Multi-Source Non-Malleable Extractors and Applications

Vipul Goyal[1], Akshayaram Srinivasan[2], and Chenzhi Zhu[3]

[1] CMU and NTT Research.
Email:vipul@cmu.edu.
[2] Tata Institute of Fundamental Research
akshayaram.srinivasan@tifr.res.in
[3] Tsinghua University
mrbrtpt@gmail.com

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 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 $\varOmega(n)$ 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 $t = \varOmega(p)$ 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:

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

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 (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*

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

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. *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 $\mathsf{MNMExt} : (\{0,1\}^n)^s \rightarrow \{0,1\}^m$ 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 2. *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.

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 3. *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. *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].

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 4. *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 $\mathsf{MNMExt} : (\{0,1\}^n)^s \rightarrow \{0,1\}^m$ 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:

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

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 detailed proof in the full version of the paper) that this construction is indeed secure against cover-free tampering if the underlying non-malleable extractor is multi-tamperable[4] and is symmetric (meaning that $\mathsf{2NMExt}(x, y) = \mathsf{2NMExt}(y, x)$ for every x, y). However, a major problem 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

[4] 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.

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 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 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 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 $j \in [\lambda]$)[5] 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.

[5] 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.

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 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

$$|\mathsf{2NMExt}(X, Y), \{\mathsf{2NMExt}(f_i(X), g_i(Y))\}_{i\in[t]} - U_m, \mathsf{copy}^{(t)}(D_F, U_m)| < \epsilon \quad (1)$$

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

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

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 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].

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.[6]

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 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.

[6] 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.

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 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 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.

We give the standard definitions and results about min-entropy, seeded and seedless extractors and non-malleable codes in the full version of this paper.

3.1 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 2 ((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*

$$|\mathsf{2NMExt}(X, Y), \{\mathsf{2NMExt}(f_i(X), g_i(Y))\}_{i \in [t]} - U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}}, U_m)| < \epsilon$$

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* $z_i = \begin{cases} x_i & \textit{if } x_i \neq \mathsf{same}^* \\ y & \textit{if } x_i = \mathsf{same}^* \end{cases}$.

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

Theorem 5 ([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* $\mathsf{2NMExt} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^{n^{\Omega(1)}}$ *at min-entropy $n - n^{\gamma}$ with error $2^{-n^{\gamma}}$.*

Theorem 6 ([Li17b]). *For any $n > 0$, there exists a constant γ such that there exists a non-malleable 2-source extractor* $\mathsf{NMExt} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^m$ *with min-entropy $(1-\gamma)n$, $m = \Omega(k)$ and error $\epsilon = 2^{-\Omega(n/\log(n))}$.*

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.

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

Definition 3 (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* $\mathsf{AdvBC} : \{0,1\}^w \times \{0,1\}^l \times \{0,1\}^a \rightarrow \{0,1\}^m$ *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*

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

Theorem 7 ([CGL16]). *For all integers ℓ, w, a, t and for any $\epsilon \in (0,1)$ such that*

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

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

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

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

4.1 Transformation

Building blocks and parameters

1. Let $\mathsf{NMExt} : \{0,1\}^{d_1} \times \{0,1\}^{d_1} \rightarrow \{0,1\}^{l_1}$ be a non-malleable 2-source extractor with min-entropy $d_1 - \Delta$ and error ϵ, where $l_1 = \Omega(\log(1/\epsilon))$.

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 $l_2 = \Omega(\log(1/\epsilon))$.
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 $n - \Delta - (1+t)(d_1 + l_2) - \log(1/\epsilon)$ and that for the second source is $d_2 - \Delta - (1+t)(d_1 + l_2) - \log(1/\epsilon)$.
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 $d_3 - \Delta - (1+t)(d_1 + l_2 + d_2) - \log(1/\epsilon)$, 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 8. *In the above construction,* NMExt' *is a t-non-malleable 2-source extractor with min-entropy $n - \Delta$ and error $O(t\sqrt{\epsilon})$.*

With the proper instantiation, we get the following corollary. The proof of the above theorem and the instantiation are presented in the full version of this paper.

Corollary 2. *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 2 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 4 (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*

$$|\mathsf{2SLNMExt}(X,Y), \{\mathsf{2SLNMExt}(f_i(X), g_i(h(X),Y))\}_{i\in[t]} \\ -U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}, h}, U_m)| < \epsilon$$

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 5.3, we give a variant of our extractor that is additionally preimage sampleable.

5.1 Leakage-Resilient Non-Malleable Extractors

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

Definition 5 (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*

$$|\mathsf{2NMExt}(X,Y), \{\mathsf{2NMExt}(f_i(X, h(Y)), g_i(Y))\}_{i\in[t]} \\ -U_m, \mathsf{copy}^{(t)}(D_{\overrightarrow{f}, \overrightarrow{g}, h}, U_m)| < \epsilon$$

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 1 ([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 $n \geq k' \geq k + \mu + \log 1/\epsilon$.*

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 $(n_1' - \Delta - p(n_2) - \log(1/\epsilon), n_2' - \Delta - \log(1/\epsilon))$ 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 $n_1 - n_1' - \Delta - p(n_2) - \log(1/\epsilon)$ 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 $n_2 - n_2' - \Delta - \mu - 2\log(1/\epsilon)$ and error ϵ. By Lemma 1, we infer that $\mathsf{2NMExt}$ is μ-leakage-resilient for min-entropy $n_2 - n_2' - \Delta - \log(1/\epsilon)$ 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 9. *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.*

With the proper instantiation, we get the following corollary. The proof of the above theorem and the instantiation are presented in the full version of this paper.

Corollary 3. *For any polynomial p and constant t, there exists constants γ, $n_0 > 0$ 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 $n_1 = 4n_2 + 2p(n_2)$.*

5.3 Efficient Pre-image Sampleability

We also get a construction of strong leakage-resilient non-malleable extractor with *efficient pre-image sampleability* and we obtain the following corollary. See the full version of this paper for more details.

Corollary 4. *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 $m = n_2^{\Omega(1)}$.*

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 (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*

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

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

$$\mathsf{copy}(x,y) = \begin{cases} x & \textit{if } x \neq \mathsf{same}^* \\ y & \textit{if } x = \mathsf{same}^* \end{cases}.$$

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 4). 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 $\epsilon < 1/2^m$.

Construction 2. 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 10. *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* $n - \Delta + \log(1/\epsilon)$ *and error* $O(s(\epsilon + s2^{-m}))$*.*

The proof of the above theorem is given in the full version of this paper.

6.3 Instantiation

We now instantiate construction 3 with the strong leakage-resilient non-malleable extractors from section 4.1.

From Corollary 3, 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 $n_1 = 4n_2 + 2p(n_2)$, $m = n_2^{\Omega(1)}$, $\Delta = \gamma n_2$, $\epsilon = 2^{-n_2^{\Omega(1)}}$ for some constant γ. We can assume $m < \log 1/\epsilon$ 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 $\Delta > 2 \log 1/\epsilon$ by enlarging ϵ.

Let $n = (2s^2 + s + 4)n_2$ and $\gamma' = \gamma/(2s^2 + s + 4)$. From theorem 10, 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 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.

Instantiation We now instantiate $\mathsf{2SLNMExt}$ from section 5.3. Recall that this extractor has efficient pre-image sampleability.

From Corollary 4, by setting $p(n_2) = s^2 n_2$, we get could a p-strong leakage-resilient $(2,s)$-non-malleable extractor $\mathsf{2SLNMExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \rightarrow \{0,1\}^m$ with min-entropy (n_1, n_2) and error ϵ, where $n_1 = 4n_2 + p(n_2)$, $m = n_2^{\Omega(1)}$, $\epsilon = 2^{-n_2^{\Omega(1)}}$ and $s < n_2^{\gamma}$ for some constant γ. We assume $m < \log 1/\epsilon$ 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 10, 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. *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 (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*

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

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 11 ([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* $(\mathsf{MNMEnc}, \mathsf{MNMDec})$ *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* $\epsilon(2^{(m+1)} + 1)$.

The proof of the above theorem and the instantiation are presented in the full version of this paper. With the instantiation, we get the following corollary.

Corollary 7. *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. We give the definition of NMSS and t-cover-free tampering in the full version of this paper.

8.1 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 $\epsilon < 1/2^{3m}$.[7]

Construction 3. 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 $i \in [n]$, $(\mathsf{R}^{(i)}_1, \ldots, \mathsf{R}^{(i)}_n) \leftarrow \mathsf{Share}(\mathsf{R}^{(i)})$.
  4. Set $\mathsf{share}_i = (\mathsf{L}^{(i)}, \mathsf{R}^{(1)}_i, \ldots, \mathsf{R}^{(n)}_i)$.
  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 $(\mathsf{L}^{(i)}, \mathsf{R}^{(1)}_i, \ldots, \mathsf{R}^{(n)}_i)$.
     (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 $s := \mathsf{Rec}(\mathsf{Sh}_{i_1}, \ldots, \mathsf{Sh}_{i_t})$.

Theorem 12. *For any $t \geq 2$, $(\mathsf{NMShare}, \mathsf{NMRec})$ described above is a $(t, n, 0, 0)$ secret sharing scheme that is $O(n(\epsilon \cdot 2^{3m} + t2^{-m}))$-non-malleable against $\mathcal{F}_{t-\mathsf{cover-free}}$.*

The proof of the above theorem and the instantiation of the protocol are presented in the full version of this paper. With the instantiation, we get the following corollary.

Corollary 8. *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 $2^{-m^{\Omega(1)}}$.*

[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 $\mathsf{2SLNMExt}(L^*, R_s) = s$.

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 8 (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*

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

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. *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 10 ((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 $D_{\vec{f}}$ with support on $(\{0,1\}^m)^t$ which is independent of the random variables $X_1, \ldots, X_s$, such that*

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

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

We show in the full version of this paperthat 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 $x_1^{(i)}, \dots, x_p^{(i)}$.
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 $\{i_1, \dots, i_{s-1}\} \subseteq [p]$ of size $s - 1$,
   (a) Compute $y_{i_1,\dots,i_{s-1}} = \mathsf{MNMExt}(x_i^{(i)}, x_i^{(i_1)}, \dots, x_i^{(i_{s-1})})$.
5. Remove the duplicates from the sequence $(y_{i_1,\dots,i_{s-1}})_{i_1,\dots,i_{s-1}}$ to get $y'_1, \dots, y'_k$.
6. Output $z_i = y'_1 \oplus \dots \oplus y'_k$.

Theorem 13. *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* $(t, g, 2g \cdot \epsilon)$ *network extractor protocol for min-entropy* $n - \Delta + \log(1/\epsilon)$*. When* $s = O(1)$*, the running time of the protocol is* $\mathrm{poly}(n, p)$*.*

The proof of the above theorem and the instantiation are presented in the full version of this paper. With the instantiation, we get the following corollary.

Corollary 9. *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.*

Acknowledgements. We thank the anonymous reviewers of Eurocrypt 2021 for useful comments on our manuscript. The first author was supported in part by NSF grant 1916939, a gift from Ripple, a JP Morgan Faculty Fellowship, and a Cylab seed funding award. Work partially done while the second author was 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 work was partially done while the second and third authors were visiting CMU. The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies.

References

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.

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.

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 An-*

nual 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.

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.