Improved Computational Extractors and their Applications

Dakshita Khurana[*] Akshayaram Srinivasan[†]

Abstract

Recent exciting breakthroughs, starting with the work of Chattopadhyay and Zuckerman (STOC 2016) have achieved the first two-source extractors that operate in the low min-entropy regime. Unfortunately, these constructions suffer from non-negligible error, and reducing the error to negligible remains an important open problem. In recent work, Garg, Kalai, and Khurana (GKK, Eurocrypt 2020) investigated a meaningful relaxation of this problem to the computational setting, in the presence of a common random string (CRS). In this relaxed model, their work built explicit two-source extractors for a restricted class of unbalanced sources with min-entropy n^γ (for some constant γ) and negligible error, under the sub-exponential DDH assumption.

In this work, we investigate whether computational extractors in the CRS model be applied to more challenging environments. Specifically, we study network extractor protocols (Kalai et al., FOCS 2008) and extractors for adversarial sources (Chattopadhyay et al., STOC 2020) in the CRS model. We observe that these settings require extractors that work well for balanced sources, making the GKK results inapplicable. We remedy this situation by obtaining the following results, all of which are in the CRS model and assume the sub-exponential hardness of DDH.

- We obtain "optimal" computational two-source and non-malleable extractors for balanced sources: requiring both sources to have only poly-logarithmic min-entropy, and achieving negligible error. To obtain this result, we perform a tighter and arguably simpler analysis of the GKK extractor.
- We obtain a single-round network extractor protocol for poly-logarithmic min-entropy sources that tolerates an optimal number of adversarial corruptions. Prior work in the information-theoretic setting required sources with high min-entropy rates, and in the computational setting had round complexity that grew with the number of parties, required sources with linear min-entropy, and relied on exponential hardness (albeit without a CRS).
- We obtain an "optimal" *adversarial source extractor* for poly-logarithmic min-entropy sources, where the number of honest sources is only 2 and each corrupted source can depend on either one of the honest sources. Prior work in the information-theoretic setting had to assume a large number of honest sources.

1 Introduction

Randomness is fundamental in the design of algorithms and cryptographic systems. For many problems (such as Polynomial Identity Testing), the fastest known algorithms use randomness. The role of randomness is more pronounced in the design of cryptographic systems such as bit commitment, encryption, etc., as one needs unbiased random bits to achieve security [DOPS04].

Most sources of randomness found in nature are not perfect. The amount of randomness in a source is usually formalized via the notion of min-entropy. The min-entropy of a random source X is defined as the $\max_{x \in \mathsf{Supp}(X)} \log 1/\Pr[X = x]$. A natural, fundamental question is: Can we extract uniform random bits out of these weak sources? The answer is: Yes, and this is achieved by a tool called as randomness

[*] University of Illinois Urbana-Champaign, USA. Email: dakshita@illinois.edu

[†] Tata Institute of Fundamental Research, India. Email: akshayaram.srinivasan@tifr.res.in

extractors. However, it is well-known that it is impossible to extract uniform random bits given only a single weak source. To side step this impossibility, two notions have been considered. One is the seeded setting where you assume the existence of a uniform short seed that is independent of the weak source. The other setting is the independence source setting. The independence setting is weaker than the seeded setting as it only needs indpendent sources $X_1, \ldots, X_p$ such that each have sufficient min-entropy. In this work, we are interested in the independent source setting.

Independent Source Extractor. Starting with the seminal work of Chor and Goldreich [CG88], there has been a long line of work on constructing better independent source extractors.[1] A recent breakthrough work of Chattopadhyay and Zuckerman [CZ16] gave a construction of two-source extractor for poly logarithmic min-entropy sources. However, the error of the extractor was inverse polynomial. Even though the subsequent works [Li16, Coh16a, Coh16b, Coh16c, Coh16d, Li17, BADTS16] improved the min-entropy of the sources to nearly logarithmic, none of these works achieved negligible error (which is important for cryptographic applications).

A recent work of Garg, Kalai, and Khurana [GKK20] considered the problem of constructing two-source computational extractors with negligible error. They additionally assumed the existence of a common random string that is sampled once and for all, and the weak sources can depend on the CRS. This precludes constructions where the common random string can be used as a seed to extract uniform random bits from these weak sources. They provided a construction of a computational two-source extractor with negligible error in the CRS model for sources with min-entropy $\Omega(n^\gamma)$ (for some constant $\gamma \in (0, 1)$) under the sub-exponential hardness of the DDH assumption.

Challenges. The independent source setting makes two crucial assumptions. First, it assumes that each of the sources $X_1, \ldots, X_p$ are independently generated. Second, it assumes that each of these sources have sufficient min-entropy. However, neither of these assumptions may be true in general for many sources found in nature. For instance, it could be possible that one or more of these weak sources are biased and have little or no min-entropy. It could also be the case that some of these sources are adversarially corrupted so as to introduce a dependence between them. Hence, it is only safe to assume that some of these sources have sufficient min-entropy and are independent whereas other sources may have low min-entropy and may also depend on these honest sources. The main challenge is that we do not know a-priori which sources are honest and which ones are corrupted.

Can we nevertheless construct an extractor that outputs uniform random bits given a sample from such sources?

This question is not new and has already been previously investigated in two types of contexts: network extractor protocols [DO03, GSV05, KLRZ08, KLR09] and extractors for adversarial sources [CGGL20].

Network Extractor Protocols. Consider a setting where there are multiple parties and each party has an independent weak random source. The parties want to communicate with each other over a public channel and at the end of the protocol, each party outputs uniform random bits. These random bits could be used to run a distributed computation protocol or for securely computing a multiparty functionality. The challenge, however, is that some of these parties may be corrupted by a malicious adversary that may instruct them to deviate arbitrarily from the protocol. Can honest parties still end up with uniform random bits under such an adversarial attack? This is precisely what is achieved by a network extractor protocol [DO03, GSV05, KLRZ08, KLR09].

Here, the key barrier is that adversarial messages may be derived from sources that have little or no min-entropy and furthermore, these messages may depend on the messages from the honest parties. In

[1] The quality of an independent source extractor is determined by three parameters, (i) the number of independent sources, (ii) the min-entropy of these sources, and (iii) the error which is the statistical distance between the output of the extractor and the uniform distribution.

the information-theoretic setting, the work of Kalai et al. [KLRZ08] provided constructions of network extractor protocol for sources that have min-entropy of $2^{\log^{\beta} n}$ (for some constant $\beta < 1$). However, the main drawback is that they could guarantee that only a fraction of the honest parties end up with uniform random bits. In a recent work, Goyal et al. [GSZ21] gave a protocol that did not have this limitation, but their protocol only worked in a setting where the min-entropy of the sources was very high. Specifically, they required that for any p number of parties, there exists a constant γ such that min-entropy is $n(1-\gamma)$. In the computational setting, the work of Kalai et al. [KLR09] gave a protocol for sources with min-entropy $\Omega(n)$ but relied on exponential hardness of one-way permutations and the round complexity of the protocol grew with the number of parties.

Extractors for Adversarial Sources. In this setting, we consider a distribution of p sources $(X_1, \ldots, X_p)$ where some them are guaranteed to be independent and have sufficient min-entropy (called honest sources) and the others are adversarially generated and could depend on the honest sources in some limited ways (called corrupt sources). Given a sample from this distribution, we need to extract bits that are close to the uniform distribution. Of course, the main challenge here is that we do not know apriori which sources are honest and which sources are corrupt and how the corrupt sources depend on the honest sources. The work of Chattopadhyay et al. [CGGL20] formally studied this primitive[2] and gave constructions (in the information-theoretic setting) where the number of honest sources K is at least $p^{1-\gamma}$ (for some contant γ), their min-entropy is poly logarithmic and each corrupted source could depend on at most K^{γ} honest sources.

Our Work. We continue the line of work initiated by Garg et al. [GKK20] on constructing computational extractors in the CRS model and provide new constructions that extract uniform bits in the setting of network extractors and from adversarial sources.

1.1 Our Results

The key technical tool that allows us to obtain the above applications is a *better* analysis of the GKK computational two-source extractor in the CRS model.

The GKK extractor as analyzed in [GKK20] had two drawbacks: first, it required sources that have min-entropy of $\Omega(n^{\gamma})$ (for some constant $\gamma \in (0,1)$) and second, it worked only for sources that were heavily imbalanced: requiring that one of the sources have entropy equal to the size of the other source.

Our first result is a much cleaner analysis of this construction. Our improved analysis essentially shows, somewhat surprisingly, that the extractor from [GKK20] actually does not suffer from either of the limitations stated above. That is, it works for *balanced* sources that are each only required to have *poly logarithmic* min-entropy, and achieves negligible error.

Informal Theorem 1. *Let λ denote the security parameter. Assuming the sub-exponential hardness of DDH, there exists a constant $c > 1$ such that for any $\lambda \leqslant n_1, n_2 \leqslant \mathrm{poly}(\lambda)$, there exists a construction of a negligible error, two-source computational extractor in the CRS model where sources have lengths n_1, n_2 respectively and min-entropy $O(\log^c n)$.*

Our tighter analysis is also arguably *simpler* than the one in [GKK20]. As a corollary, we use the transformation from [GKK20] to obtain a construction of a negligible-error, *non-malleable* two-source extractor for balanced sources with polylogarithmic min-entropy, where one source can be tampered an arbitrary polynomial number of times (this is called a one-sided non-malleable extractor). Specifically,

[2] In a work that is concurrent and independent to Chattopadhyay et al., Aggarwal et al. [AOR+20b] studied another model of adversarial sources called SHELA sources. They showed that it is impossible to extract uniform random bits from SHELA sources and gave constructions of extractors whose output is somewhere random. In another work, Dodis et al. [DVW20] studied a notion of extractor dependent sources which arise in the setting where the source sampler could depend on the output of the previous invocations of the extractor using the same seed.

in the one-sided setting, the adversary gets access to a tampering oracle and can specify any efficiently computable tampering function on one of the sources. The oracle responds with the output of the extractor computed on the first source and the tampered second source.

Informal Theorem 2. *Let λ denote the security parameter. Assuming the sub-exponential hardness of the DDH assumption, there exists a constant $c > 1$ such that for any $\lambda \leqslant n_1, n_2 \leqslant \mathsf{poly}(\lambda)$, there exists a construction of a negligible error, two-source, one-sided computational non-malleable extractor in the CRS model where both sources have lengths n_1, n_2 respectively and have min-entropy $O(\log^c n)$.*

We then use the above non-malleable extractor as the main building block and give a construction of network extractor protocol that has a single round of communication, works with poly logarithmic min-entropy sources and can tolerate an optimum number of malicious corruptions.

Informal Theorem 3. *Let λ be the security parameter. Assuming sub-exponential hardness of the DDH assumption, there exists a constant $c > 1$ s.t. for any $\lambda \leqslant n \leqslant \mathsf{poly}(\lambda)$, there exists a construction of a single round, negligible error, computational network extractor protocol in the CRS model for any p (which is a polynomial in the security parameter) number of parties each having an independent source of length n and min-entropy $O(\log^c n)$. The protocol tolerates $p - 2$ corruptions by a malicious adversary (which is optimum). Furthermore, all the honest parties end up with an output that is computationally indistinguishable to the uniform distribution given the view of the adversary.*

We also give a construction of an adversarial source extractor that works in the extreme setting where there are only two honest sources and every corrupted source can depend on either one of the honest sources. This construction uses our computational two-source extractor as the main building block.

Informal Theorem 4. *Let $p \in \mathbb{N}$ be fixed and let λ be the security parameter. Assuming that sub-exponential hardness of DDH assumption, there exists some constant $c > 1$ s.t. for $\Omega(\lambda) \leqslant n \leqslant \mathsf{poly}(\lambda)$, there exists a construction of negligible error adversarial source extractor in the CRS model that works for an arbitrary adversarial p-source distribution where (i) each source has length n, (ii) there are two honest independent sources with min-entropy $O(\log^c \lambda)$, and (iii) every other source is the output of an (efficient) function of either one of the two honest sources.*

Comparison with [AOR$^+$20a]. We now compare our results with the prior work of Aggarwal et al. [AOR$^+$20a]. While both papers build on [GKK20] and obtain new types of computational non-malleable extractors, there are some important differences in the results. In the setting where only one of the sources is tamperable and the number of tamperings is unbounded,

- Techniques in [AOR$^+$20a] give non-malleable extractors for linear min-entropy (min-entropy greater than 0.46n) based on quasi-polynomial DDH. To achieve poly-logarithmic min-entropy, they additionally assume the existence of near optimal (exponentially hard) collision-resistant hash functions.

- Our work gives a construction for poly-logarithmic min-entropy based on sub-exponential DDH.

We remark that [AOR$^+$20a] also (primarily) considers a setting where both sources can be tampered but the number of tamperings is bounded. Among other results, they provide new constructions in this setting for linear min-entropy (min-entropy greater than 0.46n) based on quasi-polynomial DDH and for poly-logarithmic min-entropy based on near-optimal (exponential) hardness of collision-resistant hash functions.

An important objective of our work is to achieve new applications: these applications require a setting where the number of tamperings is unbounded, with only one source being tampered. For this setting, as discussed above, our work shows that the [GKK20] construction achieves poly-logarithmic min-entropy for balanced sources from sub-exponential DDH.

2 Technical Overview

In this section, we provide an overview of our results.

2.1 Improved Two-Source and Non-Malleable Extractors

We start with an overview of our improved two-source and non-malleable extractors. The key technical bulk of this part of our work is an improved two-source extractor, and plugging in the resulting extractor into the work of [GKK20] also immediately yields an improved non-malleable extractor, as we will discuss below.

2.1.1 Background: The Blueprints of [BHK11, BACD$^+$17, GKK20].

As a first step, we recall the construction of two-source extractors in [GKK20], which itself combines the blueprint of [BHK11] with that of [BACD$^+$17]. As discussed above, we will show that essentially the same construction serves as a strong computational extractor even for *balanced sources*, and even in settings where sources have only *polylogarithmic min-entropy*. In contrast, the techniques in [GKK20] limited them to highly unbalanced sources and required λ^ϵ min-entropy

At a high level, [GKK20] obtain two-source extractors with low error via two steps.

Step 1. Following a blueprint suggested in [BHK11], [GKK20] build a computational non-malleable extractor in the CRS model, in a setting where one of the sources has min entropy rate larger than $1/2$. We use the same blueprint in this work also, and therefore we describe it below.

First, start with any 2-source extractor

$$\mathsf{2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m,$$

with negligible error (eg., [Bou05, Raz05]), min-entropy $(\text{poly} \log n_1)$ for one of the sources and min-entropy rate slightly larger than $1/2$ for the other.

The construction makes use of the following cryptographic primitives, which can be obtained based on the (sub-exponential) hardness of DDH.

1. A collision resistant function family $\mathcal{H}$, where for each $h \in \mathcal{H}$, $h : \{0,1\}^{n_2} \to \{0,1\}^k$, where k is significantly smaller than the min-entropy of the second source of 2Ext.
2. A family of lossy functions $\mathcal{F}$, where for each $f \in \mathcal{F}$, $f : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$. A lossy function family consist of two types of functions: injective and lossy. Each lossy function loses most of the information about the input (i.e., image size is very small). It is hard to distinguish between a random injective and a random lossy function in the family.

The actual construction is as follows. The CRS consists of a random function $h \leftarrow \mathcal{H}$ from the collision-resistant hash family, and consists of $2k$ random functions from family $\mathcal{F}$, denoted by

$$\begin{array}{c} f_{1,0}, f_{2,0}, \ldots, f_{k,0} \\ f_{1,1}, f_{2,1}, \ldots, f_{k,1} \end{array}$$

where for a randomly sampled $b \leftarrow \{0,1\}^k$, for all $i \in [k]$, f_{i,b_i} are injective, and $f_{i,1-b_i}$ are lossy.

The computational non-malleable extractor (in the CRS model) is defined by

$$\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}) := \mathsf{2Ext}(f_{\mathsf{crs},h(y)}(x), y),$$

where

$$f_{\mathsf{crs},s}(x) := f_{1,s_1} \circ \ldots \circ f_{k,s_k}(x)$$

Consider any polynomial size adversary $\mathcal{A}$ that obtains either $(\textsf{cnm-Ext}(x,y),y,\textsf{crs})$ or $(U,y,\textsf{crs})$, together with an oracle $\mathcal{O}$ that has $(x,y,\textsf{crs})$ hardwired, and on input y' outputs $\perp$ if $y'=y$, and otherwise outputs $\textsf{cnm-Ext}(x,y',\textsf{crs})$. By the collision resistance property of h, $\mathcal{A}$ queries the oracle on input y' s.t. $h(y')=h(y)$ only with negligible probability. Therefore, the oracle $\mathcal{O}$ can be replaced by a different oracle, that only hardwires $(\textsf{crs},h(y),x)$ and on input y' outputs $\perp$ if $h(y')=h(y)$, and otherwise outputs $\textsf{cnm-Ext}(x,y')$.

It is observed in [BHK11, GKK20] that access to this oracle can be simulated entirely given only crs, $h(y)$ and $(Z_1,\ldots Z_k)$, where for every i, $Z_i=f_{1,1-h(y)_1}(f_{2,1-h(y)_2}(\ldots f_{i,1-h(y)_i},(\ldots f_{k,h(y)_k}(x)))$. Now suppose that the functions $\{f_{i,1-h(y)_i}\}_{i\in[k]}$ were all lossy – then it is easy to see that (for small enough k), Y has high min-entropy conditioned on $h(y)$ and $Z=(Z_1,\ldots,Z_k)$. At the same time, as long as the functions $\{f_{i,h(y)_i}\}_{i\in[k]}$ are all injective, the output $f_{\textsf{crs},h(y)}(x)$ continues to have high entropy conditioned on $h(y)$ and Z. Then one could use the fact that 2Ext is a (strong) 2-source extractor, to argue that the output of our non-malleable extractor is close to uniform.

Moreover, since the adversary $\mathcal{A}$ cannot distinguish between random injective functions and random lossy ones, it should be possible to (indistinguishably) change the CRS to ensure that functions $f_{1,h(y)_1},\ldots,f_{k,h(y)_k}$ are injective, whereas the functions $f_{1,1-h(y)_1},\ldots,f_{k,1-h(y)_k}$ are all lossy.

This intuition is converted into a formal proof by [BHK11, GKK20]. In summary, these works show that the resulting non-malleable extractor (very roughly) inherits the entropy requirements of the underlying two-source extractor. Moreover, the resulting extractor is non-malleable w.r.t. *arbitrarily many* tampering functions (this is impossible to achieve information theoretically).

Looking ahead, the analysis in [BHK11, GKK20] appears to be fairly tight, and is *not* why [GKK20] are limited to unbalanced sources and λ^ϵ min-entropy. These restrictions appear to be a result of the next transformation, which converts non-malleable extractors with high entropy for one source, to two-source extractos with low min-entropy for both sources. We describe this next.

Step 2. Next, [GKK20] convert the resulting non-malleable extractor (for a setting where one source has high min-entropy rate) to a two-source extractor for a setting where both sources have low min-entropy, by following a blueprint of [BACD+17].

An important difference between [BACD+17] and [GKK20] is that the reduction in [BACD+17] is not efficient: specifically, even given an efficient adversary that contradicts the security of the 2-source extractor, [BACD+17] obtain an *inefficient* adversary that contradicts security of the underlying non-malleable extractor.

To better understand this issue, we briefly summarize the transformation of [BACD+17]. Their transformation uses a disperser as a building block.

A (K,K') disperser is a function

$$\Gamma:\{0,1\}^{n_2}\times[t]\to\{0,1\}^d$$

such that for every subset A of $\{0,1\}^{n_2}$ that is of size $\geqslant K$, it holds that the size of the set of neighbors of A under Γ is at least K'.

The [BACD+17]-transformation starts with a seeded non-malleable extractor nm-Ext : $\{0,1\}^{n_1}\times\{0,1\}^d\to\{0,1\}^m$ and a disperser $\Gamma:\{0,1\}^{n_2}\times[t]\to\{0,1\}^d$, and constructs the following 2-source extractor $\textsf{2Ext}:\{0,1\}^{n_1}\times\{0,1\}^{n_2}\to\{0,1\}^m$, defined by

$$\textsf{2Ext}(x_1,x_2)=\bigoplus_{y:\exists i\text{ s.t. }\Gamma(x_2,i)=y}\text{nm-Ext}(x_1,y)$$

Intuitively, by the definition of an (information-theoretic) t-non-malleable extractor nm-Ext, for a random $y\in\{0,1\}^d$, for all $y'_1,\ldots,y'_t$ that are distinct from y, it holds that

$$\big(\text{nm-Ext}(X_1,y),\text{nm-Ext}(X_1,y'_1),\ldots,\text{nm-Ext}(X_1,y'_t)\big)\equiv$$

$$\big(U,\text{nm-Ext}(X_1,y'_1),\ldots,\text{nm-Ext}(X_1,y'_t)\big)\,.$$

This means that for "most" y, nm-Ext(X_1, y) is stastistically close to uniform, even given nm-Ext$(X_1, \Gamma(x_2, j))$ for every $j \in [t]\backslash\{i\}$ such that $\Gamma(x_2, j) \neq y$, which in turn implies that the XOR of these (distinct) values is close to uniform, which implies that 2Ext(X_1, x_2) is close to uniform.

But to formally prove that the resulting extractor is a strong (information-theoretic) non-malleable extractor, one would need to construct a reduction $\mathcal{R}$ that breaks the non-malleable extractor, given any adversary $\mathcal{A}$ that breaks the two-source extractor. In the computational setting, $\mathcal{R}$ is required to be efficient, which causes the bulk of the technical difficulty in [GKK20].

In more detail, $\mathcal{R}$ obtains input $(\alpha, \widehat{y})$, where $\widehat{y}$ is a random seed for the non-malleable extractor and where α is either chosen according to cnm-Ext$(X_1, \widehat{y})$ or is chosen uniformly at random. In addition, $\mathcal{R}$ obtains an oracle that outputs cnm-Ext(X_1, y') on input $y' \neq \widehat{y}$. $\mathcal{R}$ must *efficiently* distinguish between the case where $\alpha \leftarrow$ cnm-Ext$(X_1, \widehat{y})$ and the case where α is chosen uniformly at random. In order to use the (two-source extractor adversary) $\mathcal{A}$, $\mathcal{R}$ needs to generate a challenge for $\mathcal{A}$ that corresponds either to the output of the 2-source extractor (if α was the output of cnm-Ext) or uniform (if α was uniform). In addition, the reduction $\mathcal{R}$ must generate a corresponding x_2 for $\mathcal{A}$, that is sampled according to X_2. This is easy to do in unbounded time by simply sampling $x_2 \leftarrow X_2$ conditioned on the existence of i such that $\Gamma(x_2, i) = y$.

To enable a reduction in the computational setting, [GKK20] view the inefficient computation involved; i.e. sampling $x_2 \leftarrow X_2$ conditioned on the existence of i such that $\Gamma(x_2, i) = y$; as the output of a leakage function. Unfortunately, this means that the running time of the reduction grows as $2^{|x_2|}$, which restricts $|x_2|$ to being extremely small, in fact much smaller than the size of the first source. This also restricts the sources in such a way that the min-entropy in the first source is required to be larger than the size of the second source. As discussed above, the highly asymmetric state of affairs does not bode well for many natural applications of two-source and non-malleable extractors.

2.1.2 Our Key Ideas.

To remedy this situation, we develop a completely different analysis for essentially the same construction. In contrast with [GKK20], our analysis is arguably simpler, does not impose any asymmetric restrictions on each source, and leads to significantly improved min-entropy parameters.

First, we do not split the analysis of the resulting two-source extractor into two steps as described above. In other words, unlike [GKK20], we *do not* attempt to prove that the [BACD$^+$17] template as described in Step 2, when applied to *any computational non-malleable extractor*, yields a good two-source extractor with low min-entropy and low error.

Instead, we apply the [BHK11] transform to an *information-theoretic* two-source extractor with low error but min-entropy rate of 1/2 for one of the sources (eg., [Bou05, Raz05]). Next, we consider the [BACD$^+$17] transform applied to the result of this extractor. We then give a monolithic proof that the result of applying these transformations one after the other results in a two-source extractor for balanced sources, polylogarithmic min-entropy and negligible error.

At a very high level, this monolithic approach enables us to *strip off* all computational components one by one, to eventually end up with a purely information theoretic experiment. This allows us to sidestep the need to invert the disperser in any of our computational reductions; limiting our use of inefficient reductions to the information-theoretic step in the proof.

We now discuss our proof strategy in additional detail. We will start with an experiment where the adversary obtains either the output of the (final) two-source extractor, which we will denote by c2Ext(x_1, x_2) or a uniformly random value (in each case the adversary also obtains the sample x_2). As discussed above, we will modify this experiment in steps, slowly stripping off computational assumptions until we end up in an experiment that does not require any assumptions.

Discarding Hash Collisions. Recall that the [BHK11] blueprint uses $z = h(y)$ to choose a subset of functions f_{i,z_i} to apply to the first source. As a first step, we will modify the experiment so that if in the process of computing c2Ext(x_1, x_2), a hash collision is encountered, then we simply outputs a uniformly random sample instead of $\mathsf{c2Ext}(x_1, x_2)$. In more detail, the output of the two-source extractor c2Ext is replaced by a slightly modified $\mathsf{c2Ext}'$. The replacement $\mathsf{c2Ext}'(x_1, x_2)$ first checks if $\exists(i_1, i_2)$ such that $\Gamma(x_2, i_1) \neq \Gamma(x_2, i_2)$ but $h(\Gamma(x_2, i_1)) = h(\Gamma(x_2, i_2))$. If such (i_1, i_2) exist, then $\mathsf{c2Ext}'$ outputs a uniformly random value.

At the same time, the oracle $\mathcal{O}$ is replaced with $\mathcal{O}'$ that is identical to $\mathcal{O}$, except that on input any y' such that $h(y') = h(y)$, $\mathcal{O}'$ outputs $\perp$.

We rely on the collision resistance of the hash function family to argue that as long as the sources are efficiently sampleable, this experiment is statistically indistinguishable from the previous one. This argument will allow us to simply discard hash collisions throughout the rest of this overview. The other remaining assumption is that of the lossy function family.

Working around Lossy Functions. Recall that the approach in [GKK20] is to (indistinguishably) switch the crs so that the functions $\{f_{i,1-h(y)_i}\}_{i\in[k]}$ are all lossy, and the rest are injective. This 'nicely distributed' CRS allows them to efficiently "simulate" the output of the oracle $\mathcal{O}$, and prove that the resulting construction is a non-malleable extractor[3] But this approach runs into the barriers described above, as the eventual two-source extractors do not support balanced sources or poly-logarithmic min-entropy.

In this work, as a first stab, we attempt to make statistical arguments about the sources in an (imagined) experiment where the CRS is assumed to be 'nicely distributed'. In more detail, we say that the random variable y takes a "bad" value if it becomes possible for an oracle-aided *unbounded* adversary to distinguish the output of the [BHK11] non-malleable extractor from uniform, *when conditioned on the CRS being 'nicely distributed' for* y. That is, for a function $\epsilon = \epsilon(\lambda)$, we define the set $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$ (roughly) as the set of y, for which the following holds: conditioned on the CRS being such that functions at positions indexed by $h(y)$ are injective and the others are lossy, the output of the non-malleable extractor is at least ϵ-statistically distinguishable from a uniformly random value in presence of the oracle $\mathcal{O}'$.

Bounding $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$. We prove that for large enough (but still negligible) ϵ, the size of the set $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$ is negligibly small. Fortunately, since the definition of $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$ already conditions on the CRS being nicely distributed, this argument does not involve any computational assumptions, and follows by a reduction to the underlying *information-theoretic* two-source extractor of [Bou05, Raz05], as long as the number of tampering queries is polynomially bounded. Intuitively, conditioned on the CRS being nice, we can establish that the sources (for the non-malleable extractor) retain high entropy even in the presence of the oracle $\mathcal{O}'$, and therefore, the output of the two-source extractor, applied to $(f_{\mathsf{crs},h(y)}(x), y)$ is statistically indistinguishable from uniform. Then a simple averaging argument allows us to prove that $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$ is small.

From non-malleable to two-source extractors. Next, we aim to use the definition of $\mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$ to derive a meaningful (statistical) conclusion about the final two-source extractor. Specifically, we begin by fixing a (large enough, but still negligible) ϵ.

We consider a game that samples sources (x_1, x_2) for the final two-source extractor, and samples $i \leftarrow [t]$, *conditioned on* $y = \Gamma(x_2, i)$ lying outside the set $\mathsf{BAD\text{-}seed}_{\epsilon,X_1}$. By definition of the set $\mathsf{BAD\text{-}seed}_{\epsilon,X_1}$, for any y outside this set, when the CRS is such that the functions indexed by $h(y)$ are injective and others are lossy, the output of the non-malleable extractor is *statistically* indistinguishable from uniform, even given (polynomial-query) access to the tampering oracle. Recall that the output of the two-source extractor is

$$\mathsf{2Ext}(x_1, x_2) = \bigoplus_{y:\exists i \text{ s.t. } \Gamma(x_2,i)=y} \text{nm-Ext}(x_1, y)$$

[3] There are many other subtleties involved, most importantly, a circularity: the CRS must be programmed according to $h(y)$, but y is sampled as a function of the CRS. The work of [GKK20] develops techniques to avoid these subtleties, but we do not discuss them here as they are less relevant to the current approach.

This means that for $y \notin \mathsf{BAD\text{-}seed}_{\epsilon,X_1}$, for all $y'_1, \ldots, y'_t$ that are distinct from y, it holds that

$$\big(\text{nm-Ext}(X_1, y), \text{nm-Ext}(X_1, y'_1), \ldots, \text{nm-Ext}(X_1, y'_t)\big) \text{ and}$$

$$\big(U, \text{nm-Ext}(X_1, y'_1), \ldots, \text{nm-Ext}(X_1, y'_t)\big)$$

are at most ϵ-statistically distinguishable.

This means that for such y, nm-Ext(X_1, y) is statistically close to uniform, even given nm-Ext$(X_1, \Gamma(x_2, j))$ for every $j \in [t]\backslash\{i\}$ such that $\Gamma(x_2, j) \neq y$, which in turn implies that the XOR of these (distinct) values is close to uniform, which implies that $\mathsf{2Ext}(X_1, x_2)$ statistically is close to uniform.

Because we carefully conditioned on $y = \Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon,\mathcal{X}}$, we are able to (again, *statistically*) argue that the output of the *two-source* extractor in this game will be statistically indistinguishable from uniform, even given x_2.

At this point, we have argued that in an idealized game where the CRS is conditioned on being nicely distributed, the output of the (strong) two-source extractor will be indistinguishable from uniform. But the in the actual construction, the CRS is distributed in such a way that for a random $b \leftarrow \{0,1\}^k$ the functions $f_{i,1-b_i}$ are lossy, and the others are injective. This only very rarely matches the idealized game (where we essentially condition on $b = h(y)$). At this point, we would like to use the fact that lossy functions are indistinguishable from injective ones, to argue that the adversary cannot distinguish an actual game from the idealized game. Formalizing this intuition runs into a few subtle issues, that we briefly describe next.

The Computational Argument. Note that in the idealized game described above, (x_2, i) are sampled conditioned on:

- The crs being such that functions indexed by $\Gamma(x_2, i)$ are injective and the others are lossy, and
- $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon,X_1}$.

We begin by removing the first requirement, and moving to a game where we *only* condition on $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon,X_1}$. We prove that removing the first conditioning does not (significantly) affect a PPT distinguisher's ability to distinguish between the output of the extractor and uniform. The proof of this makes careful use of Chernoff bounds to show that if the two games are different, then one can *guess* which functions in the CRS are injective and which ones are lossy, with advantage better than what is allowed by the security of the lossy function family.

At this point, we have moved to a game where (x_2, i) are sampled only subject to the restriction that $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon,X_1}$. Next, we prove that this restriction can also be removed without (significantly) affect an unbounded distinguisher's ability to distinguish between the output of the extractor and uniform. Intuitively, this follows because of the disperser and because the set $\mathsf{BAD} - \mathsf{seed}_{\epsilon,X_1}$ is small. Recall that the disperser maps every "large enough" set of x_2's to a "large enough" set of y's. This implies that if the set of y's for which $y \in \mathsf{BAD} - \mathsf{seed}_{\epsilon,X_1}$ is small, their inverses (under the disperser) are also small. We show that as long as the source x_2 has polylogarithmic min-entropy, the probability that x_2 is such that $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon,X_1}$ for *any* i will be negligibly small.

This allows us to argue that the output of the strong two-source extractor is indistinguishable from uniform. A careful separation of the information-theoretic and computational components allows us to set parameters so that the entropy loss from the first source is only polylogarithmic. As discussed above, existing dispersers (eg., from [GUV09]) already suffice in a setting where the second source also has polylogarithmic min-entropy.

Here, we clarify that the exact min-entropy loss depends on our computational assumptions. In more detail, we assume that there exists a constant $0 < \epsilon < 1$ such that DDH with security parameter λ is hard against poly(2^{λ^ϵ})-size machines. The exact polylogarithmic min-entropy requirement on our sources then depends on ϵ.

This completes a high-level picture of our proof strategy, where we swept a few details under the rug for the sake of conceptual simplicity. We refer to Section 4 for a detailed proof.

2.1.3 From Two-Source to Non-Malleable Extractors.

Once we obtain two-source extractors as discussed above, we directly invoke a theorem from [GKK20] (that builds on the [BHK11] blueprint) to bootstrap our low entropy, low error two-source extractors to low entropy, low error *non-malleable extractors*. Since this follows almost immediately from prior work (modulo a few parameter choices), we omit details in this overview.

2.2 Network Extractor Protocol

In the network extractor setting, there are p parties and each party P_i for $i \in [p]$ has an independent weak random source X_i. There is a centralized adversary that controls an arbitrary subset $M \subset [p]$ of the parties. This adversary is malicious, which means that it can instruct the corrupted parties to deviate arbitrarily from the protocol specification and is rushing which means that in each round of the protocol, it can wait until it receives all the messages from the honest parties before sending its own message on behalf of the corrupted parties. We consider the parties to be connected via public channels and the adversary can view all the communication sent by honest parties. At the end of the protocol, we want all the honest parties to output uniform random bits that are independent of the view of the adversary.

In the computational setting, we restrict the adversary to be computationally bounded and independence mentioned above is required to hold in the computational sense. The quality of the network extractor protocol is determined by three parameters, (i) the number of corrupted parties $|M|$, (ii) the min-entropy of the weak random source available with the parties $H_\infty(X_i)$, and (iii) the number of rounds of the protocol. It is easy to observe that if $|M| = p - 1$, then we cannot construct a network extractor protocol as this task amounts to extracting uniform random bits from a single weak random source. So, the best we can hope for is the case where $|M| \leqslant p - 2$. In this work, we give a construction of network extractor protocol in the computational setting in the CRS model that tolerates $|M| \leqslant p - 2$ corruptions, runs in a single round, and works with polylogarithmic min-entropy for each source.

Key Challenge. To understand the key challenge, let us first weaken the requirements from the network extractor protocol. Let us assume for now that the first party P_1 is never corrupted but the identity of the other honest party is not known at the beginning of the protocol. Furthermore, we only require the output of honest P_1 to be uniform and independent of the view of the adversary. Can we construct a single round protocol for this weaker setting?

We observe that the techniques developed in the work of Goyal et al. [GSZ21] gives such a protocol based on any two-source non-malleable extractor. Specifically, we ask every party to send its source in the clear to the first party P_1. For every $j \neq 1$, P_1 applies the two-source non-malleable extractor on its source and the source received from P_j and outputs the XOR of all such computations. We now argue that the output of P_1 is uniform and independent of the view of the adversary if the non-malleable extractor is strong and is multi-tamperable. Let us assume that P_i for some $i \neq 1$ is the other honest party. Now, the messages sent by the adversarial parties are an efficiently computable function of P_i's source. Thus, one can view the messages from the adversarial parties as a tampering of the honest source. The security of the non-malleable extractor guarantees that the output of the extractor on the good source is close to uniform even conditioned on its output on the tampered sources. This allows us to argue that the output of P_1 is close to uniform given the view of the adversary (which includes the other honest source and that is why we require the extractor to be strong).

However, we quickly run into trouble if we want to extend this to the setting where we require the outputs of two honest parties to be uniform and independent of the view of the adversary. Indeed, if P_1 were to send its source in the clear, then we cannot use the security of the non-malleable extractor to argue that the output of P_1 is close to uniform. In the “very high" min-entropy setting, the work of [GSZ21] gave a method to overcome this barrier. Specifically, party P_i divides its source into p slices, retains the i-th slice with itself and broadcasts the rest of the slices. It now uses the i-th slice received from the other parties along with its own slice to compute the output as mentioned above. It was argued in their work that if the min-entropy source was "very high", then the outputs of the all honest parties are close to uniform and independent of the view of the adversary. However, we cannot extend their argument to the setting where the min-entropy of each weak source $\delta \cdot n$ for some universal constant δ.

Our Approach. In order to overcome this barrier, we rely on computational tools (namely, lossy functions) to artificially create independence between the messages transmitted by each party and the sources used to compute their outputs. We now elaborate on this.

For each $i \in [p]$ and $b \in \{0,1\}$, we sample $f_{i,b}$ uniformly in the injective mode and include the descriptions of these functions as part of the CRS. In the protocol, party P_i first computes $f_{i,b}(X_i)$ for each $b \in \{0,1\}$ and broadcasts $f_{i,1}(X_i)$ and retains $f_{i,0}(X_i)$ with itself. To compute the output, it evaluates the non-malleable extractor with one source as $f_{i,0}(X_i)$ and the other source as $f_{j,1}(X_j)$ for each $j \neq i$. It then outputs the XOR of these evaluations. We now show how to use the security of lossy functions to argue that the joint distribution of the outputs of the honest parties are close to uniform conditioned on the view of the adversary.

We consider a sequence of hybrids where the first hybrid in the sequence consists of the outputs of the honest parties as computed in the protocol along with the view of the adversary and last hybrid is the distribution where the outputs of all the honest parties are replaced with uniform and independent bits. In the i-th intermediate hybrid, we replace the outputs of the first i uncorrupted parties with uniform. By a standard averaging argument, it is sufficient to show that the i-th hybrid in this sequence is computationally indistinguishable to the $(i-1)$-th hybrid. Let us assume that the i-th honest party is k_i and the identity of the other honest party is k'_i.

We first consider an intermediate distribution where we sample $f_{k_i,1}$ and $f_{k'_i,0}$ in the CRS using the lossy mode instead of the injective mode. It follows from the computational indistinguishability of the injective and the lossy modes that this intermediate distribution is indistinguishable to the $(i-1)$-th hybrid. Since $f_{k_i,1}$ and $f_{k'_i,0}$ are sampled in the lossy mode, we can view these as bounded leakages from the source X_{k_i} and $X_{k'_i}$. Now, conditioned on these leakages, we can argue that $f_{k_i,0}(X_{k_i})$ and $f_{k'_i,1}(X_{k'_i})$ are independent and have sufficient min-entropy (since $f_{k_i,0}$ and $f_{k'_i,1}$ are sampled in the injective mode). Now, we can rely on the argument sketched above and view the adversarial messages as tamperings of the honest source $f_{k'_i,1}(X_{k'_i})$ and use the security of the non-malleable extractor to replace the output of P_{k_i} with uniform bits independent of the view of the adversary. To show this distribution is indistinguishable to the i-th hybrid, we again rely on the indistinguihability of the lossy and injective modes and switch sampling $f_{k_i,1}$ and $f_{k'_i,0}$ in the CRS to the injective mode. This allows us to show that the $(i-1)$-th hybrid is computationally indistinguishable to the i-th hybrid.

2.3 Extractors for Adversarial Sources

An adversarial source distribution [CGGL20] is a sequence of p random variables $(X_1, \ldots, X_p)$ such that a subset of them are independent and have sufficient min-entropy (called as the honest sources) and the rest can depend on the honest sources in a limited way (called as the corrupt sources). The goal is to construct an extractor such that given a sample from the adversarial source distribution, it outputs a string that is close to random. Here, the parameters of interest are the (i) number of honest sources in the distribution, and (ii) the min-entropy of the honest sources. We are interested in constructing extractors that work in the extreme setting where the number of honest sources is only 2 and every corrupted source is an (efficiently computable) function of either one of the honest sources.

Challenge with the Prior Approaches. The works of Chattopadhyay et al. [CGGL20] and Goyal et al. [GSZ21] gave a method of constructing such an extractor using a non-malleable extractor that satisfies an additional security property. Specifically, the adversary is allowed to specify a set of tampering functions $\{(f_i, g_i)\}_{i \in [t]}$ as well as a sequence of bits $\{b_i\}_{i \in [t]}$. If $b_i = 0$, then the adversary receives the output of the non-malleable extractor applied on $f_i(X)$ and $g_i(Y)$. Otherwise, it receives the output of the extractor on $g_i(Y)$

and $f_i(X)$. Unfortunately, we do not know how to show that the non-malleable extractor constructions in the works of [GKK20, AOR+20a] satisfy this additional property. Hence, in this work, we take new approach towards this problem that is partly inspired by our network extractor construction and relies only on a computational two-source extractor (rather than a non-malleable extractor).

Our Construction. We first explain why a network extractor protocol doesn't directly give rise to an extractor for adversarial source distribution. In the case of a network extractor protocol, only the messages sent by the corrupted parties depend on the honest party's messages whereas in the case of the adversarial sources, the corrupted source could depend on the honest source. This difference precludes a direct construction. However, we use the techniques developed for the network extractor construction to construct an extractor for adversarial sources.

Our extractor for adversarial sources is similar to our network extractor construction except that we replace the non-malleable extractor with a computational two-source extractor. Specifically, we consider p parties and provide the i-th source X_i to party P_i and run the network extractor construction described above using a two-source extractor. Once we have obtained the outputs of each of the parties, we XOR them together to output a single string. We now argue that the distribution of the output string is close to the uniform distribution.

To show this, it is sufficient to show that the output of one of the honest parties is close to uniform and is independent of the outputs of every other party. Let us assume that X_i and X_j are honest sources. We first consider an intermediate distribution where we sample $f_{k,b}$ for every $(k,b) \notin \{(i,0),(j,1)\}$ in the lossy mode. It again follows from the indistinguishability of the injective and the lossy modes that this distribution is computationally close to the original output. Now, for every corrupted source k that is derived from X_i, we can view $\{f_{k,b}(X_k)\}_{b\in\{0,1\}}$ as bounded leakage from the honest source X_i. Similarly, for every source k that is derived from X_j, we can view $\{f_{k,b}(X_k)\}_{b\in\{0,1\}}$ as bounded leakage from the honest source X_j. We can additionally leak $f_{i,1}(X_i)$ and $f_{j,0}(X_j)$. This allows us to argue that conditioned on these leakages, the sources $f_{i,0}(X_i)$ and $f_{j,1}(X_j)$ are independent and have sufficient min-entropy. We can now invoke the two-source extractor security to argue that the output of the i-th party is close to uniform even conditioned on the outputs of every other party.[4]

This completes an overview of our techniques.

Roadmap We list some preliminaries in Section 3. We recall definitions of computational extractors in Section 3.3. In Section 4 we derive theorems and corollaries for improved two-source and non-malleable extractors. Finally, in Sections 5 and 6, we describe improved constructions of network and adversarial source extractors respectively.

3 Preliminaries

In this section, we discuss some preliminaries needed for the later sections. This includes facts about min-entropy, lossy functions and dispersers. Many parts of this section are taken from [GKK20].

Definition 5. *A distribution X over a domain D is said to have min-entropy k, denoted by $H_\infty(X) = k$, if for every $z \in D$,*

$$\Pr_{x \leftarrow X}[x = z] \leqslant 2^{-k}.$$

In this paper, we consider sources with average conditional min entropy, as defined in [DORS08] (and also in the quantum information literature). This notion is less restrictive than worst case conditional

[4] The reason why two-source extractor is sufficient in this case but non-malleable extractor was needed in the previous case is that the parties here can be thought of as following the protocol whereas in the previous case, they could deviate arbitrarily from the protocol specification.

min-entropy (and therefore this strengthens our results), and is sometimes more suitable for cryptographic applications.

Definition 6. *[DORS08] Let X and Y be two distributions. The average conditional min-entropy of X conditioned on Y, denoted by $H_\infty(X|Y)$*[5] *is*

$$H_\infty(X|Y) = -\log E_{y \leftarrow Y} \max_x \Pr[X = x|Y = y] = -\log(\mathbb{E}_{y\leftarrow Y}[2^{-H_\infty(X|Y=y)}])$$

Note that $2^{-H_\infty(X|Y)}$ is the highest probability of guessing the value of the random variable X given the value of Y.

We will rely on the following useful claims about average conditional min-entropy.

Claim 7. *[DORS08] Let X, Y and Z be three distributions, where 2^b is the number of elements in the support of Y. Then,*

$$H_\infty(X|Y, Z) \geqslant H_\infty(X, Y|Z) - b$$

Claim 8 ([GKK20])**.** *Let X, Y and Z be three (arbitrary) distributions, then*

$$H_\infty(X|Y) \geqslant H_\infty(X|Y, Z)$$

3.1 Lossy Functions

Lossy functions were defined by Peikert and Waters in [PW08]. A lossy function family consists of functions of two types: lossy functions and injective ones. The lossy ones (information theoretically) lose most of the information about the input; i.e., the image is significantly smaller than the domain. It is (computationally) hard to distinguish between a random lossy function in the family and a random injective function in the family. In our setting, we will need a lossy function family where the range and the domain are of a similar size (or close to being a similar size). Intuitively, the reason is that we apply these functions to our min-entropy source, and if the functions produce output strings that are much longer than the input strings then we will lose in the min-entropy rate.

Definition 9 (Lossy functions). *A function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a (T, n, n', w)-lossy function family if the following conditions hold:*

- *There are two probabilistic polynomial time seed generation algorithms* $\mathsf{Gen}_{\mathsf{inj}}$ *and* $\mathsf{Gen}_{\mathsf{loss}}$ *s.t. for any* $\mathrm{poly}(T(\lambda))$*-size $\mathcal{A}$, it holds that*

$$\left| \Pr_{s\leftarrow\mathsf{Gen}_{\mathsf{inj}}(1^\lambda)}[\mathcal{A}(s) = 1] - \Pr_{s\leftarrow\mathsf{Gen}_{\mathsf{loss}}(1^\lambda)}[\mathcal{A}(s) = 1] \right| = \mathsf{neg}(T(\lambda)).$$

- *For every $\lambda \in \mathbb{N}$ and every $f \in \mathcal{F}_\lambda$, $f : \{0,1\}^{n(\lambda)} \to \{0,1\}^{n'(\lambda)}$.*
- *For every $\lambda \in \mathbb{N}$ and every $s \in \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)$, $f_s \in \mathcal{F}_\lambda$ is injective.*
- *For every $\lambda \in \mathbb{N}$ and every $s \in \mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$, $f_s \in \mathcal{F}_\lambda$ is lossy i.e. its image size is at most $2^{n'(\lambda)-w}$.*
- *There is a polynomial time algorithm* Eval *s.t.* $\mathsf{Eval}(s, x) = f_s(x)$ *for every $\lambda \in \mathbb{N}$, every s in the support of* $\mathsf{Gen}_{\mathsf{inj}}(1^\lambda) \cup \mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$ *and every $x \in \{0,1\}^{n(\lambda)}$.*

[PW08, BHK11] For some constant $\epsilon > 0$ and for all $c_1 \geqslant 1/\epsilon$, and for every $\Omega(\lambda) \leqslant n(\lambda) \leqslant \mathrm{poly}(\lambda)$, there exists a (T, n, n, w)-lossy function family, with $T(\lambda) = 2^{(\log\lambda)^{c_1\epsilon}}$ and $w = n - (O(\log\lambda))^c$, assuming the sub-exponential DDH assumption.

[5] This is often denoted by $\widetilde{H}_\infty(X|Y)$ in the literature.

3.2 Dispersers

Definition 10. *A function $\Gamma : [N] \times [t] \to [D]$ is a (K, K') disperser if for every $A \subseteq [N]$ with $|A| \geqslant K$ it holds that $\left|\bigcup_{a \in A, i \in [t]} \{\Gamma(a,i)\}\right| \geqslant K'$.*

We will rely on dispersers which follow from the known constructions of seeded extractors (e.g. [GUV09]).

Theorem 11 (e.g. [GUV09])**.** *There exists a constant c such that the following holds. For every N, K, K', D such that $D \leqslant \sqrt{K}$ and $K' \leqslant D/2$, there exists an efficient (K, K')- disperser*

$$\Gamma : [N] \times [t] \to [D]$$

with degree

$$t = \log^c(N)$$

3.3 Computational Extractors: Definitions

In this section, we recall definitions of extractors in the computational setting with a CRS. We define both a 2-source extractor and a non-malleable extractor in this setting.

Like [GKK20], in both defintions, we allow the min-entropy sources to depend on the CRS, but require that they are efficiently sampleable conditioned on the CRS (where the efficiency is specified by a parameter T). We also allow each source to partially leak, as long as the source has sufficient min-entropy conditioned on the CRS and the leakage.

As discussed in [GKK20], it may seem that there is no need to consider leakage explicitly. However, in general a source conditioned on fixed leakage may not be efficiently sampleable. Therefore, in the definions below we consider leakage explicitly. More specifically, for two sources X and Y we allow leakage on Y, which we will denote by L_{init}; and then allow leakage on X (that can also depend on L_{init}), which we will denote by L_{final}. Moreover, both L_{init} and L_{final} can depend on the CRS.

For technical reasons, and specifically to enable a proof of security for their two-source extractor, [GKK20] included an additional source of auxiliary information, AUX, that could be sampled jointly with Y. We do not require this auxiliary source in any of our applications or proofs. The following definitions are essentially identical to [GKK20], except we omit AUX for notational convenience.

Definition 12 (T-Admissible Leaky (n_1, n_2, k_1, k_2) Source Distribution)**.** *A T-admissible leaky (n_1, n_2, k_1, k_2) source distribution with respect to a CRS distribution $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$ consists of an ensemble of sources $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$, $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, and leakage $L = \{L_\lambda\}_{\lambda \in \mathbb{N}}$, such that $\forall \lambda \in \mathbb{N}$, the following holds:*

- *For every $\mathsf{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, $\mathsf{Supp}(X_\lambda | \mathsf{crs}) \subseteq \{0,1\}^{n_1(\lambda)}$ and $\mathsf{Supp}(Y_\lambda | \mathsf{crs}) \subseteq \{0,1\}^{n_2(\lambda)}$.*
- *The leakage L_λ consists of two parts, L_{init} and L_{final}, such that for every $\mathsf{crs} \in \mathsf{Supp}(\mathsf{CRS})$, $(Y, L_{\mathsf{init}} | \mathsf{crs})$ is sampleable in time $\mathrm{poly}(T)$, and for every $\ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}} | \mathsf{crs})$, $(X, L_{\mathsf{final}} | \mathsf{crs}, \ell_{\mathsf{init}})$ is sampleable in time $\mathrm{poly}(T)$.*
- *$H_\infty(X_\lambda | \mathsf{CRS}_\lambda, L_\lambda) \geqslant k_1$ and $H_\infty(Y_\lambda | \mathsf{CRS}_\lambda, L_\lambda) \geqslant k_2$.*
- *For every $\mathsf{crs} \in \mathsf{CRS}_\lambda$ and $\ell \in \mathsf{Supp}(L_\lambda | \mathsf{crs})$, the distributions $(X_\lambda | \mathsf{crs}, \ell)$ and $(Y_\lambda | \mathsf{crs}, \ell)$ are independent.*[6]

Definition 13 (Computational Strong 2-source Extractors)**.** *For functions $n_1 = n_1(\lambda)$, $n_2 = n_2(\lambda)$, $c = c(\lambda)$, and $m = m(\lambda)$, a function ensemble $\mathsf{2Ext} = \{\mathsf{2Ext}_\lambda\}_{\lambda \in \mathbb{N}}$, where*

$$\mathsf{2Ext}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)},$$

[6] This condition follows from the way X and Y are sampled, and like [GKK20], we add it only for the sake of being explicit.

is said to be a (n_1, n_2, k_1, k_2) strong T-computational 2-source extractor in the CRS model *if there is an ensemble* $\{\mathsf{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ *where* $\mathsf{CRS}_\lambda \in \{0,1\}^{c(\lambda)}$*, such that the following holds:*

For every T-admissible leaky (n_1, n_2, k_1, k_2) source distribution (X, Y, L) with respect to CRS*, for every polynomial p, there exists a negligible function $\nu(\cdot)$ s.t. for every λ and every $p(T(\lambda))$-size adversary $\mathcal{A}$,*

$$\left|\Pr\left[\mathcal{A}\left(\mathsf{2Ext}_\lambda(x,y,\mathsf{crs}), y, \mathsf{crs}, \ell\right) = 1\right] - \Pr\left[\mathcal{A}\left(U, y, \mathsf{crs}, \ell\right) = 1\right]\right| = \nu(T(\lambda)),$$

where the probabilities are over the randomness of sampling $(\mathsf{crs}, x, y, \ell) \leftarrow (\mathsf{CRS}_\lambda, X_\lambda, Y_\lambda, L_\lambda)$*, and over U which is uniformly distributed over $\{0,1\}^{m(\lambda)}$ independent of everything else.*

Definition 14 (Computational Strong Non-malleable Extractors)**.** *For functions $n_1 = n_1(\lambda)$, $n_2 = n_2(\lambda)$, $c = c(\lambda)$, and $m = m(\lambda)$, a function ensemble* $\mathsf{cnm\text{-}Ext} = (\mathsf{cnm\text{-}Ext}_\lambda)_{\lambda\in\mathbb{N}}$*, where*

$$\mathsf{cnm\text{-}Ext}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_2(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$

is said to be a (n_1, n_2, k_1, k_2) strong T-computational non-malleable extractor in the CRS model *if there is an ensemble* $\{\mathsf{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $\mathsf{CRS}_\lambda \in \{0,1\}^{c(\lambda)}$*, such that the following holds:*

For every T-admissible leaky (n_1, n_2, k_1, k_2) source distribution (X, Y, L) with respect to CRS*, for every polynomial p, there exists a negligible function $\nu(\cdot)$ such that for every λ and every $p(T(\lambda))$-size adversary $\mathcal{A}$,*

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}\left(\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs}), y, \mathsf{crs}, \ell\right) = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}\left(U, y, \mathsf{crs}, \ell\right) = 1\right]\right| = \nu(T(\lambda)),$$

where the oracle $\mathcal{O}^y_{x,\mathsf{crs}}$ on input $y' \neq y$ outputs $\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs})$*, and otherwise outputs $\perp$; and where the probabilities are over the randomness of sampling* $(\mathsf{crs}, x, y, \ell) \leftarrow (\mathsf{CRS}_\lambda, X_\lambda, Y_\lambda, L_\lambda)$*, and over U which is uniformly distributed over $\{0,1\}^{m(\lambda)}$ independent of everything else.*

We will occasionally need to impose a different requirement on the error distribution. In such cases we specify the error requirement explicitly. Specifically, we say that a (n_1, n_2, k_1, k_2) strong T-computational two source (or non-malleable) extractor has error $\text{neg}(\gamma(\lambda))$ if it satisfies Definition 13 (or Definition 14), where the adversary's distinguishing advantage is required to be at most negligible in $\gamma(\lambda)$.

We will also rely on the following theorem from [Raz05] (simplified to our setting). This is a statistical 2-source extractor; i.e., one that considers sources that are sampled in unbounded time, and fools adversaries with unbounded running time.

Theorem 15. *[Raz05] There exists a (n_1, n_2, k_1, k_2) strong statistical 2-source extractor with output length $O(k_2)$ according to Definition 13 where $n_2 = \omega(\log n_1)$, $k_1 \geqslant \log n_1$, and $k_2 \geqslant \alpha n_2$ for any constant $\alpha > \frac{1}{2}$, and error* $\exp^{-\Theta(\min\{k_1,k_2\})}$.

Finally, we recall the following result from [GKK20] that transforms any two-source extractor in the CRS model to a non-malleable extractor.

Theorem 16 ([GKK20])**.** *Let $T, T', n_1, n_2, k_1, k_2, k_3, w : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter where $T \geqslant 2^{k_3}$, such that the following primitives exist.*

- *A (n_1, n_2, k_1, k_2) strong T-computational 2-source extractor in the CRS model.*

- *A (T, n_1, n_1, w)-lossy function family.*
- *T'-secure collision resistant hash functions mapping $\{0,1\}^{n_2} \to \{0,1\}^{k_3}$.*

Then, there exists a (n_1, n_2, K_1, K_2) strong T'-computational non-malleable extractor satisfying definition 14 where $K_1 = k_1 + k_3(n_1 - w + 1) + 1$ and $K_2 = k_2 + k_3 + 1$.

4 Computational Strong Two-Source Extractors in the CRS Model

In this section, we describe our construction of computational two-source extractors in the CRS model. We have the following theorem.

Theorem 17. *Let $T, T', n_1, n_2, k_1, k_2, k_3, d, t, w, K_1, K_2 : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter, where $T \geqslant 2^{\max(k_3, d)}$, and such that the following primitives exist.*

- *A $(n_1, d, k_1, d - k_3 - 1)$ strong information-theoretic 2-source extractor denoted by:*
$$\mathsf{2Ext}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{d(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$
- *A (T, n_1, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$, where $w = n_1 - n_1^\gamma$ for some constant $\gamma \in (0,1)$.*
- *A T'-secure family of collision resistant hash functions $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ with $h : \{0,1\}^d \to \{0,1\}^{k_3}$.*
- *A $\left(\frac{2^{K_2/2}}{T' \log T'}, 2^{d-1}\right)$ disperser*
$$\Gamma : \{0,1\}^{n_2} \times [t] \to \{0,1\}^d$$

Then there exists a (n_1, n_2, K_1, K_2) strong T'-computational two-source extractor, satisfying Definition 14, where $K_1 = k_1 + k_3(n - w) + k_3 + 1$.

Corollary 18. *Assuming the sub-exponential hardness of DDH, there exists constants $c_0 > 1$ and c' such that for all $c > c_0$, for every $\Omega(\lambda) \leqslant n_1 \leqslant \mathrm{poly}(\lambda), \Omega(\log \lambda) \leqslant n_2 \leqslant \mathrm{poly}(\lambda)$, there exists an (n_1, n_2, K_1, K_2) λ-computational strong two-source extractor in the CRS model, with $K_1 = O(\log \lambda)^c$, $K_2 = O(\log \lambda)^c$ and output length $O(\log \lambda)^{c'}$.*

Proof. The sub-exponential hardness of DDH implies that there exists a constant $0 < \epsilon < 1$ such that DDH with security parameter λ is hard against $\mathrm{poly}(2^{\lambda^\epsilon})$-sized adversaries.

- This implies that for all $c_1 \geqslant \frac{1}{\epsilon}$, there exist lossy functions with equal domain and co-domain, where $w = n_1 - (\log \lambda)^{c_1}$, and where no $T = \mathrm{poly}(2^{\log \lambda^{c_1 \cdot \epsilon}})$-sized adversary can distinguish the lossy mode from the injective mode. This follows by setting, eg., $\log q = (\log \lambda)^{c_1}$ in the construction of lossy functions from DDH in [BHK11].
- This also implies that for all $c_2 \geqslant \frac{1}{\epsilon}$, there exist collision-resistant hash functions with range $k_3 = (\log \lambda)^{c_2}$, and where no $T' = \mathrm{poly}(2^{\log \lambda^{c_2 \cdot \epsilon}})$-sized adversary can find collisions.

Setting $c_2 = \frac{1}{\epsilon}, c_1 = \frac{1}{\epsilon^2}$, we get $T' = \lambda, k_3 = (\log \lambda)^{\frac{1}{\epsilon}}$ and $T = (2^{\log \lambda^{\frac{1}{\epsilon}}})$.

By the disperser construction in [GUV09], there exists a polynomial $t = \mathrm{poly}(\lambda)$ for which there exists a $\left(\frac{2^{K_2/2}}{T'(\log T')}, 2^{d-1}\right)$ disperser
$$\Gamma : \{0,1\}^{n_1} \times [t] \to \{0,1\}^d$$
for any d, k_2, T' that satisfy
$$K_2 \geqslant 4d + 2 \log^2 T' \tag{1}$$

Set $d = (\log \lambda)^{\frac{1}{\epsilon^2}}$. By Theorem 15, there exists a $(n_1, d, k_1, d - k_3 - 1)$ strong statistical 2-source extractor for $k_1 = (\log \lambda)^{\frac{1}{\epsilon^2}}$, with error $\exp^{-\Theta(\min(k_1, d-k_3-1))} = \mathrm{neg}(2^{k_3})$. In particular, this extractor is a $(n_1, d, k_1, d - k_3 - 1)$ strong T-computational 2-source extractor in the CRS model (where the CRS is empty), with error $\mathrm{neg}(2^{k_3})$.

Setting $d = (\log \lambda)^{\frac{1}{\epsilon^2}}$ and $T' = \lambda$ in Equation (2), we have $K_2 \geqslant 4(\log \lambda)^{\frac{1}{\epsilon^2}} + 2\log^2 \lambda$. Fixing K_2 to be $5(\log \lambda)^{\frac{1}{\epsilon^2}}$ satisfies this inequality. From Theorem 17, we have $K_1 \geqslant k_1 + k_3(n - w) + k_3 + 1 \geqslant (\log \lambda)^{\frac{1}{\epsilon^2}} + (\log \lambda)^{\frac{1}{\epsilon}} \cdot (\log \lambda)^{\frac{1}{\epsilon^2}} + (\log \lambda)^{\frac{1}{\epsilon}} + 1$. Fixing $K_1 \geqslant 2(\log \lambda)^{\frac{1}{\epsilon^3}}$ satisfies this inequality.

This completes the proof. □

Corollary 19. *Assuming the sub-exponential hardness of DDH, there exists constants $c_0 > 1$ and c' such that for all $c > c_0$, for every $\Omega(\lambda) \leqslant n_1 \leqslant \mathrm{poly}(\lambda), \Omega(\log \lambda) \leqslant n_2 \leqslant \mathrm{poly}(\lambda)$, there exists an (n_1, n_2, K_1, K_2) λ-computational non-malleable extractor in the CRS model, with $K_1 = O(\log \lambda)^c$, $K_2 = O(\log \lambda)^c$ and output length $O(\log \lambda)^{c'}$.*

Proof. This corollary can be obtained by combining Theorem 17 with 16, as follows.

- First, we apply Theorem 17 but with somewhat scaled-up parameters than in the previous corollary, to obtain an (n_1, n_2, k_1, k_2) T-computational non-malleable extractor in the CRS model, with error $\mathrm{neg}(2^{k_3})$. This extractor will be parameterized by a (small enough) constant $0 < \epsilon < 1$. It will have $T = 2^{(\log \lambda^{1/\epsilon^2})}$, and $k_3 = \log \lambda^{1/\epsilon^2}$.

  The sub-exponential hardness of DDH implies that there exists a constant $0 < \epsilon < 1$ such that DDH with security parameter λ is hard against $\mathrm{poly}(2^{\lambda^\epsilon})$-sized adversaries.

  - This implies that for all $c_1 \geqslant \frac{1}{\epsilon}$, there exist (T, n_1, n_1, w)-lossy functions with equal domain and co-domain, where $w = n_1 - (\log \lambda)^{c_1}$, and where no $\mathrm{poly}(T)$ for $T = (2^{\log \lambda^{c_1 \cdot \epsilon}})$ sized adversary can distinguish the lossy mode from the injective mode. This follows by setting, eg., $\log q = (\log \lambda)^{c_1}$ in the construction of lossy functions from DDH in [BHK11].
  - This also implies that for all $c_2 \geqslant \frac{1}{\epsilon}$, there exist collision-resistant hash functions with range $k_3 = (\log \lambda)^{c_2}$, and where no $\mathrm{poly}(T')$ for $T' = 2^{\log \lambda^{c_2 \cdot \epsilon}}$-sized adversary can find collisions.

  Setting $c_2 = \frac{1}{\epsilon^2}, c_1 = \frac{1}{\epsilon^4}$, we get $T' = 2^{\log \lambda^{\frac{1}{\epsilon}}}$, $k_3 = (\log \lambda)^{\frac{1}{\epsilon^2}}$ and $T = (2^{\log \lambda^{\frac{1}{\epsilon^3}}})$.

  By the disperser construction in [GUV09], there exists a polynomial $t = \mathrm{poly}(\lambda)$ for which there exists a $\left(\frac{2^{K_2/2}}{T'(\log T')}, 2^{d-1}\right)$ disperser
  $$\Gamma : \{0,1\}^{n_1} \times [t] \rightarrow \{0,1\}^d$$
  for any d, k_2, T' that satisfy
  $$K_2 \geqslant 4d + 2\log^2 T' \tag{2}$$

  Set $d = (\log \lambda)^{\frac{1}{\epsilon^3}}$. By Theorem 15, there exists a $(n_1, d, k_1, d - k_3 - 1)$ strong statistical 2-source extractor for $k_1 = (\log \lambda)^{\frac{1}{\epsilon^3}}$, with error $\exp^{-\Theta(\min(k_1, d-k_3-1))} = \mathrm{neg}(2^{k_3})$. In particular, this extractor is a $(n_1, d, k_1, d - k_3 - 1)$ strong T-computational 2-source extractor in the CRS model (where the CRS is empty), with error $\mathrm{neg}(2^{k_3})$.

  Setting $d = (\log \lambda)^{\frac{1}{\epsilon^3}}$ and $T' = 2^{\log \lambda^{\frac{1}{\epsilon}}}$ in Equation (2), we can set $K_2 \geqslant 4(\log \lambda)^{\frac{1}{\epsilon^3}} + 2(\log \lambda)^{\frac{2}{\epsilon}}$. Fixing $K_2 \geqslant 5(\log \lambda)^{\frac{1}{\epsilon^3}}$ satisfies the above inequality. From Theorem 17, we can set $K_1 \geqslant k_1 + k_3(n - w) + k_3 + 1$ or $K_1 \geqslant (\log \lambda)^{\frac{1}{\epsilon^3}} + (\log \lambda)^{\frac{1}{\epsilon^2}} \cdot (\log \lambda)^{\frac{1}{\epsilon^3}} + (\log \lambda)^{\frac{1}{\epsilon^2}} + 1$. Fixing $K_1 \geqslant 2(\log \lambda)^{\frac{1}{\epsilon^5}}$ satisfies this inequality.

- Re-defining some variables, we say that previous step results in a T-strong computational (n_1, n_2, k_1, k_2) non-malleable extractor in the CRS model, with $\Omega(\lambda) \leqslant n_1 \leqslant \mathrm{poly}(\lambda), \Omega(\log \lambda) \leqslant n_2 \leqslant \mathrm{poly}(\lambda)$,

$T = 2^{\log \lambda^{1/\epsilon}}$, $k_1 = 2(\log \lambda)^{\frac{1}{\epsilon^5}}, k_2 \geqslant 5(\log \lambda)^{\frac{1}{\epsilon^3}}$, and error $\mathsf{neg}(T) = \mathsf{neg}(2^{(\log \lambda)^{\frac{1}{\epsilon}}})$. Next, we apply Theorem 16 to this extractor.

As before, the subexponential hardness of DDH implies that for all $c_1' \geqslant \frac{1}{\epsilon}$, there exist (T, n_1, n_1, w)-lossy functions with equal domain and co-domain, where $w = n_1 - (\log \lambda)^{c_1'}$, and where no $\mathsf{poly}(T)$ for $T = (2^{\log \lambda^{c_1' \cdot \epsilon}})$ sized adversary can distinguish the lossy mode from the injective mode. We will set $c_1' = \frac{1}{\epsilon^2}$. We also set $k_3 = (\log \lambda)^{\frac{1}{\epsilon}}$, and by subexponential DDH, there exists a T'-secure family of collision resistant hash functions mapping $\{0,1\}^{n_2} \to \{0,1\}^{k_3}$ for $T' = \lambda$.

Then, by Theorem 16, there exists an (n_1, n_2, K_1, K_2) strong T'-computational non-malleable extractor satisfying definition 14 where $K_1 = k_1 + k_3(n_1 - w + 1) + 1 = 2(\log \lambda)^{\frac{1}{\epsilon^5}} + (\log \lambda)^{\frac{1}{\epsilon}} \cdot (\log \lambda)^{\frac{1}{\epsilon^2}} + 1$, or $K_1 \geqslant 3(\log \lambda)^{\frac{1}{\epsilon^5}}$ and $K_2 = k_2 + k_3 + 1 = 5(\log \lambda)^{\frac{1}{\epsilon^3}} + (\log \lambda)^{\frac{1}{\epsilon}} + 1$, or $K_2 \geqslant 6(\log \lambda)^{\frac{1}{\epsilon^3}}$.

This completes the proof. □

4.1 Construction

As discussed above, we will prove that the construction of two-source extractors in [GKK20] is a strong non-malleable extractor for balanced sources, and additionally only requires polylogarithmic min-entropy. We first recall the construction in [GKK20], and begin by defining the CRS distribution.

Generating the common reference string (CRS). For a given security parameter $\lambda \in \mathbb{N}$, the common reference string is generated as follows.

1. Sample $h \leftarrow \mathcal{H}_\lambda$.
2. Sample $b = (b_1, \ldots, b_{k_3}) \leftarrow \{0,1\}^{k_3}$.
3. Sample independently k_3 pairs of random injective functions from $\mathcal{F}_\lambda$,
$$f_{1,b_1}, f_{2,b_2}, \ldots, f_{k_3,b_{k_3}} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda).$$
4. Sample independently k_3 pairs of random lossy functions from $\mathcal{F}_\lambda$,
$$f_{1,1-b_1}, f_{2,1-b_2}, \ldots, f_{k_3,b_{1-k_3}} \leftarrow \mathsf{Gen}_{\mathsf{loss}}(1^\lambda).$$

Output
$$\mathsf{crs} = \left(h, \begin{array}{c} f_{1,0}, f_{2,0}, \ldots, f_{k_3,0} \\ f_{1,1}, f_{2,1}, \ldots, f_{k_3,1} \end{array} \right)$$

The (Computational) Two-Source Extractor: Construction.

The computational two-source extractor $\mathsf{c2Ext} = \{\mathsf{c2Ext}_\lambda\}_{\lambda \in \mathbb{N}}$ is defined as follows. For any $\lambda \in \mathbb{N}$, denote by $c = c(\lambda) = |\mathsf{crs}|$, then
$$\mathsf{c2Ext}_\lambda : \{0,1\}^c \times \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m,$$
where $\forall (\mathsf{crs}, x_1, x_2) \in \{0,1\}^c \times \{0,1\}^{n_1} \times \{0,1\}^{n_2}$,
$$\mathsf{c2Ext}_\lambda(\mathsf{crs}, x_1, x_2) = \bigoplus_{y : \exists i \text{ s.t. } \Gamma(x_2, i) = y} \mathsf{cnm\text{-}Ext}_\lambda(\mathsf{crs}, x_1, y)$$
where $\Gamma : \{0,1\}^{n_2} \times [t] \to \{0,1\}^d$ is a $(\frac{2^{k_2}}{T' \log T'}, 2^{d-1})$ disperser, and $\forall (\mathsf{crs}, x_1, y) \in \{0,1\}^c \times \{0,1\}^{n_1} \times \{0,1\}^d$, and crs parsed as $\left(h, \begin{array}{c} f_{1,0}, f_{2,0}, \ldots, f_{k_3,0} \\ f_{1,1}, f_{2,1}, \ldots, f_{k_3,1} \end{array} \right)$,
$$\mathsf{cnm\text{-}Ext}_\lambda(\mathsf{crs}, x_1, y) = \mathsf{2Ext}_\lambda\Big(f_{1,h(y)_1} \circ f_{2,h(y)_2} \circ \ldots \circ f_{k_3,h(y)_{k_3}}(x_1), y \Big)$$

4.2 Proof of Security of Computational Extractor

This section contains the proof of Theorem 17.

First, we prove the following claim.

Claim 20. *Define* $\mathsf{c2Ext}'_\lambda$ *such that on input* (crs, x_1, x_2) *for* crs *parsed as*

$$\left(h, \begin{array}{c} f_{1,0}, f_{2,0}, \ldots, f_{k_3,0} \\ f_{1,1}, f_{2,1}, \ldots, f_{k_3,1} \end{array} \right),$$

$\mathsf{c2Ext}'_\lambda(\mathsf{crs}, x_1, x_2)$ *first checks if* $\exists(i_1, i_2)$ *such that* $\Gamma(x_2, i_1) \neq \Gamma(x_2, i_2)$ *but* $h(\Gamma(x_2, i_1)) = h(\Gamma(x_2, i_2))$*. If so,* $\mathsf{c2Ext}'_\lambda(\mathsf{crs}, x_1, x_2)$ *outputs a uniformly random value in* $\{0,1\}^{m(\lambda)}$*. Otherwise* $\mathsf{c2Ext}'_\lambda(\mathsf{crs}, x_1, x_2) = \mathsf{c2Ext}_\lambda(\mathsf{crs}, x_1, x_2)$*.*

For every $\mathrm{poly}(T')$*-sampleable* (n_1, n_2, k_1, k_2) *source distribution* (X, Y) *with respect to* CRS*, for every polynomial* p*, there exists a negligible function* $\nu(\cdot)$ *such that for every* λ *and every unbounded adversary* $\mathcal{A}$*,*

$$\left| \Pr\left[\mathcal{A}\left(\mathsf{c2Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell\right) = 1\right] - \Pr\left[\mathcal{A}\left(\mathsf{c2Ext}'(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell\right) = 1\right] \right| = \nu(T'(\lambda))$$

where the probabilities are over the randomness of sampling $(\mathsf{crs}, x, y) \leftarrow (\mathrm{CRS}_\lambda, X_\lambda, Y_\lambda)$*.*

Proof. Suppose the claim is not true. Then we break the collision resistance property of $\mathcal{H}$ by constructing an algorithm $\mathcal{A}'$, running in time $\mathrm{poly}(T')$ that on input $h \leftarrow \mathcal{H}$, finds collisions as follows.

1. Sample $\mathsf{crs}' \leftarrow \mathrm{CRS}'$, set $\mathsf{crs} = (h, \mathsf{crs}')$
   and sample $r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}$. Set $x_2 = X_2(\mathsf{crs}; r)$.
2. Check if there exist $(i, j) \in [t]$ such that $\Gamma(x_2, i) \neq \Gamma(x_2, j)$ and $h(\Gamma(x_2, i)) = h(\Gamma(x_2, j))$.
3. If such a pair (i, j) is found then output $(\Gamma(x_2, i), \Gamma(x_2, j))$ as a collision, and otherwise output $\perp$.

Because x_2 and crs' are sampleable in time $\mathrm{poly}(T')$, the disperser is computable in time $\mathrm{poly}(\lambda)$ and the degree of the disperser is $\mathrm{poly}(\lambda)$, $\mathcal{A}'$ runs in time $\mathrm{poly}(T')$. The two distributions differ only if there is a collision. Moreover, by our assumption it finds a collision with probability at least $\frac{1}{p(T')}$ (for some polynomial $p(\cdot)$ and infinitely many $\lambda \in \mathbb{N}$). This contradicts collision resistance of the hash function, as desired. □

Fix the distribution CRS defined in the construction, and any (n_1, n_2, k_1, k_2) source distribution $(\mathcal{X}_1, \mathcal{X}_2, L)$ for $L = (L_{\mathsf{init}}, L_{\mathsf{final}})$ for which there exists a PPT adversary that contradicts Definition 13.

Next, for any unbounded adversary $\mathcal{A}$, and bit $c \in \{0,1\}$, define experiment $\mathsf{Exp}_{\mathcal{A},\mathsf{crs},y,\ell_{\mathsf{init}},c}$ as follows.

- Sample $(x, \ell_{\mathsf{final}}) \leftarrow (\mathcal{X}_1, L_{\mathsf{final}} | \mathsf{crs}, \ell_{\mathsf{init}})$.
- Obtain $g_1, \ldots, g_t \leftarrow \mathcal{A}(\mathsf{crs}, y)$.
- If $\exists i \in [t]$ such that $h(y) = h(g_i(y))$, abort.
- If $c = 0$, output
  $\mathcal{A}\Big(\mathsf{cnm\text{-}Ext}_\lambda(x, y), \mathsf{cnm\text{-}Ext}_\lambda(x, g_1(y)), \mathsf{cnm\text{-}Ext}_\lambda(x, g_2(y)), \ldots, \mathsf{cnm\text{-}Ext}_\lambda(x, g_t(y)), \mathsf{crs}, \ell\Big)$.
- If $c = 1$, output
  $\mathcal{A}\Big(U_m, \mathsf{cnm\text{-}Ext}_\lambda(x, g_1(y)), \mathsf{cnm\text{-}Ext}_\lambda(x, g_2(y)), \ldots, \mathsf{cnm\text{-}Ext}_\lambda(x, g_t(y)), \mathsf{crs}, \ell\Big)$, where U_m denotes a uniformly random value in $\{0,1\}^m$.

For any $b \in \{0,1\}^{k_3}$, let $\mathrm{CRS}(b)$ denote the set of all common reference strings such that $f_{1,b_1}, \dots, f_{k_3,b_{k_3}}$ in crs are injective, and the others are lossy.

For function $\epsilon = \epsilon(\lambda)$, define set $\mathsf{BAD\text{-}seed}_\epsilon$ as:

$$\mathsf{BAD\text{-}seed}_\epsilon = \Big\{ y : \exists \mathsf{crs} \in \mathrm{CRS}(h(y)), \mathcal{A}, \ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}} | \mathsf{crs}) \text{ s.t. } \big| \Pr[\mathsf{Exp}_{\mathcal{A},\mathsf{crs},y,\ell_{\mathsf{init}},0}] = 1] - \Pr[\mathsf{Exp}_{\mathcal{A},\mathsf{crs},y,\ell_{\mathsf{init}},1}] = 1] \big| > \epsilon \Big\}$$

Next we have the following information-theoretic argument.

Claim 21. *Let $\nu(\lambda)$ denote the error in Raz's extractor. Then*

$$\Pr_{y \leftarrow \{0,1\}^d} [y \in \mathsf{BAD\text{-}seed}_{\sqrt{\nu(\lambda)}}] \leqslant \sqrt{\nu(\lambda)} \tag{3}$$

Proof. Suppose the claim is false. This implies that there exists an (unbounded) adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\begin{aligned} &\Bigg| \Pr_{\substack{(\mathsf{crs},x,y,\ell) \leftarrow (\mathrm{CRS},\mathcal{X},\mathcal{Y},L) \\ g_1,\dots,g_t \leftarrow \mathcal{A}(\mathsf{crs},y) \\ h(y)=b \text{ where } \mathsf{crs} \in \mathrm{CRS}(b)}} \Big[\mathcal{A}\Big(\mathsf{cnm\text{-}Ext}_\lambda(x,y), \mathsf{cnm\text{-}Ext}_\lambda(x, g_1(y)), \dots, \mathsf{cnm\text{-}Ext}_\lambda(x, g_t(y)), \mathsf{crs}, \ell \Big) = 1 \Big] \\ &- \Pr_{\substack{(\mathsf{crs},x,y,\ell) \leftarrow (\mathrm{CRS},\mathcal{X},\mathcal{Y},L) \\ g_1,\dots,g_t \leftarrow \mathcal{A}(\mathsf{crs},y) \\ h(y)=b \text{ where } \mathsf{crs} \in \mathrm{CRS}(b)}} \Big[\mathcal{A}\Big(U_m, \mathsf{cnm\text{-}Ext}_\lambda(x, g_1(y)), \dots, \mathsf{cnm\text{-}Ext}_\lambda(x, g_t(y)), \mathsf{crs}, \ell \Big) = 1 \Big] \Bigg| \\ &> \nu(\lambda) \end{aligned} \tag{4}$$

We will now define an (n_1, n_2, k_1, k_2) source distribution (X', Y') for the underlying statistical two-source extractor where $k_1 = K_1 - k_3 \cdot (n_1 - w + 1) - 1$ and $k_2 = K_2 - k_3 - 1$, such that $\mathcal{A}$ breaks the (n_1, n_2, k_1, k_2) statistical two-source extractor for (X', Y').

Define (X', Y') as follows.

1. We first define Y':
   (a) Sample $b \leftarrow \{0,1\}^{k_3}$.
   (b) Sample $f_h = \left(h, \begin{array}{c} f_{1,0}, f_{2,0}, \dots, f_{k_3,0} \\ f_{1,1}, f_{2,1}, \dots, f_{k_3,1} \end{array} \right)$ s.t. $\{f_{i,b_i}\}_{i \in [k_3]}$ are injective and the rest are lossy.
   (c) Sample $(y, \ell_{\mathsf{init}}) \leftarrow (Y, L_{\mathsf{init}} | \mathsf{crs})$.
   (d) Set $y' = (y, d, \ell_{\mathsf{init}}, f_h, b)$, where $d = 0$ if $h(y) \neq b$ and 1 otherwise.
2. We next define X':
   (a) Sample $(x, \ell_{\mathsf{final}}) \leftarrow (X, L_{\mathsf{final}} | \mathsf{crs}, \ell_{\mathsf{init}})$. Set $x' = f_{1,b_1} \circ f_{2,b_2} \circ \dots \circ f_{k_3,b_{k_3}}(x), (\ell_{\mathsf{final}}, z_{x,b})$, where $z_{x,b} = \{z_1, \dots, z_{k_3}\}$ and for every $i \in [\ell]$, $z_i := f_{i,1-b_i}(f_{i+1,b_{i+1}}(\dots f_{k_3,b_{k_3}}(x)))$..

It remains to show that (X', Y', L') is a T-admissible leaky (n_1, n_2, k_1, k_2) source distribution with respect to $\mathrm{CRS}_{\mathsf{2Ext}}$, where $k_1 = K_1 - k_3 \cdot (n_1 - w + 1) - 1$ and $k_2 = K_2 - k_3 - 1$.

Note that

$$\begin{aligned} H_\infty(Y' | \mathsf{crs}_{\mathsf{2Ext}}, d, \ell_{\mathsf{init}}, f_h, b) &\geqslant H_\infty(Y | \mathsf{crs}_{\mathsf{2Ext}}, \ell_{\mathsf{init}}, f_h) - k_3 - 1 \quad \text{(by Claim 7)} \\ &= H_\infty(Y | \mathsf{crs}_{\mathsf{2Ext}}, \ell_{\mathsf{init}}) - k_3 - 1 \\ &\geqslant K_2 - k_3 - 1 \quad \text{(by assumption).} \end{aligned}$$

Similarly,

$$
\begin{aligned}
&H_\infty(X' | \mathsf{crs}_{\mathsf{2Ext}}, \ell_{\mathsf{final}}, z_{x,b}, d, \ell_{\mathsf{init}}, f_h, b) \\
&= H_\infty(X | \mathsf{crs}_{\mathsf{2Ext}}, \ell_{\mathsf{final}}, z_{x,b}, d, \ell_{\mathsf{init}}, f_h, b) \quad \text{(since } f_{i,b_i}\text{'s are injective)} \\
&\geqslant H_\infty(X' | \mathsf{crs}_{\mathsf{2Ext}}, \ell_{\mathsf{final}}, \ell_{\mathsf{init}}, f_h) - k_3 \cdot (n_1 - \omega + 1) - 1 \\
&\qquad\qquad\qquad\qquad \text{(by Claim 7 and since } f_{i,1-b_i}\text{'s are lossy)} \\
&= H_\infty(X' | \mathsf{crs}, \ell) - k_3 \cdot (n_1 - \omega + 1) - 1 \\
&\geqslant K_1 - k_3 \cdot (n_1 - \omega + 1) - 1 \qquad \text{(by assumption).}
\end{aligned}
$$

Next, note that $\ell' = (\ell'_{\mathsf{init}}, \ell'_{\mathsf{final}})$, where $\ell'_{\mathsf{init}} = (\ell_{\mathsf{init}}, f_h, b, d)$ and $\ell'_{\mathsf{final}} = (\ell_{\mathsf{final}}, z)$. Let $\mathsf{crs} = (\mathsf{crs}_{\mathsf{2Ext}}, f_h)$, and let $\ell = (\ell_{\mathsf{init}}, \ell_{\mathsf{final}})$. In this case $X' = f_{1,b_1} \circ \ldots f_{k_3,b_{k_3}}(X)$ where X is sampled conditioned on (crs, ℓ), and $Y' = Y$ where Y is sampled conditioned on $(\mathsf{crs}, \ell_{\mathsf{init}})$ and on $h(Y) = b$.

The fact that (X, Y, L) is T-admissible w.r.t. CRS, implies that X and Y are independent conditioned on (crs, ℓ). Moreover, since $h(Y), d$ are a function of Y and the crs, we have that X and Y are independent conditioned on (crs, ℓ, d) and on $h(Y) = b$. This implies that $f_{1,b_1} \circ \ldots f_{k_3,b_{k_3}}(X)$ and Y are indepedent conditioned on (crs, ℓ, d) and on $h(Y) = b$, and moreover z is just a function of crs, x. This in turn implies that indeed X' and Y' are independent conditioned on $(\mathsf{crs}_{\mathsf{2Ext}}, \ell')$, as desired.

We next argue that Equation (4), together with the definition of $(X', Y', L' | \mathsf{crs}_{\mathsf{2Ext}})$, implies that there exists a T-size adversary $\mathcal{A}'$, that simulates the adversary $\mathcal{A}$, as well as its oracle, such that for infinitely many $\lambda \in \mathbb{N}$,

$$
\begin{aligned}
&\Pr[\mathcal{A}'(\mathsf{2Ext}(X', Y', \mathsf{crs}_{\mathsf{2Ext}}), y', \mathsf{crs}_{\mathsf{2Ext}}, \ell') = 1] - \Pr[\mathcal{A}'(U, y', \mathsf{crs}_{\mathsf{2Ext}}, \ell') = 1] \\
&> \nu(\lambda)
\end{aligned} \tag{5}
$$

The algorithm $\mathcal{A}'$ on input $(\alpha, y', \mathsf{crs}_{\mathsf{2Ext}}, \ell')$ does the following:

1. Parse $\ell' = (\ell'_{\mathsf{init}}, \ell'_{\mathsf{final}})$ and further parse $\ell'_{\mathsf{init}} = (d, \ell_{\mathsf{init}}, f_h, h(y))$, $\ell'_{\mathsf{final}} = (\ell_{\mathsf{final}}, z_{x,h(y)})$.
2. If $d = 0$ then output $\perp$.
3. Else, set $\ell = (\ell_{\mathsf{init}}, \ell_{\mathsf{final}})$, and set $\mathsf{crs} = (\mathsf{crs}_{\mathsf{2Ext}}, f_h)$.
4. Output $\mathcal{A}^{\tilde{\mathcal{O}}}(\alpha, y', \mathsf{crs}, \ell)$, where the oracle $\tilde{\mathcal{O}}$ is simulated using $(h(y), z_{x,h(y)}, \mathsf{crs})$.

Equation (4) implies that indeed Equation (5) holds, as desired. This contradicts the fact that 2Ext is a strong T-computational 2-source extractor for (X', Y', L'). This completes the proof that Equation (3) holds. □

Next, we fix $\epsilon = \epsilon(\lambda) = \sqrt{\nu(\lambda)}$, where $\nu(\lambda)$ denotes the error in Raz's extractor. For any adversary $\mathcal{A}$ we consider a set of games, $\mathsf{Game}_{\mathcal{A},0,\alpha}$, $\mathsf{Game}_{\mathcal{A},1,\alpha}$, $\mathsf{Game}_{\mathcal{A},2,\alpha}$ for $\alpha \in \{0, 1\}$. Before formally describing these games, we informally discuss them and our high-level approach for the rest of the proof.

- $\mathsf{Game}_{\mathcal{A},0,\alpha}$ sends to the adversary either the output of the extractor $\mathsf{cnm\text{-}Ext}'$ on randomly sampled $(x_1, x_2, \ell | \mathsf{crs})$ or a uniformly random value, depending on the choice of α.
- $\mathsf{Game}_{\mathcal{A},1,\alpha}$ is identical to Game 0 except that it samples x_2 restricted to the existence of an i such that $\Gamma(x_2, i)$ is not in $\mathsf{BAD\text{-}seed}_\epsilon$.
- $\mathsf{Game}_{\mathcal{A},2,\alpha}$ is identical to Game 1 except that it additionally conditions on $h(y) = b$, for b s.t. the functions $\{f_{i,b_i}\}_{i \in [k_3]}$ in the CRS are injective, and the rest are lossy.

Once we formally define these games, we discuss the intuition for some claims that we will establish about these games:

- For every unbounded $\mathcal{A}$, $\mathsf{Game}_{\mathcal{A},2,0}$ and $\mathsf{Game}_{\mathcal{A},2,1}$ are ϵ-statistically indistinguishable. We will show that this will follow by definition of the set $\mathsf{BAD\text{-}seed}_\epsilon$.

- If there exists a poly(T)-size adversary $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathcal{A},1,0}$ and $\mathsf{Game}_{\mathcal{A},1,1}$ with advantage better than neg(T), then $\mathcal{A}$ also distinguishes $\mathsf{Game}_{\mathcal{A},2,0}$ from $\mathsf{Game}_{\mathcal{A},2,1}$. This is a computational argument that relies on the fact that lossy trapdoor functions "hide" the string b.

  This, combined with the previous bullet establishes that $\mathsf{Game}_{\mathcal{A},1,0}$ and $\mathsf{Game}_{\mathcal{A},1,1}$ are computationally indistinguishable w.r.t. poly(T)-sized adversaries $\mathcal{A}$, with advantage $\epsilon + \text{neg}(T)$.

- For every unbounded $\mathcal{A}$ and every $\alpha \in \{0, 1\}$, $\mathsf{Game}_{\mathcal{A},0,\alpha}$ and $\mathsf{Game}_{\mathcal{A},1,\alpha}$ are statistically indistinguishable. Intuitively, this is because the set $\mathsf{BAD\text{-}seed}_\epsilon$ is small: thus by the property of the disperser, the probability of sampling x_2 for which no index i exists s.t. $\Gamma(x_2, i)$ is not in $\mathsf{BAD\text{-}seed}_\epsilon$, is negligible.

  This, combined with the previous bullet establishes that $\mathsf{Game}_{\mathcal{A},0,0}$ and $\mathsf{Game}_{\mathcal{A},0,1}$ are computationally indistinguishable w.r.t. poly(T)-sized adversaries, with advantage $\epsilon + \text{neg}(T) + \text{poly}\left(\frac{1}{T' \log T'}\right)$.

We now proceed to formally define the games, then formalize and prove the above claims.
$\mathsf{Game}_{\mathcal{A},0,\alpha}$:

1. Sample $\mathsf{crs} \leftarrow \text{CRS}$.
2. Sample $(x_2, \ell_{\mathsf{init}} \leftarrow X_2, L_{\mathsf{init}}|\mathsf{crs})$.
   - If $\alpha = 0$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(\mathsf{c2Ext}'(\mathsf{crs}, x_1, x_2), x_2, \mathsf{crs}, \ell)$.
   - If $\alpha = 1$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(U_m, x_2, \mathsf{crs}, \ell)$.

$\mathsf{Game}_{\mathcal{A},1,\alpha}$:

1. Sample $\mathsf{crs} \leftarrow \text{CRS}$ and let b denote the underlying value such that the functions $\{f_{i,b_i}\}_{i\in[k_3]}$ in the CRS are injective and the rest are lossy.
2. Sample $(x_2, \ell_{\mathsf{init}} \leftarrow X_2, L_{\mathsf{init}}|\mathsf{crs})$.
3. If $\exists i \in [t]$ such that $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_\epsilon$, do:
   - If $\alpha = 0$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(\mathsf{c2Ext}'(\mathsf{crs}, x_1, x_2), x_2, \mathsf{crs}, \ell)$.
   - If $\alpha = 1$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(U_m, x_2, \mathsf{crs}, \ell)$.

   Otherwise, go back to Step 1.

$\mathsf{Game}_{\mathcal{A},2,\alpha}$:

1. Sample $\mathsf{crs} \leftarrow \text{CRS}$ and let b denote the underlying value such that the functions $\{f_{i,b_i}\}_{i\in[k_3]}$ in the CRS are injective and the rest are lossy.
2. Sample $(x_2, \ell_{\mathsf{init}} \leftarrow X_2, L_{\mathsf{init}}|\mathsf{crs})$.
3. Sample $i \leftarrow [t]$ such that $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_\epsilon$. If such i does not exist, go back to Step 1.
4. If $h(\Gamma(x_2, i)) = b$, do:
   - If $\alpha = 0$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(\mathsf{c2Ext}'(\mathsf{crs}, x_1, x_2), x_2, \mathsf{crs}, \ell)$.
   - If $\alpha = 1$, sample $(x_1, \ell_{\mathsf{final}} \leftarrow X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(U_m, x_2, \mathsf{crs}, \ell)$.

   Otherwise go back to step 1.

Next, we prove that the distributions $\mathsf{Game}_{\mathcal{A},0,0}$ and $\mathsf{Game}_{\mathcal{A},0,1}$ are statistically close, for all $\mathcal{A}$.

Claim 22. *For all unbounded $\mathcal{A}$,*

$$\left| \Pr[\mathsf{Game}_{\mathcal{A},2,0} = 1] - \Pr[\mathsf{Game}_{\mathcal{A},2,1} = 1] \right| \leqslant \epsilon$$

Proof. We will show that this claim follows by the definition of $\mathsf{BAD\text{-}seed}_\epsilon$. Towards a contradiction, assume that the claim is not true.

This implies that there exists an unbounded $\mathcal{A}$ such that

$$\Big| \Pr[\mathsf{Game}_{\mathcal{A},2,0} = 1] - \Pr[\mathsf{Game}_{\mathcal{A},2,1} = 1] \Big| > \epsilon$$

By an averaging argument, this means that there exists $b, \mathsf{crs} \in \mathsf{CRS}(b), (x_2, \ell_{\mathsf{init}}) \in \mathsf{Supp}(\mathcal{X}_2, L_{\mathsf{init}}|\mathsf{crs}), i$ and $\mathcal{A}'$ such that for $y = \Gamma(x_2, i)$, we have that $y \notin \mathsf{BAD\text{-}seed}_\epsilon, h(y) = b, \forall i \in [t], h(y) \neq h(g_i(y))$ and:

$$\Big| \Pr[\mathcal{A}'(\mathsf{c2Ext}'(\mathcal{X}_1, x_2, \mathsf{crs}), x_2, \mathsf{crs}) = 1] - \Pr[\mathcal{A}'(U_m, x_2, \mathsf{crs}) = 1] \Big| > \epsilon \tag{6}$$

Fix such x_2, y and $\mathcal{A}'$, and note that $y \notin \mathsf{BAD\text{-}seed}_\epsilon$. By definition of $\mathsf{BAD\text{-}seed}_\epsilon$, for all unbounded $\mathcal{B}$, all $y \notin \mathsf{BAD\text{-}seed}_\epsilon$, $\mathsf{crs} \in \mathsf{CRS}(h(y)), \ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$,

$$\big| \Pr[\mathsf{Exp}_{\mathcal{B},\mathsf{crs},y,\ell_{\mathsf{init}},0} = 1] - \Pr[\mathsf{Exp}_{\mathcal{B},\mathsf{crs},y,\ell_{\mathsf{init}},1} = 1] \big| \leqslant \epsilon \tag{7}$$

We construct an adversary $\mathcal{B}$ that contradicts Equation (7) as follows. Recall that we fixed x_2, y. For every $i \in [t]$, $\mathcal{B}$ sets $g_i = h(\Gamma(x_2, i))$ except whenever $\Gamma(x_2, j) = y$ for some j, it sets $g_j = \perp$. Upon obtaining challenge $(\alpha, c_1, \ldots, c_t)$, it outputs $\mathcal{A}'\big((\bigoplus_{i\in[t]} c_i \oplus \alpha), x_2, \mathsf{crs}\big)$. This contradicts Equation (7) as desired, and proves the claim. □

Next, we will show that any distinguisher that successfully distinguishes $(\mathsf{Game}_{\mathcal{A},1,0}, \mathsf{Game}_{\mathcal{A},1,1})$, also successfully distinguishes $(\mathsf{Game}_{\mathcal{A},2,0}, \mathsf{Game}_{\mathcal{A},2,1})$.

Claim 23. *Suppose there exists a* $\mathrm{poly}(T)$*-size adversary* $\mathcal{A}$*, polynomial* $p(\cdot)$*, and* $\mathrm{poly}(T)$*-sampleable* (n_1, n_2, k_1, k_2) *source distribution* (X_1, X_2) *with respect to* CRS *such that:*

$$\Big| \Pr\left[\mathsf{Game}_{\mathcal{A},1,0} = 1\right] - \Pr\left[\mathsf{Game}_{\mathcal{A},1,1} = 1\right] \Big| \geqslant \frac{1}{p(2^{k_3})} \tag{8}$$

then

$$\Big| \Pr\left[\mathsf{Game}_{\mathcal{A},2,0} = 1\right] - \Pr[\mathsf{Game}_{\mathcal{A},2,1} = 1] \Big| \geqslant \frac{1}{8p(2^{k_3})} \tag{9}$$

Proof. Suppose the claim is not true. Then there exists a $\mathrm{poly}(T)$-size adversary $\mathcal{A}$, polynomial $p(\cdot)$, and $\mathrm{poly}(T)$-sampleable (n_1, n_2, k_1, k_2) source distribution $(X, Y, L|\mathsf{CRS})$ such that:

$$\Pr\left[\mathsf{Game}_{\mathcal{A},1,0} = 1\right] - \Pr\left[\mathsf{Game}_{\mathcal{A},1,1} = 1\right] \geqslant \frac{1}{p(2^{k_3})} \text{ and} \tag{10}$$

$$\Pr\left[\mathsf{Game}_{\mathcal{A},2,0} = 1\right] - \Pr\left[\mathsf{Game}_{\mathcal{A},2,1} = 1\right] < \frac{1}{8p(2^{k_3})} \tag{11}$$

The other case can be handled symmetrically, so this assumption is w.l.o.g. Define an additional game as follows.

$\mathsf{Game}_{\mathcal{A},3,\alpha}$:

- Sample $\mathsf{crs} \leftarrow \mathsf{CRS}$ and let b denote the underlying value such that the functions $\{f_{i,b_i}\}_{i\in[k_3]}$ in the CRS are injective and the rest are lossy.
- Sample $(x_2, \ell_{\mathsf{init}}) \leftarrow X_2, L_{\mathsf{init}}|\mathsf{crs}$.
- Sample $i \leftarrow [t]$ such that $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon(2^{k_3})}$. If such i does not exist, go back to Step 1.

- If $h(\Gamma(x_2, i)) \neq b$, do:
  - If $\alpha = 0$, sample $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}} | \mathsf{crs}, \ell_{\mathsf{init}})$ and output $\mathcal{A}(\mathsf{c2Ext}'(\mathsf{crs}, x_1, x_2), x_2, \mathsf{crs}, \ell)$.
  - If $\alpha = 1$, output $\mathcal{A}(U_m, x_2, \mathsf{crs}, \ell)$.

  Otherwise go back to step 1.

Then note that

$$\begin{aligned}
&\Pr[\mathsf{Game}_{\mathcal{A},3,0} = 1] - \Pr[\mathsf{Game}_{\mathcal{A},3,1} = 1] \\
&= \Pr[\mathsf{Game}_{\mathcal{A},2,0} = 1] - \Pr[\mathsf{Game}_{\mathcal{A},2,1} = 1] \\
&- \Pr[\mathsf{Game}_{\mathcal{A},1,0} = 1] + \Pr[\mathsf{Game}_{\mathcal{A},1,1} = 1] \\
&\geqslant \frac{1}{p(2^{k_3})} - \frac{1}{8p(2^{k_3})} \geqslant \frac{7}{8p(2^{k_3})}
\end{aligned} \tag{12}$$

where the second-from-last inequality holds for infinitely many $\lambda \in \mathbb{N}$ by Equations (10) and (11).

We will now construct an adversary $\mathcal{A}'$ that contradicts key indistinguishability of the lossy function family, as follows. $\mathcal{A}'$ obtains

$$\begin{aligned}
&f_{1,0}, f_{2,0}, \ldots, f_{k_3,0} \\
&f_{1,1}, f_{2,1}, \ldots, f_{k_3,1}
\end{aligned}$$

externally, where functions corresponding to (hidden value) $b = (b_1, \ldots, b_{k_3})$ are injective. Next, it does the following:

- Set $N = 2^{k_3} \cdot p^2(2^{k_3})$.
- Sample $z \leftarrow \{0,1\}^{k_3}$.
- Initialize $\alpha = 0, \beta = 0, i = 0$ and do:
  1. If $i = N + 1$, output α, β and stop.
  2. Else sample $h \leftarrow \mathcal{H}_\lambda$ and set

     $$\mathsf{crs} = \left(h, \begin{array}{c} f_{1,0}, f_{2,0}, \ldots, f_{k_3,0} \\ f_{1,1}, f_{2,1}, \ldots, f_{k_3,1}. \end{array}\right)$$

  3. Sample $(x_2, \ell_{\mathsf{init}}) \leftarrow (X_{2,\lambda}, L_{\mathsf{init}} | \mathsf{crs})$.

     Define function F that on input $(x_2, \ell_{\mathsf{init}}, \mathsf{crs})$ outputs a uniformly random $i \leftarrow [t]$ s.t. $\Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_{\epsilon(2^{k_3})}$. It outputs $\perp$ if no such i exists.

     Note that function F can be implemented by a 2^d-sized circuit.
  4. If $h(y) = z$, continue. Else go back to Step 1.
  5. Sample $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}} | \mathsf{crs}, \ell_{\mathsf{init}})$.
  6. Set $i = i + 1, \alpha = \alpha + \mathcal{A}\left(\mathsf{c2Ext}'(x_1, x_2, \mathsf{crs}), x_2, \mathsf{crs}, \ell\right), \beta = \beta + \mathcal{A}\left(U, x_2, \mathsf{crs}, \ell\right)$. Go to Step 1.
- Set $\delta = |\alpha - \beta|$.
- If $\delta < \frac{1}{2p(2^{k_3})}$ then output $b' = z$, otherwise output $b' \leftarrow \{0,1\}^{k_3}$.

Now when i is sampled as the output of function F, by equation (11),

$$\mathbb{E}[\delta | z = b] < \frac{1}{8p(2^{k_3})}$$

and by equation (12),

$$\mathbb{E}[\delta | z \neq b] \geqslant \frac{7}{8p(2^{k_3})}$$

Define $\mathbb{H}$ as the event that $\left(\delta < \frac{1}{2p(2^{k_3})}\right)$. By the Chernoff bound[7],

$$\Pr[\mathbb{H}|z = b] = 1 - \Pr\left[\left(\delta \geqslant \frac{1}{2p(2^{k_3})}\right)\middle| z = b\right] \geqslant 1 - \left(e^{-\frac{2^{k_3}\cdot p^2(2^{k_3})}{32p^2(2^{k_3})}}\right) > 1 - \mathsf{neg}(2^{k_3}) \tag{13}$$

Note that (by definition) $\neg\mathbb{H}$ is the event that $\left(\delta \geqslant \frac{1}{2p(2^{k_3})}\right)$. Therefore, by the Chernoff bound[8],

$$\Pr[\neg\mathbb{H}|z \neq b] = 1 - \Pr\left[\left(\delta < \frac{1}{2p(2^{k_3})}\right)\middle| z \neq b\right] \geqslant 1 - e^{-\frac{0.75^2}{p^2(2^{k_3})}\cdot\frac{2^{k_3}\cdot p^2(2^{k_3})}{2}} = 1 - \mathsf{neg}(2^{k_3}) \tag{14}$$

Next, note that the T-security of the lossy function family, together with the assumption that $T \geqslant 2^{k'}$ where $k' = \max\{k_3, d\}$, implies that for every poly(T)-size adversary $\mathcal{B}$ (recall $b \in \{0,1\}^{k_3}$ is used to determine which functions are lossy or injective in the crs),

$$2^{-k'} + \mathsf{neg}(T) \geqslant \Pr[\mathcal{B}(\mathsf{crs}) = b] \geqslant 2^{-k'} - \mathsf{neg}(T). \tag{15}$$

This, together with the fact that $(X, Y, L|\mathsf{crs})$ can be sampled in time poly(T), implies that

$$2^{-k'} + \mathsf{neg}(T) \geqslant \Pr\left[h(y) = b\right] \geqslant 2^{-k'} - \mathsf{neg}(T), \tag{16}$$

where the probability is over $\mathsf{crs} \leftarrow$ CRS, and over $(x, y, \ell) \leftarrow (X, Y, L|\mathsf{crs})$.
Furthermore, by construction,

$$\Pr[b' = b|\mathbb{H} \wedge z = b] = 1 \tag{17}$$

and

$$\Pr[b' = b|\neg\mathbb{H} \wedge z \neq b] = 2^{-k_3} \tag{18}$$

[7] We are using the following version of the Chernoff bound: Let $X_1, \ldots X_N$ be independent random variables taking values in $[-1, 1]$. Let X denote their mean, and $\mu = \mathbb{E}[X]$ denote the expected value of their mean. Then for every $\alpha > 0$,

$$\Pr[X \geqslant \mu + \epsilon] \leqslant e^{-\frac{\epsilon^2 N}{2}}$$

We derive Equation (13) by setting $\epsilon = \frac{1}{4p(2^{k_3})}$, $N = 2^{k_3} \cdot p^2(2^{k_3})$.

[8] Here we are using the following version of the Chernoff bound: Let $X_1, \ldots X_N$ be independent random variables taking values in $[-1, 1]$. Let X denote their mean, and $\mu = \mathbb{E}[X]$ denote the expected value of their mean. Then,

$$\Pr[X \leqslant \mu - \epsilon] \leqslant e^{-\frac{\epsilon^2 N}{2}}$$

Therefore,

$$\begin{aligned}
&\Pr[b' = b] \\
&\geqslant \Pr[b' = b | \mathbb{H} \wedge z = b] \cdot \Pr[\mathbb{H} | z = b] \cdot \Pr[z = b] \\
&+ \Pr[b' = b | \neg\mathbb{H} \wedge z \neq b] \cdot \Pr[\neg\mathbb{H} | z \neq b] \cdot \Pr[z \neq b] \\
&\geqslant 1 \cdot \Pr[\mathbb{H} | z = b] \cdot \Pr[z = b] + 2^{-k_3} \cdot \Pr[\neg\mathbb{H} | z \neq b] \cdot \Pr[z \neq b] \\
&\qquad\qquad\qquad\qquad \text{(By substituting with Equations (17) and (18))} \\
&= 1 \cdot (1 - \mathrm{neg}(2^{k_3})) \cdot \Pr[z = b] + 2^{-k_3} \cdot (1 - \mathrm{neg}(2^{k_3})) \cdot \Pr[z \neq b] \\
&\qquad\qquad\qquad\qquad \text{(By substituting with Equations (13) and (14))} \\
&\geqslant 1 \cdot \big(1 - \mathrm{neg}(2^{k_3})\big)(2^{-k_3} - \mathrm{neg}(2^{k_3})) + 2^{-k_3} \cdot (1 - \mathrm{neg}(2^{k_3})) \cdot (1 - \mathrm{neg}(2^{k_3})) \\
&\qquad\qquad\qquad\qquad \text{(By substituting with Equation (16))} \\
&\geqslant 2^{-k_3} \cdot \big(2 - \mathrm{neg}(2^{k_3})\big) > 1.5 \cdot 2^{-k_3}.
\end{aligned}$$

This contradicts Equation (15) and completes the proof, as desired. □

Proof. To complete the proof of the theorem, it remains to show that $\mathsf{Game}_{\mathcal{A},0,\alpha}$ and $\mathsf{Game}_{\mathcal{A},1,\alpha}$ are statistically indistinguishable for both choices of $\alpha \in \{0,1\}$. The only difference between the games is that $\mathsf{Game}_{\mathcal{A},1,\alpha}$ samples x_2 conditioned on the existence of i s.t. $y = \Gamma(x_2, i) \notin \mathsf{BAD\text{-}seed}_\epsilon$, whereas $\mathsf{Game}_{\mathcal{A},0,\alpha}$ does not condition on this event. The statistical distance between $\mathsf{Game}_{\mathcal{A},1,\alpha}$ and $\mathsf{Game}_{\mathcal{A},0,\alpha}$ (for any choice of α) is therefore bounded by the following probability:

$$\Pr_{\mathsf{crs} \leftarrow \mathsf{CRS}, (x_2, \ell_{\mathsf{init}}) \leftarrow (X_2, L_{\mathsf{init}} | \mathsf{crs})} [\forall i \in [t], \Gamma(x_2, i) \in \mathsf{BAD\text{-}seed}_\epsilon]$$

We will now argue that this probability is bounded by $\mathrm{neg}(T')$. If not, then there exists a polynomial $p(\cdot)$ such that for

$$\mathbb{S} = \{x_2 : \forall i, \Gamma(x_2, i) \in \mathsf{BAD\text{-}seed}_\epsilon\}, \quad |\mathbb{S}| \geqslant \frac{2^{k_2}}{p(T')}.$$

But the disperser Γ maps every set of size at least $\frac{2^{k_2}}{T' \log T'}$ to a set of size at least 2^{d-1}. This implies that $|\{\Gamma(x_2, i)\}_{x_2 \in \mathbb{S}, i \in [t]}| \geqslant 2^{d-1}$. But this contradicts Equation (3)/Claim 21, as desired. This completes the proof of Theorem 17. □

5 Network Extractor Protocol in the CRS Model

We start with the definition of the T-admissible leaky (p, n, k)-source distribution.

Definition 24 (T-Admissible Leaky (p, n, k) Source Distribution). *A T-admissible leaky (p, n, k) source distribution with respect to a CRS distribution $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$ consists of an ensemble of sources $X = \{X_{i,\lambda}\}_{i \in [p], \lambda \in \mathbb{N}}$, and leakage $L = \{L_{i,\lambda}\}_{i \in [p], \lambda \in \mathbb{N}}$ such that for every $\lambda \in \mathbb{N}$, the following holds:*

- *For every $\mathsf{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, $\mathsf{Supp}(X_{i,\lambda} | \mathsf{crs}) \subseteq \{0,1\}^{n(\lambda)}$ for every $i \in [p]$.*
- *For every $\mathsf{crs} \in \mathsf{Supp}(\mathsf{CRS}_\lambda)$, $(X_{i,\lambda}, L_{i,\lambda} | \mathsf{crs})$ is sampleable in time $\mathrm{poly}(T(\lambda))$ for every $i \in [p]$.*
- *For every $i \in [p]$, $H_\infty(X_{i,\lambda} | \mathsf{CRS}_\lambda, L_\lambda) \geqslant k(\lambda)$ where $L_\lambda = \{L_{i,\lambda}\}_{i \in [p]}$.*
- *For every $\mathsf{crs} \in \mathsf{CRS}_\lambda$, $\ell \in \mathsf{Supp}(L_\lambda | \mathsf{crs})$ and for every distinct $i, j \in [p]$, the distributions $(X_{i,\lambda} | \mathsf{crs}, \ell)$ and $(X_{j,\lambda} | \mathsf{crs}, \ell)$ are independent.*[9]

[9] This condition follows from the way X and Y are sampled, and we add it only for the sake of being explicit.

- $\mathsf{CRSGen}(1^\lambda)$:
  1. Sample $\mathsf{CRS}_{\mathsf{NMExt}}$ for the non-malleable extractor NMExt.
  2. For each $i \in [p]$ and $b \in \{0,1\}$, sample $f_{i,b} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)$.
  3. Output $\mathsf{CRS} := (\mathsf{CRS}_{\mathsf{NMExt}}, \{f_{i,b}\}_{i \in [p], b \in \{0,1\}})$.
- **Description of the Protocol.** Party P_i on input $x_i \in \{0,1\}^n$ does the following:
  1. For each $b \in \{0,1\}$, it computes $f_{i,b}(x_i)$ and broadcasts $f_{i,1}(x_i)$.
  2. It receives $\{f_{j,1}(x_j)\}_{j \neq i}$ from the other parties.
  3. It outputs $\bigoplus_{j \neq i} \mathsf{NMExt}(f_{i,0}(x_i) \circ i, f_{j,1}(x_j) \circ j, \mathsf{CRS}_{\mathsf{NMExt}})$.

Figure 1: Network Extractor Protocol in the CRS Model

We now provide the definition of network extractor protocol in the CRS model adapting the definitions from [KLRZ08, KLR09].

Definition 25. *A protocol for p processors is a (T, t, g) network extractor with respect to CRS distribution $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$ with source length $n(\lambda)$, min-entropy $k(\lambda)$ and output length $m(\lambda)$ if for any T-admissible leaky (p, n, k) source distribution (X, L) w.r.t. $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$ (see Definition 24) and any choice M of t faulty processors, after running the protocol, there exists a set $G \in [p] \backslash T$ of size at least g such that*

$$|\mathsf{CRS}, B, \{X_i\}_{i \notin G}, \{L_i\}_{i \in [p]}, \{Z_i\}_{i \in G} - \mathsf{CRS}, B, \{X_i\}_{i \notin G}, \{L_i\}_{i \in [p]}, U_{gm}| < \mathsf{negl}(\lambda)$$

Here, $(\mathsf{CRS}, \{X_i, L_i\}_{i \in [p]}) \leftarrow (\mathsf{CRS}_\lambda, \{X_{i,\lambda}, L_{i,\lambda}\}_{i \in [p]})$, B is the transcript of the protocol and Z_i denote the output of the i-th party in the protocol, U_{gm} is the uniform distribution on gm bits independent of B, $\{X_i\}_{i \notin G}$ and $\{L_i\}_{i \in [p]}$.

5.1 Building Blocks

We use the following building blocks in the construction.

1. A (n, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{n_1(\lambda)}\}_{\lambda \in \mathbb{N}}$.
2. A (n_1, k_1) T-strong computational non-malleable extractor in the CRS model denoted by

$$\mathsf{NMExt}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$

5.2 Construction

We give the construction of the network extractor protocol in Figure 1.

Theorem 26. *Let $\gamma \in (0,1)$ be a fixed constant and let $k(\lambda)$ be an arbitrary polynomial larger than $n_1(\lambda) - w(\lambda)$. Assuming the existence of the following primitives:*

- *A (n, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{n_1(\lambda)}\}_{\lambda \in \mathbb{N}}$, where $w(\lambda) = n_1(\lambda) - (n_1(\lambda))^\gamma$.*
- *A (n_1, k_1) T-strong computational non-malleable extractor in the CRS model denoted by*

$$\mathsf{NMExt}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$

  where $k_1(\lambda) \geqslant k(\lambda) - (n_1(\lambda) - w(\lambda))$.

Then, the construction given in Figure 1 is a $(T, p-2, 2, \mathsf{negl})$ network extractor with respect to the CRS distribution in Figure 1 and min-entropy $k(\lambda)$.

5.3 Proof of Our Network Extractor

In this section, we prove Theorem 26.

Let M be an arbitrary subset of corrupted parties of size at most $t \leqslant p-2$. Let $H = [p] \backslash M$ denote the set of uncorrupted parties. Let $H = \{i_1, \ldots, i_{|H|}\}$. For each $j \in [|H|]$, we define Hyb_1 to be identical to $(\mathsf{CRS}, B, \{X_i\}_{i \notin H}, \{L_i\}_{i \in [p]}, \{Z_i\}_{i \in H})$ (see Definition 25) and Hyb_j as follows. It is same as Hyb_1 except that for every $j' < j$ such that $i_{j'} \in H$, we replace $Z_{i_{j'}}$ with U_m. We now observe that $\mathsf{Hyb}_{|H|+1}$ is identically distributed to $(\mathsf{CRS}, B, \{X_i\}_{i \notin H}, \{L_i\}_{i \in [p]}, U_{gm})$ (see Definition 25) and to complete the proof, it is sufficient to show that for every $j \in [|H|]$, Hyb_j is computationally indistinguishable to Hyb_{j+1}. We now define a sequence of hybrids to prove the above claim. Let $j' \neq j$ such that $i_{j'} \in H$. Since $t \leqslant p-2$, we infer that such a j' always exists.

- $\underline{\mathsf{Hyb}_{j,1}}$: This hybrid is same as Hyb_j except that we sample $f_{i_j,1}$ and $f_{i_{j'},0}$ as output of $\mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$. It follows immediately from the indistinguishability of the injective and the lossy modes that Hyb_j is computationally indistinguishable to $\mathsf{Hyb}_{j,1}$.
- $\underline{\mathsf{Hyb}_{j,2}}$: This hybrid is same as $\mathsf{Hyb}_{j,1}$ except that we replace Z_{i_j} with U_m. We argue in Claim 27 that $\mathsf{Hyb}_{j,1}$ is computationally indistinguishable to $\mathsf{Hyb}_{j,2}$ from the security of the non-malleable extractor.
- $\underline{\mathsf{Hyb}_{j,3}}$: In this hybrid, we reverse the changes made in $\mathsf{Hyb}_{j,1}$. Again, it follows from the indistinguishability of the lossy and the injective modes that $\mathsf{Hyb}_{j,3}$ and $\mathsf{Hyb}_{j,2}$ are computationally indistinguishable. We note that $\mathsf{Hyb}_{j,3}$ is identically distributed to Hyb_{j+1}.

Claim 27. *Assuming that* NMExt *is* (n_1, k_1) *T-strong computational non-malleable extractor in the CRS model, we have* $\mathsf{Hyb}_{j,1} \approx_c \mathsf{Hyb}_{j,2}$.

Proof. Assume for the sake of contradiction that there exists a distinguisher D that can distinguish between $\mathsf{Hyb}_{j,1}$ and $\mathsf{Hyb}_{j,2}$ with non-negligible advantage. We give a reduction that breaks the security of NMExt.

The reduction defines the CRS distribution to first sample $\mathsf{CRS}_{\mathsf{NMExt}}$ and then samples the rest of the components in CRS as in $\mathsf{Hyb}_{j,1}$. It then defines functions L'_{i_j} and $L'_{i_{j'}}$ as follows:

- $L'_{i_j}(X_{i_j})$ outputs $f_{i_j,1}(X_{i_j})$.
- $L'_{i_{j'}}(X_{i_{j'}})$ outputs $f_{i_{j'},0}(X_{i_{j'}})$.

It then provides the following sampler for sampling the sources $f_{i_1,0}(X_{i_1}) \circ i_1$ and $f_{i_2,1}(X_{i_2}) \circ i_2$ and the leakage function $\{\overline{L}_i\}_{i \in \{i_1, i_2\}}$ defined below.

- For each $i \in \{i_j, i_{j'}\}$, the sampler samples $(X_i, L_i) \leftarrow (X_{i,\lambda}, L_{i,\lambda} | \mathsf{CRS})$ from the source distribution. It defines the leakage function to be $\overline{L}_i = (L_i, L'_i(X_i))$.
- The sampler outputs $f_{i_j,0}(X_{i_j}) \circ i_j$ as the first source and $f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}$ as the second source.

The reduction obtains crs, $\ell_{i_j} = (L_{i_j}, L_{i_j}(X_{i_j}))$ and $\ell_{i_{j'}} = (L_{i_{j'}}, L'_{i_{j'}}(X_{i_{j'}})))$ from the external challenger. Now, conditioned on ℓ_{i_j} and $\ell_{i_{j'}}$, we have that X_{i_j} and $X_{i_{j'}}$ are independent sources. Since for each $f_{i_j,1}$ and $f_{i_{j'},0}$ are generated in the lossy mode, and $f_{i_j,0}$ and $f_{i_{j'},1}$ are generated in the injective mode, it follows from Claim 7 that $H_\infty(f_{i_j,0}(X_{i_j}) \circ i_j | \mathsf{crs}, \ell_{i_j}, \ell_{i_{j'}}) \geqslant k - (n_1 - w)$ and $H_\infty(f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'} | \mathsf{crs}, \ell_{i_j}, \ell_{i_{j'}}) \geqslant k - (n_1 - w)$. For every $i \in [p] \notin \{i_j, i_{j'}\}$, the reduction samples $(X_i, L_i) \leftarrow (X_{i,\lambda}, L_{i,\lambda} | \mathsf{CRS})$. The reduction now defines the tampering function g that acts on $f_{i_{j'}}(X_{i_{j'}}) \circ i_{j'}$ as follows:

- It uses the output of the leakage function ℓ_{i_j} and $\ell_{i_{j'}}$ as well as the fixing of the other sources to generate the partial transcript B' of the protocol that includes all the messages from the honest parties except $f_{i_{j'},1}(X_{i_{j'}})$. The tampering function on input $f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}$ does the following: it uses

$f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}$ to generate the messages from all the honest parties in the protocol and then runs the adversarial strategy to generate the messages from the corrupt parties. It outputs $\{X_k^{i_j} \circ k\}_{k \neq i_{j'}}$ where $X_k^{i_j}$ is the message received by i_j-th party from the k-th party (for every $k \notin \{i_j, i_{j'}\}$) in the protocol.

We note that g is efficiently computable since the adversarial strategy is efficiently computable and for any $k \neq i_{j'}$, $X_k^{i_j} \circ k \neq f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}$. Hence, g constitutes a valid tampering function against the NMExt. The reduction provides g as the tampering function to the external challenger. It obtains the output $y, f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}, \{\mathsf{NMExt}(f_{i_j,0}(X_{i_j}) \circ i_j, X_k^{i_j}, \mathsf{CRS}_{\mathsf{NMExt}})\}_{k \neq i_j}$ from the challenger and uses it to generate the output of the hybrid.

We note that if y is generated as the output of the NMExt on $f_{i_j,0}(X_{i_j}) \circ i_j, f_{i_{j'},1}(X_{i_{j'}}) \circ i_{j'}$ and $\mathsf{CRS}_{\mathsf{NMExt}}$ then the output of the reduction is identical to $\mathsf{Hyb}_{i,1}$. Else, it is distributed to $\mathsf{Hyb}_{i,2}$. Thus, if there is a distinguisher that can distinguish between $\mathsf{Hyb}_{i,1}$ and $\mathsf{Hyb}_{i,2}$ with non-negligible advantage, then we can use the same distinguisher to break the security of NMExt and this is a contradiction. □

5.4 Instantiation

We instantiate the non-malleable extractor from Corollary 19 and the lossy functions from [PW08, BHK11] Specifically, we set the constant c of the non-malleable extractor to be $\max(c_0, c_1)$ (where c_1 is the parameter for the lossy functions). Thus, we obtain the following corollary.

Corollary 28. *Assuming the sub-exponential hardness of the DDH assumption, there exist constants $c > 1$ and c' such that for any p number of players, there exists a construction of $(\lambda, p-2, 2)$ network extractor protocol in the CRS model with sources of length $\Omega(\lambda) \leqslant n(\lambda) \leqslant \mathrm{poly}(\lambda)$, min-entropy $O(\log \lambda)^c$ and output length $O(\log \lambda)^{c'}$.*

6 Extractor for Adversarial Sources in the CRS Model

We start with the definition of the adversarial source distribution.

Definition 29. *A T-admissible leaky (p, n, k) adversarial sources with respect to CRS distribution $\{\mathsf{CRS}_\lambda\}_\lambda$ is a tuple $(i, j, (X, Y, L), I, \{x_k\}_{k \in I}, I_i, I_j, \{f_k\}_{k \in I_i \cup I_j})$ where $i, j \in [p]$, (X, Y, L) is T-admissible leaky (n, k)-source distribution w.r.t. $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$, $I \cup I_i \cup I_j = [p]$ and $f_k : \{0,1\}^n \to \{0,1\}^n$ are T-time computable functions.*

We now give the definition of the extractor for adversarial sources below.

Definition 30. *For any $p \in \mathbb{N}$, and functions $n = n(\lambda)$, $c = c(\lambda)$ and $m = m(\lambda)$, a function ensemble $\mathsf{AdvExt} = \{\mathsf{AdvExt}_\lambda\}_{\lambda \in \mathbb{N}}$, where*

$$\mathsf{AdvExt}_\lambda : (\{0,1\}^{n(\lambda)})^p \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$

is said to be a (p, n, k) T-computational adversarial source extractor in the CRS model if there exists an ensemble $\{\mathsf{CRS}_\lambda\}_{\lambda \in \mathbb{N}}$ such that the following holds:

For every T-admissible leaky (p, n, k) adversarial sources $(i, j, (X, Y, L), I, \{x_k\}_{k \in I}, I_i, I_j, \{f_k\}_{k \in I_i \cup I_j})$ wrt CRS, the following two distributions are computationally indistinguishable:

$$\{\mathsf{AdvExt}_\lambda((x'_1, \ldots, x'_p), \mathsf{crs}), \mathsf{crs}, \ell\} \approx_c \{U_m, \mathsf{crs}, \ell\}$$

where $\mathsf{crs} \leftarrow \mathsf{CRS}_\lambda$, $(x_i, x_j, \ell) \leftarrow (X, Y, L|\mathsf{crs})$, for every $k \in I$, $x'_k = x_k$, for every $k \in I_i$, $x'_k = f_k(x_i)$, and for every $k \in I_j$, $x'_k = f_k(x_j)$.

- $\mathsf{CRSGen}(1^\lambda)$:
  1. Sample $\mathsf{CRS}_{\mathsf{cnm\text{-}Ext}}$ for the non-malleable extractor $\mathsf{cnm\text{-}Ext}$.
  2. For each $i \in [p]$ and $b \in \{0,1\}$, sample $f_{i,b} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)$.
  3. Output $\mathsf{CRS} := (\mathsf{CRS}_{\mathsf{cnm\text{-}Ext}}, \{f_{i,b}\}_{i\in[p],b\in\{0,1\}})$.
- **Description of the Extractor.** On input $(x_1, \ldots, x_p) \in (\{0,1\}^n)^p$, the extractor does the following:
  1. For each $j \in [p]$ and $b \in \{0,1\}$, it computes $f_{j,b}(x_j)$.
  2. For each $i \in [p]$, it computes $r_i := \bigoplus_{j\neq i} \mathsf{cnm\text{-}Ext}(f_{i,0}(x_i) \circ i, f_{j,1}(x_j) \circ j, \mathsf{CRS}_{\mathsf{cnm\text{-}Ext}})$.
  3. It outputs $\bigoplus_{i\in[p]} r_i$.

Figure 2: Extractor for Adversarial Sources

6.1 Building Blocks

We use the following building blocks in the construction.

1. A (n, n_1, w)-lossy trapdoor function family $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{n_1(\lambda)}\}_{\lambda\in\mathbb{N}}$.
2. A (n_1, k_1) T-strong computational 2-source extractor in the CRS model denoted by
$$\mathsf{cnm\text{-}Ext}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$

6.2 Construction

We give the construction of our extractor for adversarial sources in Figure 2.

Theorem 31. *Let $p \in \mathbb{N}$ be fixed and let $m(\cdot)$ be an arbitrary polynomial. Let $k(\cdot)$ be an arbitrary polynomial such that for every $\lambda \in \mathbb{N}$, $k(\lambda) \geqslant (2p-1)(n_1(\lambda) - w(\lambda)) + m(\lambda)$. Let $n(\cdot)$ be another polynomial such that $n(\lambda) \geqslant k(\lambda)$ for every $\lambda \in \mathbb{N}$. Assuming the existence of the following primitives:*

- *A (n, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{n_1(\lambda)}\}_{\lambda\in\mathbb{N}}$.*
- *A (n_1, k_1) T-strong computational non-malleable extractor in the CRS model denoted by*
$$\mathsf{cnm\text{-}Ext}_\lambda : \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{n_1(\lambda)} \times \{0,1\}^{c(\lambda)} \to \{0,1\}^{m(\lambda)}$$
where $k_1(\lambda) \geqslant k(\lambda) - (2p-1)(n_1(\lambda) - w(\lambda)) - m(\lambda)$.

Then, the construction given in Figure 2 is a (p, n, k) adversarial source extractor with respect to the CRS distribution described in Figure 2.

6.3 Proof of Our Extractor for Adversarial Sources

We now prove Theorem 31.

Let $(i_1, i_2, (X, Y, L), I, \{x_k\}_{k\in I}, I_1, I_2, \{f_k\}_{k\in I_1\cup I_2})$ be an arbitrary T-admissible leaky adversarial sources with respect to CRS distribution described in Figure 2. We define Hyb_1 to be identical to the output of the extractor and Hyb_2 as follows. It is same as Hyb_1 except that it replaces r_{i_1} in the computation of the extractor with U_m. We now observe that Hyb_2 is identically distributed to the uniform distribution since r_i is uniform and independent of the other $\{r_k\}_{k\neq i_1}$. To complete the proof, it is sufficient to show that Hyb_2 is computationally indistinguishable to Hyb_1. We now define a sequence of hybrids to prove the above claim.

- $\underline{\mathsf{Hyb}_{1,1}}$: This hybrid is same as Hyb_1 except that for every $(i,b) \notin \{(i_1,0),(i_2,1)\}$, we sample $f_{i,b}$ the as output of $\mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$. It follows immediately from the indistinguishability of the injective and the lossy modes that Hyb_1 is computationally indistinguishable to $\mathsf{Hyb}_{1,1}$.
- $\underline{\mathsf{Hyb}_{1,2}}$: This hybrid is same as $\mathsf{Hyb}_{1,1}$ except that we replace r_{i_1} with U_m. We argue in Claim 32 that $\mathsf{Hyb}_{1,1}$ is computationally indistinguishable to $\mathsf{Hyb}_{1,2}$ from the security of the non-malleable extractor.
- $\underline{\mathsf{Hyb}_{1,3}}$: In this hybrid, we reverse the changes made in $\mathsf{Hyb}_{1,1}$. Again, it follows from the indistinguishability of the lossy and the injective modes that $\mathsf{Hyb}_{1,3}$ and $\mathsf{Hyb}_{1,2}$ are computationally indistinguishable. We note that $\mathsf{Hyb}_{1,3}$ is identically distributed to Hyb_2.

This completes the proof of theorem.

Claim 32. *Assuming that* cnm-Ext *is* (n_1,k_1) *T-strong computational 2-source extractor in the CRS model, we have* $\mathsf{Hyb}_{1,1} \approx_c \mathsf{Hyb}_{1,2}$.

Proof. Assume for the sake of contradiction that there exists a distinguisher D that can distinguish between $\mathsf{Hyb}_{1,1}$ and $\mathsf{Hyb}_{1,2}$ with non-negligible advantage. We give a reduction that breaks the security of cnm-Ext.

The reduction defines the CRS distribution to first sample $\mathsf{CRS}_{\mathsf{cnm\text{-}Ext}}$ and then sample the rest of the components in CRS as in $\mathsf{Hyb}_{1,1}$. It defines the leakage function $L'_{\mathsf{init}} = (L_{\mathsf{init}}, L_{i_2}(\cdot))$ and $L'_{\mathsf{final}} = (L_{\mathsf{final}}, L_{i_1}(\cdot))$ where $L_{i_2}(\cdot)$ and $L_{i_1}(\cdot)$ are defined below.

- L_{i_2} takes X_{i_2} as input and outputs $(f_{i_2,0}(X_{i_2}), \{f_{i,b}(X_i)\}_{i \in I_2, b \in \{0,1\}})$.
- The leakage function L_{i_i} takes X_{i_1} and first computes $(f_{i_1,1}(X_{i_1}), \{f_{i,b}(X_i)\}_{i \in I_1, b \in \{0,1\}})$. It finally computes $\oplus_{j \neq \{i_1,i_2\}} \mathsf{cnm\text{-}Ext}(f_{i_1,0}(X_{i_1}) \circ i_1, f_{j,1}(X_j) \circ i_j)$.

  It outputs $(f_{i_1,1}(X_{i_1}), \{f_{i,b}(X_i)\}_{i \in I_1, b \in \{0,1\}}, \oplus_{j \neq \{i_1,i_2\}} \mathsf{cnm\text{-}Ext}(f_{i_1,0}(X_{i_1}) \circ i_1, f_{j,1}(X_j) \circ i_j))$.

The reduction then provides the following sampler for sampling the sources $f_{i_1,0}(X_{i_1}) \circ i_1$ and $f_{i_2,1}(X_{i_2}) \circ i_2$.

- The sampler samples $(y, \ell'_{\mathsf{init}}) \leftarrow (Y, L'_{\mathsf{init}} | \mathsf{crs})$. It then samples $(x, \ell'_{\mathsf{final}}) \leftarrow (X, L'_{\mathsf{final}} | \mathsf{crs}, \ell'_{\mathsf{init}})$.
- The sampler outputs $f_{i_1,0}(x) \circ i_1$ as the first source and $f_{i_2,1}(y) \circ i_2$ as the second source.

Now, conditioned on ℓ'_{init} and ℓ'_{final}, we have that $f_{i_1,0}(X_{i_1}) \circ i_1$ and $f_{i_2,1}(X_{i_2}) \circ i_2$ are independent sources. Since for each $(i,b) \notin \{(i_1,0),(i_2,1)\}$, $f_{i,b}$ is generated in the lossy mode, it follows from Claim 7 that $H_\infty(f_{i_1,0}(X_{i_1}) \circ i_1 | \ell'_{\mathsf{init}}, \ell'_{\mathsf{final}}) \geqslant k - (2|I_1|+1)(n_1 - w) - m$ and $H_\infty(f_{i_2,1}(X_{i_2)} \circ i_2 | \ell'_{\mathsf{init}}, \ell'_{\mathsf{final}}) \geqslant k - (2|I_2|+1)(n_1 - w)$.

We obtain the challenge $y, \ell', \mathsf{aux}, \mathsf{crs}$ from the external challenger and uses it to generate $\{r_j\}_{j \in [p]}$. The reduction finally computes the output of the hybrid.

We note that if y is generated as the output of the cnm-Ext then the output of the reduction is identical to $\mathsf{Hyb}_{1,1}$. Else, it is distributed identically to $\mathsf{Hyb}_{1,2}$. Thus, if there is a distinguisher that can distinguish between $\mathsf{Hyb}_{1,1}$ and $\mathsf{Hyb}_{1,2}$ with non-negligible advantage, then we can use the same distinguisher to break the security of cnm-Ext and this is a contradiction.

□

6.4 Instantiation

We instantiate the two-source extractor from Corollary 18 and the lossy functions from [PW08, BHK11]. Specifically, for any fixed p, we set c for the two-source extractor to be large enough such that min-entropy of the two source extractor $(2p-1)O(\log^{c_1} \lambda) < \log^c \lambda$. We set $m(\lambda) < \log^c \lambda$. We, thus, obtain the following corollary.

Corollary 33. *Fix any $p \in \mathbb{N}$. Assuming the sub-exponential hardness of DDH assumption, there exists constants $c > 1$ and $c' < c$ such that for any $\Omega(\lambda) \leqslant n(\lambda) \leqslant \mathrm{poly}(\lambda)$, $k(\lambda) = O(\log^c \lambda)$ and $m(\lambda) \leqslant O(k(\lambda))$, there exists a construction of a (p, n, k) λ-computational adversarial two-source extractor in the CRS model with output length $O(\log \lambda)^{c'}$.*

References

[AOR$^+$20a] Divesh Aggarwal, Maciej Obremski, João L. Ribeiro, Mark Simkin, and Luisa Siniscalchi. Two-source non-malleable extractors and applications to privacy amplification with tamperable memory. *IACR Cryptol. ePrint Arch.*, 2020:1371, 2020. URL: https://eprint.iacr.org/2020/1371.

[AOR$^+$20b] Divesh Aggarwal, Maciej Obremski, João L. Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 343–372. Springer, 2020. doi:10.1007/978-3-030-45721-1_13.

[BACD$^+$17] Avraham Ben-Aroya, Eshan Chattopadhyay, Dean Doron, Xin Li, and Amnon Ta-Shma. Low-error, two-source extractors assuming efficient non-malleable extractors. *CCC*, 2017.

[BADTS16] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. Explicit two-source extractors for near-logarithmic min-entropy. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 88, 2016.

[BHK11] Mark Braverman, Avinatan Hassidim, and Yael Tauman Kalai. Leaky pseudo-entropy functions. *Innovations in Computer Science*, 2011.

[Bou05] Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1:1–32, 2005.

[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. doi:10.1137/0217015.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1184–1197. ACM, 2020. doi:10.1145/3357713.3384339.

[Coh16a] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM Journal on Computing*, 45(4):1297–1338, 2016.

[Coh16b] Gil Cohen. Making the most of advice: New correlation breakers and their applications. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 188–196. IEEE, 2016.

[Coh16c] Gil Cohen. Non-malleable extractors-new tools and improved constructions. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[Coh16d] Gil Cohen. Two-source extractors for quasi-logarithmic min-entropy and improved privacy amplification protocols. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 114, 2016.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 670–683. ACM, 2016.

[DO03] Yevgeniy Dodis and Roberto Oliveira. On extracting private randomness over a public channel. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *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*, volume 2764 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2003. doi:10.1007/978-3-540-45198-3_22.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 196–205, 2004. doi:10.1109/FOCS.2004.44.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008. doi:10.1137/060651380.

[DVW20] Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 313–342. Springer, 2020. doi:10.1007/978-3-030-45721-1_12.

[GKK20] Ankit Garg, Yael Tauman Kalai, and Dakshita Khurana. Low error efficient computational extractors in the CRS model. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 373–402. Springer, 2020. doi:10.1007/978-3-030-45721-1_14.

[GSV05] Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In Pierre Fraigniaud, editor, *Distributed Computing, 19th International Conference, DISC 2005, Cracow, Poland, September 26-29, 2005, Proceedings*, volume 3724 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2005. doi:10.1007/11561927_22.

[GSZ21] Vipul Goyal, Akshayaram Srinivasan, and Chenzhi Zhu. Multi-source non-malleable extractors and applications. *To appear in Eurocrypt*, 2021:157, 2021. URL: https://eprint.iacr.org/2020/157.

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

[KLR09] Yael Tauman Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 617–626. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.61.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 654–663. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.73.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 168–177. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.26.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1156. ACM, 2017.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *STOC*, pages 187–196, 2008.

[Raz05] Ran Raz. Extractors with weak random seeds. *STOC*, pages 11–20, 2005.