) \ldots \circ (x_s, y_s^{(1)}, \ldots, y_s^{(s)})$.

It is clear that the above procedure give an uniform sample from $\mathsf{MNMExt}^{-1}(\mathsf{msg})$, and if the step 3 can be done efficiently, which means the underlying $\mathsf{2SLNMExt}$ is efficiently pre-image sampleable, then the whole sampling procedure is also efficient.

6.4.1 Instantiation

We now instantiate $\mathsf{2SLNMExt}$ from section 5.4.1. Recall that this extractor has efficient pre-image sampleability.

From Corollary 5.7, by setting $p(n_2) = s^2 n_2$, we get could a p-strong leakage-resilient $(2, s)$-non-malleable extractor $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ with min-entropy (n_1, n_2) and error ε, where $n_1 = 4n_2 + p(n_2)$, $m = n_2^{\Omega(1)}$, $\varepsilon = 2^{-n_2^{\Omega(1)}}$ and $s < n_2^{\gamma}$ for some constant γ. We assume $m < \log 1/\varepsilon$ as above.

Let $n = (s^2 + s + 4)n_2$, which implies $n_2 = n^{\Omega(1)}$. Let $\gamma' > 0$ be constant such that $\gamma' < \frac{\gamma}{2\gamma+1}$. From theorem 6.2, for any $s \leq n^{\gamma'}$, we get a s-source, non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy n and error $2^{-n^{\Omega(1)}}$ with output length $n^{\Omega(1)}$, which is also efficiently pre-image sampleable.

Corollary 6.6 *For any $s \geq 2$ and $n \geq s^{1/\gamma'}$, there exists an efficiently pre-image sampleable s-source, non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy n and error $2^{-n^{\Omega(1)}}$ with output length $n^{\Omega(1)}$.*

7 Multi-Split-State Non-malleable Codes

In this section, we will define multi-split-state non-malleable codes and show how to construct the multi-split-state non-malleable codes against a certain tampering function families, such as $\mathcal{F}_{\mathsf{dis}}$ or $\mathcal{F}_{\mathsf{cover-free}}$, from a multi-source non-malleable extractor against the same tampering function families. The construction follows the same paradigm as in [CG14].

7.1 Definition

In this subsection, we define multi-split-state non-malleable codes, which is similar to multi-source non-malleable extractor. The codeword is split into s states, where the tampering function for each state takes some but not all states as input and outputs the tampered version of that state.

Definition 7.1 (Multi-Split-State Non-Malleable Codes) *A coding scheme $\mathsf{MNMEnc} : \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n$, $\mathsf{MNMDec} : \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n \to \{0,1\}^m$ is a s-split-state non-malleable code with error ε against a family of tampering functions $\mathcal{F}$ if for every $f \in \mathcal{F}$, there exists a random variable D_f on $\{0,1\}^m \cup \{\mathsf{same}\}$t such that for all messages $\mathsf{msg} \in \{0,1\}^m$, it holds that*

$$|\mathsf{MNMDec}(f(X_1, \ldots, X_s)) - \mathsf{copy}(D_f, \mathsf{msg})| \leq \varepsilon$$

where $X_1, \ldots, X_t = \mathsf{MNMEnc}(\mathsf{msg})$.

Note the tampering function families $\mathcal{F}_{\mathsf{dis}}$ and $\mathcal{F}_{\mathsf{cover-free}}$ defined in 6.1 are also the tampering function families for multi-split-state codes. Therefore, we could use them to define s-split-state non-malleable codes against $\mathcal{F}_{\mathsf{dis}}$ or $\mathcal{F}_{\mathsf{cover-free}}$.

7.2 Construction

We now recall the result of [CG14] and generalize it to s-independent sources.

Theorem 7.2 ([CG14]) *Let* $\mathsf{MNMExt} : \{0,1\}^n \times \{0,1\}^n \cdots \times \{0,1\}^n \to \{0,1\}^m$ *be a* s*-source non-malleable extractor against a tampering function family* $\mathcal{F}$ *with error* ε*. Construct* (MNMEnc, MNMDec) *as following:*

- $\mathsf{MNMEnc} : \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n$ *such that* $\mathsf{MNMEnc}(\mathsf{msg})$ *outputs a uniform sample from* $\mathsf{MNMExt}^{-1}(\mathsf{msg})$*.*
- $\mathsf{MNMDec} : \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n \to \{0,1\}^m$ *such that* $\mathsf{MNMDec}(x_1, \ldots, x_s)$ *outputs* $\mathsf{MNMExt}(x_1, \ldots, x_s)$*.*

Then, the above construction is a s*-split-state non-malleable against* $\mathcal{F}$ *with error* $\varepsilon(2^{(m+1)} + 1)$*.*

Proof It is clear that for any $\mathsf{msg} \in \{0,1\}^m$, $\mathsf{MNMDec}(\mathsf{MNMEnc}(s)) = s$ with probability 1, so $(\mathsf{MNMEnc}, \mathsf{MNMDec})$ is a valid coding scheme. To prove its non-malleability, let us fix the tampering functions $f \in \mathcal{F}$. Since the MNMExt is a s-source non-malleable extractor against $\mathcal{F}$ with error ε, by definition 6.1, there exists a distribution D_f such that for independent uniform random variables $X_1, \ldots, X_s \in \{0,1\}^n$, it holds that

$$\begin{aligned}|&\mathsf{MNMExt}(X_1, \ldots, X_s), \mathsf{MNMExt}(f(X_1, \ldots, X_s)) \\ &-\mathsf{MNMExt}(X_1, \ldots, X_s), \mathsf{copy}(D_f, \mathsf{MNMExt}(X_1, \ldots, X_s))| < 2\varepsilon.\end{aligned}$$

For $\mathsf{msg} \sim U_m$, we note that $\mathsf{MNMExt}(x_1, \ldots, x_s)$ where $(x_1, \ldots, x_s) \leftarrow \mathsf{SEnc}(M)$ is ε-close to $\mathsf{MNMExt}(x'_1, \ldots, x'_s)$ where $x'_1, \ldots, x'_s \sim U_n$. Thus, for any $\mathsf{msg} \in \{0,1\}^m$, by Lemma 3.7, it holds that

$$|\mathsf{MNMExt}(f(X_1, \ldots, X_s)) - \mathsf{copy}(D_f, \mathsf{msg})| < \varepsilon(2^{m+1} + 1),$$

where $X_1, \ldots, X_s \leftarrow \mathsf{MNMExt}^{-1}(\mathsf{msg})$. Therefore, by the definition of MNMEnc and MNMDec we have

$$|\mathsf{MNMDec}(f(X_1, \ldots, X_s)) - \mathsf{copy}(D_f, \mathsf{msg})| < \varepsilon(2^{m+1} + 1),$$

where $(X_1, \ldots, X_s) = \mathsf{MNMEnc}(\mathsf{msg})$. This completes the proof of the theorem. ■

From Corollary 6.6, there exists a constant $\gamma > 0$ such that for any $s \geq 2$ and $n \geq s^\gamma$, there exists an efficiently pre-image sampleable s-source non-malleable extractor MNMExt against $\mathcal{F}_{\mathsf{cover-free}}$ with error $\varepsilon = 2^{-n^{\Omega(1)}}$ and output length $m = n^{\Omega(1)}$. We can assume $m < 1/2 \log(1/\varepsilon)$ since we can cut any number of bits from the output of MNMExt while the error bound ε still holds. Therefore, by the above theorem, we have the following corollary.

Corollary 7.3 *For any* $s \geq 2$ *and for all* $m \in \mathbb{N}$*, there exists an efficient construction of* s*-split-state non-malleable code for messages of length* m *that is secure against cover-free tampering with error* $2^{-m^{\Omega(1)}}$ *and codeword length* $(m + s)^{O(1)}$*.*

8 Non-Malleable Secret Sharing

In this section, we give a construction of threshold non-malleable secret sharing schemes with security against t-cover-free tampering.

8.1 Definition

We first give the definition of a sharing function, then define a threshold secret sharing scheme and finally give the definition of a threshold non-malleable secret sharing. These three definitions are taken verbatim from [GK18a].

Definition 8.1 (Sharing Function) *Let $[n] = \{1, 2, \ldots, n\}$ be a set of identities of n parties. Let $\mathcal{M}$ be the domain of secrets. A sharing function Share is a randomized mapping from $\mathcal{M}$ to $\mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_n$, where $\mathcal{S}_i$ is called the domain of shares of party with identity i. A dealer distributes a secret $m \in \mathcal{M}$ by computing the vector $\mathsf{Share}(m) = (\mathsf{S}_1, \ldots, \mathsf{S}_n)$, and privately communicating each share S_i to the party i. For a set $T \subseteq [n]$, we denote $\mathsf{Share}(m)_T$ to be a restriction of $\mathsf{Share}(m)$ to its T entries.*

Definition 8.2 ($(t, n, \varepsilon_c, \varepsilon_s)$-Secret Sharing Scheme) *Let $\mathcal{M}$ be a finite set of secrets, where $|\mathcal{M}| \geq 2$. Let $[n] = \{1, 2, \ldots, n\}$ be a set of identities (indices) of n parties. A sharing function Share with domain of secrets $\mathcal{M}$ is a $(t, n, \varepsilon_c, \varepsilon_s)$-secret sharing scheme if the following two properties hold :*

- ***Correctness:*** *The secret can be reconstructed by any t-out-of-n parties. That is, for any set $T \subseteq [n]$ such that $|T| \geq t$, there exists a deterministic reconstruction function $\mathsf{Rec} : \otimes_{i \in T} \mathcal{S}_i \to \mathcal{M}$ such that for every $m \in \mathcal{M}$,*

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

  where the probability is over the randomness of the Share function. We will slightly abuse the notation and denote Rec as the reconstruction procedure that takes in T and $\mathsf{Share}(m)_T$ where T is of size at least t and outputs the secret.

- ***Statistical Privacy:*** *Any collusion of less than t parties should have "almost" no information about the underlying secret. More formally, for any unauthorized set $U \subseteq [n]$ such that $|U| < t$, and for every pair of secrets $m_0, m_1 \in M$, for any distinguisher D with output in $\{0, 1\}$, the following holds :*

  $$|\Pr[D(\mathsf{Share}(m_0)_U) = 1] - \Pr[D(\mathsf{Share}(m_1)_U) = 1]| \leq \varepsilon_s$$

We define the rate of the secret sharing scheme as

$$\lim_{|m| \to \infty} \frac{|m|}{\max_{i \in [n]} |\mathsf{Share}(m)_i|}$$

Definition 8.3 (Threshold Non-Malleable Secret Sharing [GK18a]) *Let $(\mathsf{Share}, \mathsf{Rec})$ be a $(t, n, \varepsilon_c, \varepsilon_s)$-secret sharing scheme for message space $\mathcal{M}$. Let $\mathcal{F}$ be some family of tampering functions. For each $f \in \mathcal{F}$, $m \in \mathcal{M}$ and authorized set $T \subseteq [n]$ containing t indices, define the tampered*

distribution $\mathsf{Tamper}_m^{f,T}$ *as* $\mathsf{Rec}(f(\mathsf{Share}(m))_T)$ *where the randomness is over the sharing function* Share. *We say that the* $(t, n, \varepsilon_c, \varepsilon_s)$*-secret sharing scheme,* $(\mathsf{Share}, \mathsf{Rec})$ *is* ε'*-non-malleable w.r.t.* $\mathcal{F}$ *if for each* $f \in \mathcal{F}$ *and any authorized set* T *consisting of* t *indices, there exists a distribution* $D^{f,T}$ *over* $\mathcal{M} \cup \{\mathsf{same}\}$ *such that for every* $m \in \mathcal{M}$*:*

$$|\mathsf{Tamper}_m^{f,T} - \mathsf{copy}(D^{f,T}, m)| \leq \varepsilon'$$

where copy *is defined by* $\mathsf{copy}(x, y) = \begin{cases} x & \text{if } x \neq \mathsf{same} \\ y & \text{if } x = \mathsf{same} \end{cases}$.

Tampering Function Family. Like in the case of multi-source non-malleable extractors, we will be interested in constructing non-malleable secret sharing against the following class of tampering functions. Let $T_1, \ldots, T_n \subset [n]$. The tampering family $\mathcal{F}_{T_1,\ldots,T_n}$ consists of the set of all functions $f = (f_{T_1}, \ldots, f_{T_n})$ such that on input $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n)$ (where $\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n$ are the n shares output by the Share algorithm), f outputs $(\widetilde{\mathsf{Sh}}_1, \ldots, \widetilde{\mathsf{Sh}}_n)$ where for every $i \in [n]$, $f_{T_i}(\{\mathsf{Sh}_j\}_{j \in T_i}) = \widetilde{\mathsf{Sh}}_i$. In other words, $\widetilde{\mathsf{Sh}}_i$ is generated by applying f_{T_i} on the set of shares $\{\mathsf{Sh}_j\}_{j \in T_i}$. Depending on the properties required from the sets $\{T_1, \ldots, T_n\}$, we get two interesting classes of tampering functions.

- t**-disjoint Tampering Family.** The disjoint tampering family $\mathcal{F}_{t-\mathsf{dis}}$ is the set of all $\mathcal{F}_{T_1,\ldots,T_n}$ for every possible $T_1, \ldots, T_n$ such that each T_i is non-empty, $|T_i| \leq t - 1$, and if $x \in T_i, T_j$ then $T_i = T_j$.
- t**-cover-free Tampering Family.** For every $i \in [n]$, let us define $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_n$ to be the union of all the sets T_j where $i \in T_j$. The t-cover-free tampering family $\mathcal{F}_{t-\mathsf{cover-free}}$ is the set of all $\mathcal{F}_{T_1,\ldots,T_n}$ for all possible $T_1, \ldots, T_n \subset [n]$ such that for every $i \in [n]$, the size of $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_n$ is at most $t - 1$.

Observe that $\mathcal{F}_{t-dis} \subset \mathcal{F}_{t-\mathsf{cover-free}}$ and hence in the rest of the paper, we will focus on constructing non-malleable extractors that are secure against $\mathcal{F}_{t-\mathsf{cover-free}}$.

8.2 Construction

In this subsection, we will give a construction of t-out-of-n non-malleable secret sharing scheme that is secure against $\mathcal{F}_{t-\mathsf{cover-free}}$.

Building Blocks. In our construction, we will use the following building blocks.

- Let $(\mathsf{Share}, \mathsf{Rec})$ be a t-out-of-n Shamir secret sharing scheme. The length of each share is same as the length of the message.
- Define the polynomial $p(\cdot)$ as $p(x) = xn^2$. Let $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^{3m}$ be a p-strong leakage-resilient, $(2, t)$-non-malleable extractor with efficient pre-image sampleability and error ε.
- We set $\varepsilon < 1/2^{3m}$.[7]

[7]Similar to the construction of multi-source non-malleable extractor in section 6.2, we need this condition since in proof, we need the fact that there exists L^* such that for every $s \in \{0,1\}^{3m}$ there exists an R_s such that $\mathsf{2SLNMExt}(L^*, R_s) = s$.

Construction 4. We give the description of $(\mathsf{NMShare}, \mathsf{NMRec})$.

- $\mathsf{NMShare}(s)$: On input a message $s \in \{0,1\}^m$, do:
  1. $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n) \leftarrow \mathsf{Share}(s)$.
  2. For each $i \in [n]$, compute $(\mathsf{L}^{(i)}, \mathsf{R}^{(i)}) \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{Sh}_i \circ U_{2m})$.
  3. For each $i \in [n]$, $(\mathsf{R}_1^{(i)}, \ldots, \mathsf{R}_n^{(i)}) \leftarrow \mathsf{Share}(\mathsf{R}^{(i)})$.
  4. Set $\mathsf{share}_i = (\mathsf{L}^{(i)}, \mathsf{R}_i^{(1)}, \ldots, \mathsf{R}_i^{(n)})$.
  5. Output $(\mathsf{share}_1, \ldots, \mathsf{share}_n)$.
- $\mathsf{NMRec}(\mathsf{share}_{i_1}, \ldots, \mathsf{share}_{i_\ell})$: On input $(\mathsf{share}_{i_1}, \ldots, \mathsf{share}_{i_t})$ for distinct $i_1, \ldots, i_t$:
  1. For each $i \in \{i_1, \ldots, i_t\}$,
     (a) Parse share_i as $(\mathsf{L}^{(i)}, \mathsf{R}_i^{(1)}, \ldots, \mathsf{R}_i^{(n)})$.
     (b) Compute $\mathsf{R}^{(i)} := \mathsf{Rec}(\mathsf{R}_{i_1}^{(i)}, \ldots, \mathsf{R}_{i_t}^{(i)})$.
     (c) Set $\mathsf{Sh}_i := \mathsf{2SLNMExt}(\mathsf{L}^{(i)}, \mathsf{R}^{(i)})_{[m]}$.
  2. Output $s := \mathsf{Rec}(\mathsf{Sh}_{i_1}, \ldots, \mathsf{Sh}_{i_t})$.

Theorem 8.4 *For any $t \geq 2$, $(\mathsf{NMShare}, \mathsf{NMRec})$ described above is a $(t, n, 0, 0)$ secret sharing scheme that is $O(n(\varepsilon \cdot 2^{3m} + t2^{-m}))$-non-malleable against $\mathcal{F}_{t-\mathsf{cover}-\mathsf{free}}$.*

Proof Correctness is easy to observe and we start with proving privacy.

Perfect Privacy. Let $A = \{j_1, \ldots, j_{t-1}\}$ be a set of size $t-1$. It is easy to observe from the perfect privacy of Shamir secret sharing that $(\mathsf{R}^{(1)}, \ldots, \mathsf{R}^{(n)})$ is perfectly hidden given share_A. It now follows that since share_A is independent of $\{\mathsf{L}^{(i)}\}_{i \in [n] \setminus A}$, share_A provides no information about $\mathsf{Sh}_{[n]\setminus A}$. Hence, it follows from the privacy of Shamir secret sharing that share_A perfectly hides m.

Non-Malleability. Let us fix a tampering function $f = (f_{T_1}, \ldots, f_{T_n}) \in \mathcal{F}_{t-\mathsf{cover}-\mathsf{free}}$. Recall that by definition, for every $i \in [n]$, the size of $\mathsf{Cover}(i)$ w.r.t. $T_1, \ldots, T_n$ is at most $t-1$. To prove the non-malleability of $(\mathsf{NMShare}, \mathsf{NMRec})$, we need to show that for every $T \subseteq [n]$ of size t, the existence of a distribution $D^{f,T}$ such that for every $s \in \{0,1\}^m$:

$$|\mathsf{Tamper}_s^{f,T} - \mathsf{copy}(D^{f,T}, s)| \leq \varepsilon' \tag{8.1}$$

where $\varepsilon' = O(n(\varepsilon \cdot 2^{3m} + t2^{-m}))$.

Let $T = \{i_1, \ldots, i_t\}$ and let $T_{i_1} \cup T_{i_2} \cup \ldots \cup T_{i_t} = \{j_1, \ldots, j_k\}$ where $j_1 < j_2 < \ldots < j_k$. We only consider the case where $k \geq t$ as otherwise, it follows from the perfect privacy that $\mathsf{Tamper}_s^{f,T}$ is independent of s. As before, we define the function split as follows.

Definition 8.5 *The function* split *takes as input* $\mathsf{share}_{i_1}, \ldots, \mathsf{share}_{i_t}$ *and does the following:*

1. *For each $i \in \{i_1, \ldots, i_t\}$,*
   (a) Parse share_i *as* $(\mathsf{L}^{(i)}, \mathsf{R}_i^{(1)}, \ldots, \mathsf{R}_i^{(n)})$.

(b) *Compute* $\mathsf{R}^{(i)} := \mathsf{Rec}(\mathsf{R}^{(i)}_{i_1}, \ldots, \mathsf{R}^{(i)}_{i_t})$.

(c) *Set* $\mathsf{Sh}_i = \mathsf{2SLNMExt}(\mathsf{L}^{(i)}, \mathsf{R}^{(i)})_{[m]}$.

2. *Output* $(\mathsf{Sh}_{i_1}, \ldots, \mathsf{Sh}_{i_t})$.

For every $\ell \in [n+1]$, we define a hybrid Hyb_ℓ as follows.

$\underline{\mathsf{Hyb}_\ell}$.

1. Compute $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n) \leftarrow \mathsf{Share}(s)$.
2. For each $i < \ell$, set $\mathsf{Sh}'_i \leftarrow \{0,1\}^m$. For $i \geq \ell$, set $\mathsf{Sh}'_i = \mathsf{Sh}_i$.
3. For each $i \in [n]$, compute $(\mathsf{L}^{(i)}, \mathsf{R}^{(i)}) \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{Sh}'_i \circ U_{2m})$.
4. For each $i \in [n]$, $(\mathsf{R}^{(i)}_1, \ldots, \mathsf{R}^{(i)}_n) \leftarrow \mathsf{Share}(\mathsf{R}^{(i)})$.
5. Set $\mathsf{share}_i = (\mathsf{L}^{(i)}, \mathsf{R}^{(1)}_i, \ldots, \mathsf{R}^{(n)}_i)$.
6. For each $i \in \{i_1, \ldots, i_t\}$, compute $f_{T_i}(\mathsf{share}_{T_i}) = \widetilde{\mathsf{share}}_i$.
7. Compute $(\widetilde{\mathsf{Sh}_{i_1}}, \ldots, \widetilde{\mathsf{Sh}_{i_t}}) := \mathsf{split}(\widetilde{\mathsf{share}}_{i_1}, \ldots, \widetilde{\mathsf{share}}_{i_t})$.
8. If for any $j \in \{i_1, \ldots, i_t\}$, $\widetilde{\mathsf{Sh}}_j = \mathsf{Sh}'_i$ for some $i \in [n]$, reset $\widetilde{\mathsf{Sh}}_j$ with Sh_i.
9. Output $\mathsf{Rec}(\widetilde{\mathsf{Sh}_{i_1}}, \ldots, \widetilde{\mathsf{Sh}_{i_t}})$.

Observe that Hyb_1 is identical to $\mathsf{Tamper}^{f,T}_s$. We now show that from the p-strong leakage resilience property of $\mathsf{2SLNMExt}$, $\mathsf{Hyb}_\ell \approx_{O(\varepsilon \cdot 2^{3m} + t \cdot 2^{-m})} \mathsf{Hyb}_{\ell+1}$ for every $\ell \in [n]$.

Claim 8.6 *For every* $\ell \in [n]$, $\mathsf{Hyb}_\ell \approx_{O(\varepsilon \cdot 2^{3m} + t \cdot 2^{-m})} \mathsf{Hyb}_{\ell+1}$.

Proof Notice that the only difference between Hyb_ℓ and $\mathsf{Hyb}_{\ell+1}$ is that in $\mathsf{Hyb}_{\ell+1}$, Sh'_ℓ is chosen uniformly at random independent of all other values, whereas in Hyb_ℓ, it is set to Sh_ℓ. Observe that if $\ell \notin \{j_1, \ldots, j_k\}$ then it follows from Lemma 5.10) that $\mathsf{Hyb}_\ell \approx_{O(\varepsilon \cdot 2^{3m})} \mathsf{Hyb}_{\ell+1}$. We now consider the case when $\ell \in \{j_1, \ldots, j_k\}$.

Recall the definition of Cover. Notice that by t-cover-freeness property of $T_1, \ldots, T_n$, $|\mathsf{Cover}(\ell) \cap \{j_1, \ldots, j_k\}| \leq |\mathsf{Cover}(\ell)| \leq t-1$. We will now design tampering functions $(f'_{i_1}, \ldots, f'_{i_t}), (g'_{i_1}, \ldots, g'_{i_t})$ and a leakage function h' against $\mathsf{2SLNMExt}$. Let us denote $T_{\ell_1}, \ldots, T_{\ell_c}$ to be all the sets among $T_{i_1}, \ldots, T_{i_t}$ that contain ℓ.

Description of $(f'_{i_1}, \ldots, f'_{i_t}), (g'_{i_1}, \ldots, g'_{i_t}), h'$**.**

- **Shared Randomness.** For every $i \in [n] \setminus \{\ell\}$, sample $\mathsf{L}^{(i)}, \mathsf{R}^{(i)}$ according to Hyb_ℓ. Sample the Shamir shares of $\mathsf{R}^{(i)}$ for every $i \in [n] \setminus \{\ell\}$ as per Hyb_ℓ. For every $i \in \mathsf{Cover}(\ell)$, sample $\mathsf{R}^{(\ell)}_i$ as an uniformly chosen element. Let L^* be a special string such that for every $s' \in \{0,1\}^{3m}$, there exists an $\mathsf{R}_{s'}$ such that $\mathsf{2SLNMExt}(\mathsf{L}^*, \mathsf{R}_{s'}) = s'$.
- **Description of** f'_i**.** On input $\mathsf{L}^{(\ell)}$, $\overrightarrow{f}'_i$ does the following.

1. If $i \notin \{\ell_1, \ldots, \ell_c\}$, output L^*.
2. Else, set $\mathsf{share}_\ell = (\mathsf{L}^{(\ell)}, \mathsf{R}_\ell^{(1)}, \ldots, \mathsf{R}_\ell^{(n)})$.
3. Apply $f_{T_i}(\mathsf{share}_{T_i})$ to obtain $\widetilde{\mathsf{share}}_i$.
4. Parse $\widetilde{\mathsf{share}}_i$ as $\widetilde{\mathsf{L}}^{(i)}, \widetilde{\mathsf{R}}_i^{(1)} \ldots, \widetilde{\mathsf{R}}_i^{(n)}$ and output $\widetilde{\mathsf{L}}^{(i)}$.

- **Description of h'.** On input $\mathsf{L}^{(\ell)}$, h' does the following.
  1. Set $\mathsf{share}_\ell = (\mathsf{L}^{(\ell)}, \mathsf{R}_\ell^{(1)}, \ldots, \mathsf{R}_\ell^{(n)})$.
  2. For each $i \in \{\ell_1, \ldots, \ell_c\}$, apply $f_{T_i}(\mathsf{share}_{T_i})$ to obtain $\widetilde{\mathsf{share}}_i$.
  3. Parse $\widetilde{\mathsf{share}}_i$ as $\widetilde{\mathsf{L}}^{(i)}, \widetilde{\mathsf{R}}_i^{(1)} \ldots, \widetilde{\mathsf{R}}_i^{(n)}$.
  4. Output $\{\widetilde{\mathsf{R}}_i^{(1)} \ldots, \widetilde{\mathsf{R}}_i^{(n)}\}_{i \in \{\ell_1, \ldots, \ell_c\}}$.

- **Description of g'_i.** On input $\mathsf{R}^{(\ell)}$ and the output of the leakage function $\{\widetilde{\mathsf{R}}_i^{(1)} \ldots, \widetilde{\mathsf{R}}_i^{(n)}\}_{i \in \{\ell_1, \ldots, \ell_c\}}$, g'_i does the following.
  1. Sample the Shamir shares $\mathsf{R}_1^{(\ell)}, \ldots, \mathsf{R}_n^{(\ell)}$ as shares of $\mathsf{R}^{(\ell)}$ such that these are consistent with the fixed values as part of shared randomness.
  2. For every $i \notin \{\ell_1, \ldots, \ell_c\}$, run $f_{T_i}(\mathsf{share}_{T_i})$ to obtain $\widetilde{\mathsf{share}}_i$.
  3. Parse every such $\widetilde{\mathsf{share}}_i$ as $\widetilde{\mathsf{L}}^{(i)}, \widetilde{\mathsf{R}}_i^{(1)} \ldots, \widetilde{\mathsf{R}}_i^{(n)}$.
  4. Using the above obtained values and the output of the leakage function, compute for every $j \in \{i_1, \ldots, i_t\}$, $\widetilde{\mathsf{R}}^{(j)} = \mathsf{Rec}(\widetilde{\mathsf{R}}_{i_1}^{(j)}, \ldots, \widetilde{\mathsf{R}}_{i_t}^{(j)})$.
  5. If $i \in \{\ell_1, \ldots, \ell_c\}$, output $\widetilde{\mathsf{R}}^{(i)}$.
  6. Else, compute $z_i = \mathsf{2SLNMExt}(\widetilde{\mathsf{L}}^{(i)}, \widetilde{\mathsf{R}}^{(i)})$. Let R_i^* be the value such that $\mathsf{2SLNMExt}(\mathsf{L}^*, \mathsf{R}_i^*) = z_i$. Output R_i^*.

Analysis. Observe that $(f'_{i_1}, \ldots, f'_{i_t}), (g'_{i_1}, \ldots, g'_{i_t}), h'$ are valid tampering functions against the p-strong leakage-resilient non-malleable extractor $\mathsf{2SLNMExt}$.We infer from Lemma 5.9 that the following two distributions are $O(\varepsilon + t2^{-3m})$ close.

$\underline{\mathcal{D}_0}$: Sample independent $x, x' \sim U_{n_1}$ and $y, y' \sim U_{n_2}$. Output $\mathsf{2SLNMExt}(x, y) \circ \mathsf{2SLNMExt}(x', y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x, y), \{\mathsf{2SLNMExt}(f_i(x), g_i(y, h(x)))\}_{i \in \{i_1, \ldots, i_t\}})$.

$\underline{\mathcal{D}_1}$: Sample independent $x, x' \sim U_{n_1}$ and $y, y' \sim U_{n_2}$. Output $\mathsf{2SLNMExt}(x, y) \circ \mathsf{2SLNMExt}(x', y') \circ \mathsf{Sanitize}(\mathsf{2SLNMExt}(x', y'), \{\mathsf{2SLNMExt}(f_i(x'), g_i(y', h(x')))\}_{i \in \{i_1, \ldots, i_t\}})$.

Given a sample from $\mathcal{D}_0$ or $\mathcal{D}_1$, we first give an algorithm $\mathcal{A}$ that processes the sample as follows:

1. Parse the sample as $\mathsf{2SLNMExt}(x, y) \circ \mathsf{2SLNMExt}(x', y') \circ (a_{i_1}, \ldots, a_{i_t})$.
2. Define the function $\mathsf{Truncate}$ that takes in $(c_1, \ldots, c_t)$ and outputs $b_1, \ldots, b_t$ where $b_i = \mathsf{same}^*$ if $c_i = \mathsf{same}^*$; else, $b_i = (c_i)_{[m]}$.

3. Let $(b_{i_1}, \ldots, b_{i_t}) := \mathsf{Truncate}(a_{i_1}, \ldots, a_{i_t})$.

4. Output $\mathsf{2SLNMExt}(x, y)_{[m]} \circ \mathsf{2SLNMExt}(x', y')_{[m]} \circ (b_{i_1}, \ldots, b_{i_t})$.

Since $\mathcal{D}_0$ and $\mathcal{D}_1$ are $O(\varepsilon + t2^{-3m})$-close, it follows that

$$\mathcal{A}(\mathcal{D}_0) \approx_{O(\varepsilon + t2^{-3m})} \mathcal{A}(\mathcal{D}_1)$$

Now, $\mathsf{2SLNMExt}(x, y)$ and $\mathsf{2SLNMExt}(x', y')$ are ε-close to the uniform distribution, and hence, it follows from Lemma 3.7 that for any Sh_ℓ in $\{0,1\}^m$,

$$\begin{aligned}
&(\mathsf{Sh}_\ell, \mathsf{Sh}'_\ell, \mathsf{Truncate}(\mathsf{Sanitize}(\mathsf{2SLNMExt}(x, y), \{\mathsf{2SLNMExt}(f_i(x), g_i(y, h(x)))\}_{i \in \{i_1, \ldots, i_t\}}))) \\
\approx_{O(\varepsilon 2^m + t2^{-2m})} &(\mathsf{Sh}_\ell, \mathsf{Sh}'_\ell, \mathsf{Truncate}(\mathsf{Sanitize}(\mathsf{2SLNMExt}(x', y'), \{\mathsf{2SLNMExt}(f_i(x'), g_i(y', h(x')))\}_{i \in \{i_1, \ldots, i_t\}})))
\end{aligned}$$

where $(x, y) \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{Sh}_\ell \| U_{2m})$, $\mathsf{Sh}'_\ell \sim U_m$ and $(x', y') \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{Sh}'_\ell \| U'_{2m})$.

We now describe an algorithm $\mathcal{B}$ that simulates the output of Hyb_ℓ or $\mathsf{Hyb}_{\ell+1}$ when given a sample from the first distribution or the second distribution of the above equation. $\mathcal{B}$ on input $\mathsf{Sh}_\ell, \mathsf{Sh}'_\ell, b_{i_1}, \ldots, b_{i_t}$ where Sh'_ℓ does the following:

1. If for any $j \in \{i_1, \ldots, i_t\}$, $b_j = \mathsf{same}^*$, replace b_j with Sh'_ℓ.

2. For every $j \in \{i_1, \ldots, i_t\}$ such that $b_j = \mathsf{Sh}'_\ell$, replace b_j with Sh_ℓ.

3. For every $j \in \{i_1, \ldots, i_t\}$ such that $b_j = \mathsf{Sh}'_i$ for some $i \in [n] \setminus \{\ell\}$, replace b_j with Sh_i.

4. Output $\mathsf{Rec}(b_{i_1}, \ldots, b_{i_t})$.

If $\mathcal{B}$ was given a sample from the first distribution then the output of $\mathcal{B}$ is $O(t2^{-m})$-close to Hyb_ℓ since in this case, the probability that some $b_j = \mathsf{Sh}'_\ell$ is 2^{-m}; else, it is identical to $\mathsf{Hyb}_{\ell+1}$. This completes the proof of the claim. ■

By repeated application of Claim 8.6, we infer that $\mathsf{Hyb}_1 \approx_{O(n(\varepsilon \cdot 2^{3m} + t \cdot 2^{-m}))} \mathsf{Hyb}_{n+1}$. We now define another hybrid Hyb_{n+2} and observe that Hyb_{n+1} is identical to Hyb_{n+2}.

$\underline{\mathsf{Hyb}_{n+2}}$.

1. Compute $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n) \leftarrow \mathsf{Share}(s)$.

2. For each $i \leq n$, sample $\mathsf{Sh}'_i \leftarrow \{0,1\}^m$.

3. For each $i \in [n]$, compute $(\mathsf{L}^{(i)}, \mathsf{R}^{(i)}) \leftarrow \mathsf{2SLNMExt}^{-1}(\mathsf{Sh}'_i \circ U_{2m})$.

4. For each $i \in [n]$, $(\mathsf{R}^{(i)}_1, \ldots, \mathsf{R}^{(i)}_n) \leftarrow \mathsf{Share}(\mathsf{R}^{(i)})$.

5. Set $\mathsf{share}_i = (\mathsf{L}^{(i)}, \mathsf{R}^{(1)}_i, \ldots, \mathsf{R}^{(n)}_i)$.

6. For each $i \in \{i_1, \ldots, i_t\}$, compute $f_{T_i}(\mathsf{share}_{T_i}) = \widetilde{\mathsf{share}}_i$.

7. Compute $(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t}) := \mathsf{split}(\widetilde{\mathsf{share}}_{i_1}, \ldots, \widetilde{\mathsf{share}}_{i_t})$.

8. If for any $j \in \{i_1, \ldots, i_t\}$, $\widetilde{\mathsf{Sh}}_j = \mathsf{Sh}'_i$ for some $i \in [n]$, reset $\widetilde{\mathsf{Sh}}_j$ with Sh_i.

9. Output $\mathsf{Rec}(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t})$.

We are now ready to define our simulator $D^{f,T}$.

$\underline{D^{f,T}}$.

1. For each $i \leq n$, sample $\mathsf{Sh}'_i \leftarrow \{0,1\}^m$.
2. For each $i \in [n]$, compute $(\mathsf{L}^{(i)}, \mathsf{R}^{(i)}) \leftarrow \mathsf{SEnc}(\mathsf{Sh}'_i \circ U_{2m})$.
3. For each $i \in [n]$, $(\mathsf{R}_1^{(i)}, \ldots, \mathsf{R}_n^{(i)}) \leftarrow \mathsf{Share}(\mathsf{R}^{(i)})$.
4. Set $\mathsf{share}_i = (\mathsf{L}^{(i)}, \mathsf{R}_i^{(1)}, \ldots, \mathsf{R}_i^{(n)})$.
5. For each $i \in \{i_1, \ldots, i_t\}$, compute $f_{T_i}(\mathsf{share}_{T_i}) = \widetilde{\mathsf{share}}_i$.
6. Compute $(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t}) := \mathsf{split}(\widetilde{\mathsf{share}}_{i_1}, \ldots, \widetilde{\mathsf{share}}_{i_t})$.
7. If for any $j \in \{i_1, \ldots, i_t\}$, $\widetilde{\mathsf{Sh}}_j = \mathsf{Sh}'_i$ for some $i \in [n]$, reset $\widetilde{\mathsf{Sh}}_j$ with same^*_i.
8. If $(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t})$ is of the form $\{\mathsf{same}^*_{k_1}, \ldots, \mathsf{same}^*_{k_t}\}$ for distinct $k_1, \ldots, k_t$, then output same^*.
9. Sample $\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n \leftarrow \mathsf{Share}(0^m)$.
10. Replace each same^*_i in $(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t})$ with Sh_i.
11. Output $\mathsf{Rec}(\widetilde{\mathsf{Sh}}_{i_1}, \ldots, \widetilde{\mathsf{Sh}}_{i_t})$.

It now follows from the perfect privacy of Shamir secret sharing that:

$$\mathsf{Hyb}_{n+2} \equiv \mathsf{copy}(D^{f,T}, s)$$

■

8.3 Instantiation

From Corollary 5.7, by setting $p(x) = n^2 x$, for some $\gamma > 0$ and any n_2, there exists a $(2, n_2^\gamma)$-non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^{3m}$ at min-entropy (n_1, n_2) and error $\varepsilon = 2^{-n_2^\gamma}$, where $n_1 = 4n_2 + p(n_2)$ and $m = n_2^{\Omega(1)}$. Let $n = n_2^\gamma, m = C_1 n_2^{\gamma_1}, \varepsilon' = 2^{-C_2 n_2^{\gamma_2}}$, such that

$$0 < \gamma_2 < \gamma_1 < \gamma,\ 0 < C_1 < C_2,$$

$$0 < 3C_1 + C_2 < 1.$$

Then we have $O(n(\varepsilon 2^{3m} + t2^{-m})) < \varepsilon'$, $\varepsilon < 1/2^{3m}$ and $t \leq n = n_2^\gamma$. Therefore, from Theorem 8.4, we could construct a $(t, n, 0, 0)$ secret sharing scheme that is ε'-non-malleable against $\mathcal{F}_{t-\mathsf{cover-free}}$ with message length m. The length of each share is $w = n_1 + nn_2 = n_2^{O(1)}$. Therefore, we have the following corollary.

Corollary 8.7 *For every $t \geq 2$, $n \geq t$ and any $m \in \mathbb{N}$, there exists an efficient construction of t-out-of-n non-malleable secret sharing for secrets of length m against t-cover-free tampering with error $2^{-m^{\Omega(1)}}$.*

9 Network Extractor Protocol

In this section, we show that a strong version of s-source non-malleable extractors give rise to a network extractor protocol. We start with the definition of a network extractor protocol from [KLRZ08].

Notation. We follow the same notation that was used in [KLRZ08]. Processor i begins with a sample from a weak source $x_i \in \{0,1\}^n$ and ends in possession of a hopefully uniform sample $z_i \in \{0,1\}^m$. Let b be the concatenation of all the messages that were sent during the protocol. Capital letters such as X_i, Z_i and B denote these strings viewed as random variables.

Definition 9.1 (Network Extractor Protocol [KLRZ08]) *A protocol for p processors is a (t, g, ε) network extractor for min-entropy k if for any (n,k) independent sources $X_1, \ldots, X_p$ and any choice T of t faulty processors, after running the protocol, there exists a set $G \in [p] \setminus T$ of size at least g such that*

$$|B, \{X_i\}_{i \notin G}, \{Z_i\}_{i \in G} - B, \{X_i\}_{i \notin G}, U_{gm}| < \varepsilon$$

Here U_{gm} is the uniform distribution on gm bits, independent of B, and $\{X_i\}_{i \notin G}$.

9.1 Building Block

In this subsection, we give a building block that will be used in the construction of network extractor protocols.

Weak Disjoint Tampering function family. The weak disjoint tampering function family $\mathcal{F}_{\mathsf{wDis}}$ is the set of all functions given by $f = (i, g)$. Given $(x_1, \ldots, x_s)$, f outputs $\widetilde{x}_1, \ldots, \widetilde{x}_s$ where $\widetilde{x}_i = x_i$ and $g(x_{[s]\setminus\{i\}}) = \widetilde{x}_{[s]\setminus\{i\}}$. In other words, the tampering function leaves the i-th source as it is, and for the rest of the sources, it applies the tampering function g to generate their tampered version.

Below, we give an useful definition.

Definition 9.2 *The function* $\mathsf{Deduplicate}$ *takes in $a_1, \ldots, a_t$ and removes all the duplicates in the input. That is, if for any $i \in [s]$, $a_i = a_{i_1} = \ldots = a_{i_\ell}$ where $i < i_1 < \ldots < i_\ell$, then* $\mathsf{Deduplicate}$ *removes $a_{i_1}, \ldots, a_{i_\ell}$.*

We are now ready to give the definition of the building block.

Definition 9.3 ((s,t)-Strong Multi-Source Non-Malleable Extractors) *A function* $\mathsf{MNMExt} : \{0,1\}^n \times \{0,1\}^n \ldots \times \{0,1\}^n \to \{0,1\}^m$ *is a (s,t)-strong non-malleable extractor against the tampering family $\mathcal{F}_{\mathsf{wDis}}$ at min-entropy k and error ε if it satisfies the following property: If $X_1, \ldots, X_s$ are independent (n,k)-sources and for any $f_1 = (i, g_1), \ldots, f_t = (i, g_t) \in \mathcal{F}_{\mathsf{wDis}}$, there exists a random variable $D_{\overrightarrow{f}}$ with support on $(\{0,1\}^m)^t$ which is independent of the random variables $X_1, \ldots, X_s$, such that*

$$|X_{[s]\setminus\{i\}}, \mathsf{Deduplicate}(\mathsf{MNMExt}(X), \mathsf{MNMExt}(f_1(X)), \ldots, \mathsf{MNMExt}(f_t(X))) - X'_{[s]\setminus\{i\}}, U_m, Z| < \varepsilon$$

where $X = (X_1, \ldots, X_s)$, U_m refers to an uniform m-bit string and $(X'_{[s]\setminus\{i\}}, Z) \sim D_{\overrightarrow{f}}$.

We show in Appendix A that the construction from Section 4 satisfies this definition for $s = 2$.

9.2 The protocol

In this subsection, we give the description of our network extractor protocol. Let p be the number of processors and Δ denote the entropy loss parameter. We use a $(s, \binom{p}{s-1})$-strong non-malleable extractor $\mathsf{MNMExt} : (\{0,1\}^{n/p})^s \to \{0,1\}^m$ for min-entropy $n/p - \Delta$ and error ε against tampering family $\mathcal{F}_{\mathsf{wDis}}$.

Protocol 1. On input $x_i \in \{0,1\}^n$, processor i does the following.

1. Parse x_i as $x_1^{(i)}, \ldots, x_p^{(i)}$.
2. Broadcast $\{x_j^{(i)}\}_{j \neq i}$.
3. Receive $\{x_i^{(j)}\}_{j \neq i}$ from all the processors. If some processor j does not send any message, replace $x_i^{(j)}$ with a default value.
4. For every set $\{i_1, \ldots, i_{s-1}\} \subseteq [p]$ of size $s-1$,
   (a) Compute $y_{i_1,\ldots,i_{s-1}} = \mathsf{MNMExt}(x_i^{(i)}, x_i^{(i_1)}, \ldots, x_i^{(i_{s-1})})$.
5. Remove the duplicates from the sequence $(y_{i_1,\ldots,i_{s-1}})_{i_1,\ldots,i_{s-1}}$ to get $y'_1, \ldots, y'_k$.
6. Output $z_i = y'_1 \oplus \ldots \oplus y'_k$.

Theorem 9.4 *For any $p, s, n \in \mathbb{N}$, assume $(s, \binom{p}{s-1})$-strong non-malleable extractor $\mathsf{MNMExt} : (\{0,1\}^{n/p})^s \to \{0,1\}^m$ for min-entropy $n/p - \Delta$ and error ε against tampering family $\mathcal{F}_{\mathsf{wDis}}$. Then, for any $t \leq p - s$ and $g = p - t$, protocol 1 is a $(t, g, 2g \cdot \varepsilon)$ network extractor protocol for min-entropy $n - \Delta + \log(1/\varepsilon)$. When $s = O(1)$, the running time of the protocol is* $\mathrm{poly}(n, p)$.

Proof Let us denote T of size of at most t as the set of faulty processors and let $G = [p] \setminus T$ and let $g = p - t$. We are required to show that

$$|B, \{X_i\}_{i \notin G}, \{Z_i\}_{i \in G} - B, \{X_i\}_{i \notin G}, U_{gm}| < 2g \cdot \varepsilon \tag{9.1}$$

Let $G = \{i_1, \ldots, i_{p-t}\}$. We prove equation 9.1 via a hybrid argument. For every $j \in [p - t + 1]$, we define Hyb_j as the distribution $B, \{X_i\}_{i \notin G}, U_{(j-1)m}, Z_{i_j}, \ldots, Z_{i_{p-t}}$. Notice that Hyb_1 is identical to the first distribution in equation 9.1 and Hyb_{p-t+1} is identical to the second distribution in equation 9.1. We now show the following claim which directly proves equation 9.1.

Claim 9.5 *For every $j \in [p - t]$, $\mathsf{Hyb}_j \approx_{2\varepsilon} \mathsf{Hyb}_{j+1}$.*

Proof Let us fix $\{X_i\}_{i \notin G}$ and for every $i \in G$, fix all the values in X_i except $X_{i_j}^{(i)}$. Now, it follows from Lemma 3.2 that with probability at least $1 - \varepsilon$, conditioned on all the fixed values, $\{X_{i_j}^{(i)}\}_{i \in G}$ are independent sources with min-entropy at least $n/p - \Delta$.

Let us consider the honest processor i_j. Let us assume without loss of generality that $t = p - s$. This means that $|G| = s$, and hence, there are $s - 1$ honest processors other than i_j. For every set $i'_1, \ldots, i'_{s-1}$, that contains at least one faulty processor, we can view the random variables

$X_{i_j}^{(i'_1)}, \ldots, X_{i_j}^{(i'_{s-1})}$ in the transcript B and $X_{i_j}^{(i_j)}$ as a tampering of the sources $\{X_{i_j}^{(i)}\}_{i \in G}$ using a tampering function $f_{i'_1,\ldots,i'_{s-1}]} \in \mathcal{F}_{\mathsf{wDis}}$ (we associate each set $\{i'_1, \ldots, i'_{s-1}\}$ with a canonical number from $[\binom{p}{s-1}]$). Thus, it follows from property of (s,t) strong non-malleable extractor that,

$$|\{X_{i_j}^{(i)}\}_{i \in G \setminus \{i_j\}}, \mathsf{Deduplicate}(\mathsf{MNMExt}(X), \mathsf{MNMExt}(f_1(X)), \ldots, \mathsf{MNMExt}(f_t(X))) - \{\overline{X}_{i_j}^{(i)}\}_{i \in G \setminus \{i_j\}}, U_m, Z| < \varepsilon$$

where $X = (X_{i_j}^{(i_j)}, \{X_{i_j}^{(i)}\}_{i \in G \setminus \{i_j\}})$ and $(\{\overline{X}_{i_j}^{(i)}\}_{i \in G \setminus \{i_j\}}, Z) \sim D_{\overrightarrow{f}}$. Notice that the first distribution in the above equation can be used to generate B,$\{Z_i\}_{i \neq i_j}$, Z_{i_j} as in Hyb_j and the second distribution can be used to generate B, $\{Z_i\}_{i \neq i_j}$, U_m as in Hyb_{j+1}. ■

■

Instantiation Let $s = 2$. Instantiate MNMExt from Corollary A.3 with min-entropy $(1-\gamma)n/p$ and error $\varepsilon = (n/p)^{\gamma_1}$ for some $0 < \gamma, \gamma_1 < 1$. By Theorem 9.4, for any $t \leq p-2$ and $g = p - t$, we get a $(t, g, 2g \cdot \varepsilon)$ network extractor protocol at min-entropy $n(1-\gamma')$ for some $\gamma' < \gamma/p$. We summarize the instantiation with the following corollary.

Corollary 9.6 *For any $p \geq 2$, there exists constants $\gamma, n_0 > 0$ and γ such that for all $n > n_0$ and for any $t \leq p-2$, there exists a single-round, $(t, p-t, 2^{-n^{\Omega(1)}})$-network extractor protocol for p processors and $(n, n(1-\gamma))$ sources.*

References

[ADKO15] Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 398–426. Springer, Heidelberg, March 2015.

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

[BDT17] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1185–1194, 2017.

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

[CG85] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 429–442. IEEE Computer Society Press, October 1985.

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

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 440–464. Springer, Heidelberg, February 2014.

[CGGL19] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. Manuscript, 2019. .

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 285–298. ACM Press, June 2016.

[Coh16a] Gil Cohen. Making the most of advice: New correlation breakers and their applications. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 188–196, 2016.

[Coh16b] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved ramsey graphs. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 278–284. ACM Press, June 2016.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 670–683. ACM Press, June 2016.

[DO03] Yevgeniy Dodis and Roberto Oliveira. On extracting private randomness over a public channel. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24-26, 2003, Proceedings*, pages 252–263, 2003.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38:97–139, 2008.

[DPW18] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, 2018.

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

[GK18a] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 685–698. ACM Press, June 2018.

[GK18b] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 501–530. Springer, Heidelberg, August 2018.

[GKK19] Ankit Garg, Yael Tauman Kalai, and Dakshita Khurana. Computational extractors with negligible error in the crs model. Cryptology ePrint Archive, Report 2019/1116, 2019. `https://eprint.iacr.org/2019/1116`.

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

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 1128–1141. ACM Press, June 2016.

[GS19] Vipul Goyal and Yifan Song. Correlated-source extractors and cryptography with correlated-random tapes. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 562–592, 2019.

[GSV05] Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In *Distributed Computing, 19th International Conference, DISC 2005, Cracow, Poland, September 26-29, 2005, Proceedings*, pages 288–302, 2005.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):20, 2009.

[KLR09] Yael Tauman Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *50th Annual Symposium on Foundations of Computer Science*, pages 617–626. IEEE Computer Society Press, October 2009.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *49th Annual Symposium on Foundations of Computer Science*, pages 654–663. IEEE Computer Society Press, October 2008.

[KMS18] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:200, 2018.

[Li13] Xin Li. New independent source extractors with exponential improvement. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 783–792. ACM Press, June 2013.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In Irit Dinur, editor, *57th Annual Symposium on Foundations of Computer Science*, pages 168–177. IEEE Computer Society Press, October 2016.

[Li17a] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. *STOC*, 2017.

[Li17b] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 1144–1156. ACM Press, June 2017.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer, Heidelberg, August 1997.

[Raz05] Ran Raz. Extractors with weak random seeds. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 11–20. ACM Press, May 2005.

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *JCSS*, 65(1):97–128, 2002.

[SV19] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 480–509, 2019.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.

A $(2,t)$-Strong Non-Malleable Extractors

In this section, we give a proof that a $(2,t)$-non-malleable extractor against the tampering family $\mathcal{F}_{\mathsf{wDis}}$ is also a $(2,t)$-strong non-malleable extractor against $\mathcal{F}_{\mathsf{wDis}}$ (see Definition 9.3). The high level ideas are from [Li17a] and [CGL16].

To apply the argument in [Li17a], we use the stronger definition of $(2,t)$-non-malleable randomness extractors given in Definition 4.1. Using similar ideasn as in [Li17a], we can show the $(2,t)$-non-malleable extractor is also a strong extractor, which is formalized in the following lemma. We include the proof for the sake of completeness.

Lemma A.1 ([Li17a]) *Let* $\mathsf{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a* $(2,t)$*-non-malleable extractor at min-entropy* k *and error* ε *against* $\mathcal{F}_{\mathsf{wDis}}$ *which satisfies the Definition 4.1. Then* NMExt *is also a* $(2,t)$*-non-malleable extractor at min-entropy* k' *and error* $\varepsilon' = 2^{m(t+1)}(\varepsilon + 2^{k+1-k'})$ *with the following property: if* X_1 *and* X_2 *are independent* (n,k)*-sources and* $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ *are* t *arbitrary 2-split-state tampering functions such that for all* $i \in [t]$*,* f_i *or* g_i *has no fixed points, then it holds that*

$$\begin{aligned}|X_1, \mathsf{2NMExt}(X_1, X_2), \mathsf{2NMExt}(f_1(X_1), g_1(X_2)), \ldots, \mathsf{2NMExt}(f_t(X_1), g_t(X_2))\\ -X_1, U_m, \mathsf{2NMExt}(f_1(X_1), g_1(X_2)), \ldots, \mathsf{2NMExt}(f_t(X_1), g_t(X_2))| < \varepsilon'.\end{aligned}$$

Proof Let (X_1, X_2) be independent (n,k) sources. Fix the tampering function $\mathcal{A}_1 = (f_1, g_1)$, $\ldots, \mathcal{A}_t = (f_t, g_t)$ such that for all $i \in [t]$, f_i or g_i has no fixed points.

Since NMExt is a $(2,t)$-non-malleable extractor at min-entropy k and error ε, it holds that

$$\begin{aligned}|\mathsf{NMExt}(X_1, X_2), \mathsf{NMExt}(\mathcal{A}_1(X_1, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(X_1, X_2))\\ -U_m, \mathsf{NMExt}(\mathcal{A}_1(X_1, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(X_1, X_2))| \le \varepsilon.\end{aligned}$$

For each $z, z_1, \ldots, z_t \in \{0,1\}^m$, let $\overrightarrow{z} = (z_1, \ldots, z_t)$ and define the two sets as follows.

$$\begin{aligned}B^+_{z,\overrightarrow{z}} = \{y : \Pr[\mathsf{NMExt}(y, X_2), \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})]\\ - \Pr[(U_m, \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2))) = (z, \overrightarrow{z})] > \varepsilon\},\end{aligned}$$

$$\begin{aligned}B^-_{z,\overrightarrow{z}} = \{y : \Pr[\mathsf{NMExt}(y, X_2), \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})]\\ - \Pr[(U_m, \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2))) = (z, \overrightarrow{z})] < -\varepsilon\},\end{aligned}$$

We will then prove that $|B^+_{z,\overrightarrow{z}}| \le 2^k$ and $|B^-_{z,\overrightarrow{z}}| \le 2^k$. If not, without loss of generality assume $|B^+_{z,\overrightarrow{z}}| > 2^k$. Define random variable Y as uniformly sampling y from $B^+_{z,\overrightarrow{z}}$. Then, Y is a (n,k)-source.

$$\begin{aligned}&\Pr[\mathsf{NMExt}(Y, X_2), \mathsf{NMExt}(\mathcal{A}_1(Y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(Y, X_2)) = (z, \overrightarrow{z})]\\ &\quad - \Pr[U_m, \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})]\\ = &\sum_{y \in B^+_{z,\overrightarrow{z}}} \Pr[Y = y] \Pr[\mathsf{NMExt}(y, X_2), \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})]\\ &\quad - \Pr[(U_m, \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})]\\ &> \varepsilon,\end{aligned}$$

which contradicts with the fact that NMExt is a $(2,t)$-non-malleable extractor with error ε. Therefore, we have $|B^+_{z,\overrightarrow{z}} \cup B^-_{z,\overrightarrow{z}}| \le 2^{k+1}$. Define $B = \cup_{z,z_1,\ldots,z_t \in \{0,1\}^m}(B^+_{z,\overrightarrow{z}} \cup B^-_{z,\overrightarrow{z}})$. Then we have $|B| \le 2^{m(t+1)+(k+1)}$. Therefore, it holds that

$$
\begin{aligned}
& |X_1, \mathsf{NMExt}(X_1, X_2), \mathsf{NMExt}(\mathcal{A}_1(X_1, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(X_1, X_2)) \\
& -X_1, U_m, \mathsf{NMExt}(X_1, X_2), \mathsf{NMExt}(\mathcal{A}_1(X_1, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(X_1, X_2))| \\
= & \sum_{z, z_1, \ldots, z_t \in \{0,1\}^m} \Big| \sum_{y \in \{0,1\}^n} \Pr[X_1 = y] \Big(\Pr[\mathsf{NMExt}(y, X_2), \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})] \\
& - \Pr[(U_m, \mathsf{NMExt}(\mathcal{A}_1(y, X_2)), \ldots, \mathsf{NMExt}(\mathcal{A}_t(y, X_2)) = (z, \overrightarrow{z})] \Big) \Big| \\
\leq & \Pr[X_1 \in B] + \Pr[X_1 \notin B] 2^{m(t+1)} \varepsilon \\
\leq & 2^{m(t+1)} (\varepsilon + 2^{k+1-k'}).
\end{aligned}
$$

■

We now show the above strong extractor satisfies security guarantees of $(2, t)$-strong non-malleable extractor defined in the Definition 9.3 using the idea in [CGL16].

Theorem A.2 ([CGL16]) *Let* $\mathsf{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a* $(2, t)$*-non-malleable extractor at min-entropy* k *and error* ε *against the tampering family* $\mathcal{F}_{\mathsf{wDis}}$ *which satisfies the Definition 4.1). Then, it is also a* $(2, t)$*-strong non-malleable extractor (see Definition 9.3) against the tampering family* $\mathcal{F}_{\mathsf{wDis}}$ *at min-entropy* k' *and error* $2^{m(t+1)+t}(\varepsilon + 2^{k+2-k'}) + t2^{t-m}$.

Proof Let (X_1, X_2) be independent (n, k) sources. Without loss of generality assume the tampering function is $f_1 = (1, g_1), \ldots, f_t = (1, g_t) \in \mathcal{F}_{\mathsf{wDis}}$. Let $\varepsilon' = 2^{m(t+1)}(\varepsilon + 2^{k+1-k'})$. For any $R \in [t]$, define an event

$$E^{(R)} = \{x_2 \in \{0,1\}^n : g_i(x_2) \neq x_2 \text{ if } i \in R \text{ and } g_i(x_2) = x_2 \text{ if } i \notin R\}.$$

Let $X_2^{(R)}$ be the random variable X_2 conditioned on $E^{(R)}$ and $\alpha_R = \Pr[X_2 \in E^{(R)}]$. Let $R = \{i_1, ..., i_r\}$ and denote the tampering results as

$$Z = \mathsf{NMExt}(f_{i_1}(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_{i_r}(X_1, X_2^{(R)})).$$

Define $D_{\overrightarrow{f}}^{(R)} = (X_2^{(R)}, \mathsf{Deduplicate}(Z))$.

We say $R \in [t]$ is good if $\alpha_R \geq 2^{k-k'}$. For a good R, we have the entropy of $X_2^{(R)}$ is at least k. Since g_{i_k} doesn't have fixed points on $X_2^{(R)}$ for $1 \leq k \leq r$, by Lemma A.1 we have

$$|X_2^{(R)}, \mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_{i_1}(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_{i_r}(X_1, X_2^{(R)})) - X_2^{(R)}, U_m, Z| \leq \varepsilon'.$$

Therefore, we have

$$
\begin{aligned}
|X_2^{(R)}, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_{i_1}(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_{i_r}(X_1, X_2^{(R)}))) \\
-X_2^{(R)'}, \mathsf{Deduplicate}(U_m, Z)| \leq \varepsilon'
\end{aligned}
$$

Since

$$
\begin{aligned}
& X_2^{(R)}, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_1(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_s(X_1, X_2^{(R)})) \\
= & X_2^{(R)}, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_{i_1}(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_{i_r}(X_1, X_2^{(R)}))
\end{aligned}
$$

and

$$|X_2^{(R)}, \mathsf{Deduplicate}(U_m, Z) - X_2^{(R)}, U_m, \mathsf{Deduplicate}(Z)| \leq t2^{-m},$$

we have

$$|X_2^{(R)}, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_1(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_s(X_1, X_2^{(R)}))) \\ -X_2^{(R)'}, U_m, Z'| \leq \varepsilon' + t2^{-m},$$

where $(X_2^{(R)'}, Z') = D_{\overrightarrow{f}}^{(R)}$.

We now combine all the $D_{\overrightarrow{f}}^{(R)}$ as

$$D_{\overrightarrow{f}} = \sum_{R\in[t]} \alpha_R D_{\overrightarrow{f}}^{(R)}.$$

Let $(X_2^{(R)'}, Z) = D_{\overrightarrow{f}}$ and $(X_2^{(R)\prime}, Z^{(R)}) = D_{\overrightarrow{f}}^{(R)}$. From the above equation, it holds that

$$\begin{aligned}
& |X_2, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2), \mathsf{NMExt}(f_1(X_1, X_2)), \ldots, \mathsf{NMExt}(f_s(X_1, X_2))) - X_2', U_m, Z| \\
\leq\ & \sum_{R\in[t]} \alpha_R |X_2^{(R)}, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_1(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_s(X_1, X_2^{(R)}))) \\
& -X_2'^{(R)}, U_m, Z^{(R)}| \\
\leq\ & \sum_{R\in[t], R \text{ is good}} \alpha_R \Big| X_2, \mathsf{Deduplicate}(\mathsf{NMExt}(X_1, X_2^{(R)}), \mathsf{NMExt}(f_1(X_1, X_2^{(R)})), \ldots, \mathsf{NMExt}(f_s(X_1, X_2^{(R)}))) \\
& -X_2'^{(R)}, U_m, Z^{(R)} \Big| + \sum_{R\in[t], R \text{ is bad}} \alpha_R \\
\leq\ & 2^t(2^{k-k'} + \varepsilon' + t2^{-m}) \\
\leq\ & 2^{m(t+1)+t}(\varepsilon + 2^{k+2-k'}) + t2^{t-m}.
\end{aligned}$$

■

A.1 Instantiation

By Corollary 4.6, there exists a $(2,t)$-non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ at min-entropy $k = n - \gamma n$ and error $\varepsilon = 2^{-C_1 n_1^\gamma}$ for some constant γ, C_1, γ_1, where $m = n^{\Omega(1)}$ satisfying Definition 4.1. Let $k' = n - \gamma_2 n, \varepsilon' = C_2 n^{\gamma_3}, m = C_3 n^{\gamma_4}$, such that

$$0 < \gamma_3 < \gamma_4 < 1,$$
$$C_2 + C_3(t+1) + \gamma_2 < \gamma + C_1,$$
$$C_2 > C_3.$$

Then for large enough n, we have $2^{m(t+1)+t}(\varepsilon + 2^{k+2-k'}) + t2^{t-m} < \varepsilon'$. Therefore, from Theorem A.2, we have the following corollary.

Corollary A.3 *For any $t \geq 1$, there exists constant $n_0, \gamma > 0$ such that for any $n > n_0$ there exists a $(2,t)$-strong non-malleable extractor $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ against $\mathcal{F}_{\mathsf{wDis}}$ at min-entropy $n(1-\gamma)$ and error $2^{-n^{\Omega(1)}}$ with output length $m = n^{\Omega(1)}$.*

B A Simple Construction of Multi-Source Non-Malleable Extractors

In this part, we give an alternative construction of multi-source non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ tampering, which is simpler, but is not efficiently pre-image sampleable.

Building Blocks and Parameters. We will use the following building blocks and set the parameters as shown below.

- Let $\mathsf{2NMExt} : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^m$ be a $(2, s^2)$-non-malleable extractor at min-entropy k and error ε in Definition 4.1.
- $\mathsf{2NMExt}$ is also symmetric, which means $\mathsf{2NMExt}(x, y) = \mathsf{2NMExt}(y, x)$ for all x, y.
- We set $n' = n + \log(s)$.

Construction 5. On input strings $(x_1, \ldots, x_s)$ where each $x_i \in \{0,1\}^n$, the function MNMExt is computed as follows:

1. For $i \in [s]$, let $N_i \in \{0,1\}^{\log(s)}$ be the binary representation of number i.
2. Output $\oplus_{i<j} \mathsf{2NMExt}(x_i \circ N_i, x_j \circ N_j)$

Theorem B.1 MNMExt *described above is a s-source non-malleable extractor against $\mathcal{F}_{\mathsf{cover-free}}$ at min-entropy n and error $3\varepsilon + (s+1)2^{k'-n}$, where $k' = k + \log(s^2 n) + \log(1/\varepsilon)$.*

The theorem could extend to the case that the sources do not have full entropy, but for the sake of simplicity here we only consider the case that all the sources have full entropy. The basic idea of the proof is to divide the source space $\{(x_1, \ldots, x_s) \in \{0,1\}^{ns}\}$ according to the tampering result $(\tilde{x}_1, \ldots, \tilde{x}_s)$. If $(\tilde{x}_1, \ldots, \tilde{x}_s)$ is the same as $(x_1, \ldots, x_s)$, then the tampering output is the same as the correct output. Otherwise, there exists $\tilde{x}_i$ that is not equal to x_i. Since the tampering function is cover-free, there exists i^* such that x_{i^*} is not tampered together with x_i, which means x_i and x_{i^*} are tampered independently. Then, we could show that the tampered output is independent of the correct output via a reduction to the multi-tampering of the underlying 2-source non-malleable extractor.

Proof Let us fix a tampering function $f = (f_{T_1}, \ldots, f_{T_s}) \in \mathcal{F}_{\mathsf{cover-free}}$. For any $x_1, \ldots, x_s \in \{0,1\}^n$, let $(\tilde{x}_1, \ldots, \tilde{x}_s) = f(x_1, \ldots, x_s)$.
Define

$$B_0 = \{(x_1, \ldots, x_s) \in \{0,1\}^{ns} : (\tilde{x}_1, \ldots, \tilde{x}_s) = (x_1, \ldots, x_s)\}$$

and

$$B_i = \{(x_1, \ldots, x_s) \in \{0,1\}^{ns} : (\tilde{x}_1, \ldots, \tilde{x}_{i-1}) = (x_1, \ldots, x_{i-1}) \text{ and } \tilde{x}_i \neq x_i\}.$$

Then, the set $\{(x_1, \ldots, x_s) \in \{0,1\}^{sn}\}$ is divided into $B_0, B_1, \ldots, B_s$.

Claim B.2 *For each subset B_i, there exists a distribution D_f^i and a good subset B_i' of B_i such that*

$$|\mathsf{MNMExt}(X_1,\ldots,X_s),\mathsf{MNMExt}(\tilde{X}_1,\ldots,\tilde{X}_s)-U_m,copy(D_f^i,U_m)|<3\varepsilon,$$

where $X_1,\ldots,X_s$ are uniformly sampled from B_i'. Also, $|B_i\setminus B_i'|<2^{k'+n(s-1)}$, where $k'=k+\log(s^2n)+\log(1/\varepsilon)$.

Proof For B_i where $i=1,\ldots,s$, since f is in $\mathcal{F}_{\mathsf{cover-free}}$, there exists i^* such that i and i^* are not tampering together. Let $x_J=(\{x_j\}_{j\in[s]\setminus\{i,i^*\}})$.

Define $B_i(a_J)$ to be the number of elements $(b_1,\ldots,b_s)$ in B_i such that $b_J=a_J$.

Define $B_i'=\{(x_1,\ldots,x_s)\in B_i : B_i(x_J)\geq 2^{k'+n}\}$. Therefore, we have

$$\begin{aligned}|B_i\setminus B_i'| &= \sum_{x_J\in\{0,1\}^{n(s-2)}} B_i(x_J)-B_i'(x_J)\\ &= \sum_{x_J\in\{0,1\}^{n(s-2)},B_i(x_J)<2^{k'+n}} B_i(x_J)-B_i'(x_J)\\ &< 2^{k'+n(s-1)}.\end{aligned}$$

Let $(X_1,\ldots,X_s)$ be random variables sampled uniformly from B_i'. Let $T_{i_1},\ldots,T_{i_r}$ be all the sets among $T_1,\ldots,T_s$ that contain i and $R_i=\{i_1,\ldots,i_r\}$. Let us first fix $X_J=x_J$. Since X_i and X_{i^*} are not tampering together, X_i and X_{i*} are still independent and both sources must have entropy at least k'. Let

$$\tau_1=\{\mathsf{2NMExt}(X_i\circ N_i,X_j\circ N_j)\}_{j\in J}\cup\{\mathsf{2NMExt}(\tilde{X}_{j_1}\circ N_{j_1},\tilde{X}_{j_2}\circ N_{j_2})\}_{j_1,j_2\in R_i,j_1<j_2}$$

and

$$\tau_2=\{\mathsf{2NMExt}(X_{i^*}\circ N_{i^*},X_j\circ N_j)\}_{j\in J}\cup\{\mathsf{2NMExt}(\tilde{X}_{j_1}\circ N_{j_1},\tilde{X}_{j_2}\circ N_{j_2})\}_{j_1,j_2\in[s]\setminus R_i,j_1<j_2}.$$

Now we also fixed τ_1 and τ_2. Define τ_1 to be good if $H_\infty(X_i|\tau_1)\geq k$ and bad otherwise. Since the length of τ_1 is less than s^2n, by Lemma 3.2, $\Pr[\tau_1\in\mathsf{bad}]<\varepsilon$. Similarly, define τ_2 to be good if $H_\infty(X_{i^*}|\tau_2)\geq k$ and bad otherwise. By Lemma 3.2, $\Pr[\tau_2\in\mathsf{bad}]<\varepsilon$. Thus, $\Pr[\tau_1\in\mathsf{bad}\vee\tau_2\in\mathsf{bad}]<2\varepsilon$. Define split-state tampering functions $\overrightarrow{f}'$ and $\overrightarrow{g}'$ against the underlying $\mathsf{2NMExt}$ as follows.

- f_j': For every $j\in R_i$, f_j' on input $x\circ N\in\{0,1\}^{n'}$, where $N\in\{0,1\}^{\log(s)}$, sets $x_i=x$ and computes $f_{T_j}(x_{T_j})$ to obtain $\tilde{x}_j$. It outputs $\tilde{x}_j\circ N_j$.
- g_j': For every $j\in[s]\setminus R_i$, g_j' on input $x\circ N\in\{0,1\}^{n'}$, where $N\in\{0,1\}^{\log(s)}$, sets $x_{i*}=x$ and computes $f_{T_j}(x_{T_j})$ to obtain $\tilde{x}_j$. It outputs $\tilde{x}_j\circ N_j$.

Assume τ_1 and τ_2 are good. Since τ_1 is independent of X_{i^*} and τ_2 is independent of X_i, we have X_i and X_{i^*} are independent (n,k)-sources after fixing τ_1, τ_2 and $X_J=x_J$. Also, it is easy to see that f_j' does not have fixed point over $X_i\circ N_i$ for any $j\in R_i$. Since the underlying $\mathsf{2NMExt}$ is a $(2,s^2)$-non-malleable extractor at min-entropy k and error ε which satisfies Definition **??**, it holds that

$$\begin{aligned}|\mathsf{2NMExt}(X_i\circ N_i,X_{i^*}\circ N_{i^*}),\{\mathsf{2NMExt}(f_{j_1}'(X_i\circ N_i),g_{j_2}'(X_{i^*}\circ N_{i^*}))\}_{j_1\in R_i,j_2\in[s]\setminus R_i}\\ -U_m,\{\mathsf{2NMExt}(f_{j_1}'(X_i\circ N_i),g_{j_2}'(X_{i^*}\circ N_{i^*}))\}_{j_1\in R_i,j_2\in[s]\setminus R_i}|\leq\varepsilon.\end{aligned}$$

Let

$$C_1 = \oplus_{j_1,j_2\in[s],j_1<j_2,(j_1,j_2)\notin\{(i^*,i),(i,i^*)\}}\mathsf{NMExt}(X_{j_1}\circ N_{j_1}, X_{j_2}\circ N_{j_2})$$

and

$$C_2 = (\oplus_{j_1,j_2\in R_i,j_1<j_2}\mathsf{NMExt}(X_{j_1}\circ N_{j_1}, X_{j_2}\circ N_{j_2}))\oplus(\oplus_{j_1,j_2\in[s]\setminus R_i,j_1<j_2}\mathsf{NMExt}(X_{j_1}\circ N_{j_1}, X_{j_2}\circ N_{j_2})).$$

Since we fix τ_1, τ_2 and X_J, C_1 and C_2 are constants. Define a distribution

$$D_{\overrightarrow{f'},\overrightarrow{g'}} = (\oplus_{j_1\in R_i,j_2\in[s]\setminus R_i}2\mathsf{NMExt}(f'_{j_1}(X_i\circ N_i), g'_{j_2}(X_{i^*}\circ N_{i^*})))\oplus C_2.$$

Then, it holds that

$$\begin{aligned}
& |\mathsf{MNMExt}(X_1,\ldots,X_s), \mathsf{MNMExt}(f(X_1,\ldots,X_s)) - U_m, D_{\overrightarrow{f'},\overrightarrow{g'}}| \\
= \; & |C_1\oplus 2\mathsf{NMExt}(X_i\circ N_i, X_{i^*}\circ N_{i^*}), (\oplus_{j_1\in R_i,j_2\in[s]\setminus R_i}2\mathsf{NMExt}(f'_{j_1}(X_i\circ N_i), g'_{j_2}(X_{i^*}\circ N_{i^*})))\oplus C_2 \\
& -C_1\oplus U_m, D_{\overrightarrow{f'},\overrightarrow{g'}}| \\
< \; & \varepsilon.
\end{aligned}$$

Now we construct D^i_f as follows.

$\underline{D^i_f}$: Sample $(x_1,\ldots,x_s)$ from B'_i. Condition on $X_J = x_J$, if τ_1 and τ_2 are good, outputs $D_{\overrightarrow{f'},\overrightarrow{g'}}$. Otherwise, outputs 0.

Then, if $(X_1,\ldots,X_s)$ are random variables uniformly sampled from B'_i, it holds that

$$\begin{aligned}
& |\mathsf{MNMExt}(X_1,\ldots,X_s), \mathsf{MNMExt}(f(X_1,\ldots,X_s)) - U_m, D^i_f| \\
\le \; & \sum_{x_J\in\{0,1\}^{n(s-2)}}\Pr[X_J = x_J]\sum_{\tau_1,\tau_2}\Pr[\tau_1,\tau_2\in\mathsf{good}|X_J = x_J]\varepsilon + \Pr[\tau_1\in\mathsf{bad}\vee\tau_2\in\mathsf{bad}|X_J = x_J] \\
< \; & 3\varepsilon.
\end{aligned}$$

Finally, for B_0, we set $D^0_f = same^*$. Since for inputs in B_0 the tampered output is always equal to the correct output, it is only left to show that $\mathsf{MNMExt}(X_1,\ldots,X_s)$ is close to U_m. We pick an arbitrary pair of sources (x_i, x_{i^*}) that is not tampering together and set $x_J = (\{x_j\}_{j\in[s]\setminus\{i,i^*\}})$. Then, we define B'_0 and τ_1,τ_2,C_1 in the same way as above. Similarly, we have $|B_0\setminus B'_0| < 2^{k'+n(s-1)}$ and for a fixed x_J and fixed good τ_1,τ_2, X_i and X_{i^*} are independent (n,k)-sources and C_1 is constant. Therefore, conditioned on x_J and good τ_1,τ_2, it holds that

$$|\mathsf{MNMExt}(X_1,\ldots,X_s) - U_m| = |C_1\oplus 2\mathsf{NMExt}(X_i\circ N_i, X_{i^*}\circ N_{i^*}) - C_1\oplus U_m| \le \varepsilon.$$

Then using the similar argument as above, we have $|\mathsf{MNMExt}(X_1,\ldots,X_s) - U_m| \le 3\varepsilon$, where $(X_1,\ldots,X_s)$ are random variables uniformly sampled from B'_0. ■

Now we combine all the D^i_f together to get a distribution D_f as follows.

$\underline{D_f}$: Sample $(x_1,\ldots,x_s)$ uniformly from $\{0,1\}^{ns}$. If $(x_1,\ldots,x_s)$ in B'_i, outputs D^i_f. Otherwise, outputs 0.

Then, it holds that

$$\begin{aligned}
& |\mathsf{MNMExt}(X_1,\ldots,X_s), \mathsf{MNMExt}(f(X_1,\ldots,X_s)) - U_m, \mathsf{copy}(D_f, U_m)| \\
< \;& \Pr[(X_1,\ldots,X_s) \notin \cup_{i=0}^{s} B'_i] + \sum_{i=0}^{s} \Pr[(X_1,\ldots,X_s) \in B'_i] 3\varepsilon \\
\leq \;& 3\varepsilon + (s+1)2^{k'-n},
\end{aligned}$$

where $X_1,\ldots,X_s$ are uniformly samped from $\{0,1\}^n$ and $k' = k + \log(s^2 n) + \log(\varepsilon)$.

■