Computational Extractors with Negligible Error in the CRS Model

Ankit Garg * Yael Tauman Kalai† Dakshita Khurana‡

September 30, 2019

Abstract

In recent years, there has been exciting progress on building two-source extractors for sources with low min-entropy. Unfortunately, all known explicit constructions of two-source extractors in the low entropy regime suffer from non-negligible error, and building such extractors with negligible error remains an open problem. We investigate this problem in the computational setting, and obtain the following results.

We construct an explicit 2-source extractor, and even an explicit non-malleable extractor, with negligible error, for sources with low min-entropy, under computational assumptions in the Common Random String (CRS) model. More specifically, we assume that a CRS is generated once and for all, and allow the min-entropy sources to depend on the CRS. We obtain our constructions by using the following transformations.

1. Building on the technique of [BHK11], we show a general transformation for converting any computational 2-source extractor (in the CRS model) into a computational non-malleable extractor (in the CRS model), for sources with similar min-entropy.

   We emphasize that the resulting computational non-malleable extractor is resilient to *arbitrarily many* tampering attacks (a property that is impossible to achieve information theoretically). This may be of independent interest.

   This transformation uses cryptography, and relies on the sub-exponential hardness of the Decisional Diffie Hellman (DDH) assumption.

2. Next, using the blueprint of [BACD+17], we give a general transformation converting any computational non-malleable *seeded* extractor (in the CRS model) into a computational 2-source extractor for sources with low min-entropy (in the CRS model).

   This transformation does not incur any additional assumptions. Our analysis makes a novel use of the leakage lemma of Gentry and Wichs [GW11].

*Microsoft Research India, email: garga@microsoft.com. Work done in part while at Microsoft Research, New England.

†Microsoft Research New England, email: yael@microsoft.com.

‡University of Illinois, Urbana-Champaign email: dakshita@illinois.edu. Work done in part while at Microsoft Research, New England.

Contents

1 Introduction

Randomness is fundamental for cryptography. It is well known that even the most basic cryptographic primitives, such as semantically secure encryption, commitments and zero-knowledge proofs, require randomness. Moreover, Dodis *et al.* [DOPS04] proved that these primitives require *perfect* randomness, and cannot be constructed using a weak source of randomness, not even one that has nearly full min-entropy.[1]

Unfortunately, in reality, perfect randomness is very hard to come by, and *secret* randomness is even harder. Indeed, several attacks on cryptographic systems rely on the fact that the randomness that was used in the implementation was imperfect. Very recently, this was demonstrated in the regime of cryptocurrencies by Breitner and Heninger [BH19], who computed hundreds of Bitcoin private keys by exploiting the fact that the randomness used to generate them was imperfect (other examples include [HDWH12, BCC+13]).

Randomness Extractors. These attacks give rise to a very natural question: Can we take weak sources of randomness and "boost" them into perfect random sources? This is the basic question that underlies the field of randomness extractors. Extractors are algorithms that extract perfect randomness from weak random sources. As eluded to above, one cannot hope to deterministically take only a single weak random source and generate perfect randomness from it.

Nevertheless, two common types of randomness extractors were considered in the literature. The first is a *seeded extractor*, which uses a uniform seed to extract randomness from any (n, k) source, for k as small as $k = \text{polylog}(n)$. This seed is typically very short, often of length $O(\log n)$. However, it is paramount that this seed is perfectly random, and independent of the source. In reality, unfortunately, even generating such short perfectly random strings may be challenging.

The second type of extractor is a 2-source extractor. A 2-source extractor takes as input two *independent* weak sources and outputs pure randomness. We stress that a 2-source extractor does not require perfect randomness at all! It only requires two independent sources with sufficiently large min-entropy. Such sources may be arguably easier to generate.

Until recently, we had an explicit construction of a 2-source extractor only in the high-entropy regime, i.e. assuming one of the sources has min-entropy $k \geqslant 0.499n$ [Raz05, Bou05]. Over the last three years, there has been remarkable and exciting progress [CGL16, CZ16, BADTS16, Coh16c, Coh16b, Coh16a, Coh16d, CL16, Li17], giving rise to 2-source extractors in the low-entropy regime, albeit with non-negligible error.

More formally, an $(n_1, n_2, k_1, k_2, \epsilon)$ 2-source extractor is a function $E : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ such that for any independent sources X and Y, with min-entropy at least k_1 and k_2 respectively, $E(X, Y)$ is ϵ-close (in statistical distance) to the uniform distribution over $\{0,1\}^m$. The line of recent breakthroughs discussed above can support min-entropy as small as $O(\log(n)\log(\log(n)))$ in the balanced regime $n_1 = n_2 = n$. *However, in all the above constructions, the running time of the extractor is proportional to* $\text{poly}(1/\epsilon)$*!*

This state-of-the-art is far from ideal for cryptographic applications, where typically the error is required to be negligible in the security parameter. Unfortunately, in the negligible error regime, the extractors mentioned above run in super-polynomial time. The question of whether one can

[1] A weak source is modeled as an (n, k)-source, which is a distribution that generates elements in $\{0,1\}^n$ with min-entropy k. A distribtion $X \subseteq \{0,1\}^n$ is said to have min-entropy k if for every $x \in \{0,1\}^n$, $\Pr[X = x] \leqslant 2^{-k}$.

obtain a 2-source extractor with negligible error, even for sources with min-entropy δn, for a small constant $\delta > 0$, is one of the most important open problems in the area of randomness extractors.

In this work, we explore this problem in the computational setting. We note that solving this problem, even in the computational setting, may facilitate generating useful randomness for many cryptographic applications.

1.1 Prior work on Computational extractors

There has been some prior work [KLRZ08, KLR09] on building computational extractors. However, these works rely on extremely strong computational assumptions. Loosely speaking, the assumption is (slightly stronger than) assuming the existence of an "optimally exponentially hard" one-way permutation $f : \{0,1\}^n \to \{0,1\}^n$, that is hard to invert even with probability $2^{-(1-\delta)n}$ (this gives extractors for sources with min-entropy roughly δn).

Intuitively, such a strong assumption seems to be necessary. This is the case since to prove security we need to construct a reduction that uses an adversary $\mathcal{A}$, that breaks the 2-source extractor, to break the underlying assumption. If this assumption is a standard one, then the challenge provided by the assumption comes from a specific distribution (often the uniform distribution). On the other hand, the adversary $\mathcal{A}$ may break the extractor w.r.t. *arbitrary* independent sources X and Y with sufficient min-entropy. It is completely unclear how one could possibly use $(X, Y, \mathcal{A})$ to break this challenge, since $\mathcal{A}$ only helps to distinguish the specific distribution $E(X, Y)$ from uniform (where E is the 2-source extractor). Since X and Y are *arbitrary* low min-entropy distributions, it is unclear how one could embed the challenge in X or Y, or in $E(X, Y)$.

1.2 Our Results

In this paper, we get around this barrier by resorting to the Common Random String (CRS) model.[2] As a result, under the sub-exponential hardness of DDH (which is a comparitively mild assumption), we obtain a computational 2-source extractor, and a computational non-malleable extractor, both with negligible error, for low min-entropy sources (in the CRS model).

At first one may think that constructing such extractors in the CRS model is trivial since the CRS can be used a seed. However, as mentioned above, we emphasize that this is not the case, since the CRS is fixed once and for all, and the sources can depend on this CRS. Indeed, constructing an information theoretic 2-source extractor in the CRS model is an interesting open problem.

Secondly, one can argue that assuming the existence of a CRS is unreasonable, since our starting point is the belief that fresh randomness is hard to generate, and thus in a sense assuming a CRS brings us back to square one. However, as emphasized above, this CRS is generated once and for all, and can be reused over and over again. Indeed, we believe that true randomness is hard, yet not impossible, to generate. Thus, reducing the need for true randomness to a single one-time need, is significant progress. Importantly, we emphasize that in cryptography, there are many natural applications where a CRS is assumed to exist, and in such applications this same CRS can be used to extract randomness from weak sources.

The computational CRS model. In our constructions, we assume that a CRS is (efficiently) generated once and for all. We consider any two weak sources X and Y. These sources *can each*

[2] Jumping ahead, we note that in the proof we break the assumption by embedding the challenge in the CRS.

depend on the CRS,[3] but are required to be independent from each other, and each have sufficient min-entropy, conditioned on the CRS. We require that X and Y are efficiently sampleable given the CRS. This is needed since we are in the computational setting, and in particular, security breaks down if the sources can be used to break our hardness assumption.

Our 2-source extractor. We define an (n_1, n_2, k_1, k_2) computational 2-source extractor (in the CRS model) as a function $E : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^c \to \{0,1\}^m$ such that for all sources (X, Y), which conditioned on the crs, are independent, are polynomially sampleable, and have min-entropy at least k_1, k_2 respectively, it holds that $(E(X, Y, \mathsf{crs}), Y, \mathsf{crs})$ is computationally indistinguishable from (U, Y, crs), namely, any polynomial size adversary cannot distinguish $(E(X, Y, \mathsf{crs}), Y, \mathsf{crs})$ from (U, Y, crs) with non-negligible advantage.[4]

We construct such a 2-source extractor (with unbalanced sources) assuming the sub-exponential security of DDH.[5]

Theorem 1.1 (Informal)**.** *Let $\lambda \in \mathbb{N}$ denote the security parameter and assume the sub-exponential hardness of DDH. For every constant $\epsilon > 0$, there exist constants $\delta > 0, c > 1$ such that there exists an explicit (n_1, n_2, k_1, k_2) computational 2-source extractor in the CRS model, with $n_1 = \Omega(\lambda), n_2 \leqslant \lambda^\delta$ and min-entropy $k_1 = n_1^\epsilon, k_2 = \log^c(\lambda)$.*

Our non-malleable extractor. We also construct a computational non-malleable extractor in the CRS model. A non-malleable extractor is a notion that was introduced by Dodis and Wichs [DW09]. This notion is motivated by cryptography, and was used to achieve *privacy amplification*, i.e., to "boost" a private weak key into a private uniform one.

Similar to standard extractors, one can consider non-malleable extractors both in the seeded setting and in 2-source setting. The seeded version is defined as follows: A strong (k, ϵ) t-non-malleable-extractor is a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ s.t. for all functions $f_1, \ldots, f_t : \{0,1\}^d \to \{0,1\}^d$, that have no fixed points, it holds that

$$(Y, E(X, Y), E(X, f_1(Y)), \ldots, E(X, f_t(Y))) \equiv_\epsilon (Y, U, E(X, f_1(Y)), \ldots, E(X, f_t(Y)))$$

where X, Y, U are independent, X has min-entropy at least k, Y is distributed uniformly over $\{0,1\}^d$ and U is distributed uniformly over $\{0,1\}^m$. Non-malleable 2-source extractors are defined similarly to seeded ones, except that the requirement that Y is uniformly distributed is relaxed; i.e., it is only required to have sufficient min-entropy and be independent of X. In addition, both the sources can be tampered independently.

Clearly, in the information theoretic setting, such non-malleable extractors (both seeded and 2-sources ones) can exist only for a bounded t.

In this work we construct a computational analogue of a non-malleable extractor in the CRS model. As opposed to the information theoretic setting, where the number of tampering attacks t is a-priori bounded, in the computational setting we allow the adversary to tamper an *arbitrary*

[3] In this way, the CRS is different from the seed of a seeded extractor, which must be completely independent of the source.

[4] Requiring the output of the extractor to be random even given the source Y is a standard requirement, and such an extractor is known as a *strong* extractor.

[5] The sub-exponential DDH assumption asserts that there exists a group G such that no sub-exponential time algorithm can distringuish between (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where g is a fixed generator of G, and where a, b, c are chosen randomly from $\mathbb{Z}_q$, where q denotes the order of G.

(polynomial) number of times (i.e., we do not fix an a priori bound t on the number of tampering functions). In fact, in addition to giving the adversary $Y, E(X, Y)$, we can even give the adversary access to an oracle that on input $Y' \neq Y$, outputs $E(X, Y')$.

We would like to note that the object we construct is somewhere in between a seeded and a 2-source non-malleable extractor. While the source Y need not be uniformly distributed, we only allow tampering with Y, and do not allow tampering with the other source.

More formally, we define an (n_1, n_2, k_1, k_2) computational non-malleable extractor (in the CRS model) as a function $E : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^c \to \{0,1\}^m$ such that for all sources X, Y that are polynomially sampleable, are independent, and have min-entropy at least k_1 and k_2 respectively, conditioned on the CRS, it holds that $(E(X, Y, \mathsf{CRS}), \mathsf{CRS}, Y)$ is computationally indistinguishable from (U, CRS, Y), even with respect to PPT adversaries that are given access to an oracle that on input $Y' \neq Y$ outputs $E(X, Y', \mathsf{CRS})$. Clearly, such adversaries can obtain $E(X, Y', \mathsf{CRS})$ for an arbitrary $t = \mathrm{poly}(n)$ number of different samples Y', that depend on Y and the CRS.

In this setting, we obtain the following two incomparable results, in the high and low min-entropy regimes respectively.

Theorem 1.2 (Informal)**.** *Let $\lambda \in \mathbb{N}$ denote the security parameter and assume the sub-exponential security of DDH. For every constant $\epsilon > 0$, there exists a constant $c > 0$ such that there exists an explicit (n_1, n_2, k_1, k_2) computational non-malleable extractor resisting arbitrarily polynomial tamperings where:*

$$n_1 = \Omega(\lambda), \log^c \lambda \leqslant n_2, k_1 = n_1^{\epsilon}, k_2 = 0.51 n_2$$

Theorem 1.3 (Informal)**.** *Let $\lambda \in \mathbb{N}$ denote the security parameter and assume the sub-exponential security of DDH. For every constant $\epsilon > 0$, there exist constants $\delta, c > 0$ such that there exists an explicit (n_1, n_2, k_1, k_2) computational non-malleable extractor resisting arbitrarily polynomial tamperings, where:*

$$n_1 = \Omega(\lambda), \log^c \lambda \leqslant n_2 \leqslant \lambda^{\delta}, k_1 = n_1^{\epsilon}, k_2 = \log^c n_2$$

We mention that in our formal theorems, under the sub-exponential hardness of DDH, we allow the sources to be sampled in super-polynomial time and the adversary to run in super-polynomial time. We refer the reader to Section 5 and 6 for more details.

2 Our Techniques

We obtain our results in three steps.

1. We first construct a computational non-malleable extractor in the CRS model, for sources in the *high entropy* regime (i.e., assuming one of the sources has min entropy rate larger than $1/2$). Our construction follows the blueprint of [BHK11], who built leaky pseudo-entropy functions based on the sub-exponential hardness of DDH. When viewed differently, their construction can be framed as showing how to use cryptography to convert any (information theoretic) 2-source extractor (with negligible error) into a computational non-malleable extractor in the CRS model (for sources with roughly the same min-entropy as in the underlying 2-source extractor). Since we only have information theoretic 2-source extractors for sources in the high entropy regime, we obtain a computational non-malleable extractor (in the CRS model) for sources in the high entropy regime.

Importantly, this extractor is non-malleable w.r.t. *arbitrarily many* tampering functions (a property that is impossible to achieve information theoretically). This contribution is mainly conceptual.

2. We then describe how this extractor can be used to obtain a computational 2-source extractor (in the CRS model) with negligible error for *low min-entropy* sources. This part contains the bulk of the technical difficulty of this work. Specifically, we follow the blueprint of [BACD+17], which shows how to convert any (information-theoretic) non-malleable extractor into a 2-source extractor (with negligible error for low min-entropy sources). However, this transformation assumes that the non-malleable extractor has a somewhat optimal dependence between the seed length and the allowable number of tampering functions. Prior to our work, no explicit constructions of non-malleable extractors were known to satisfy this requirement.

   Our computational non-malleable extractor does satisfy this requirement, and therefore we manage to use the [BACD+17] blueprint to construct the desired 2-source extractor. Nevertheless, there are multiple unique challenges that come up when trying to apply their transformation in the computational setting. One of our key ideas to overcome these challenges involves using the leakage lemma of Gentry and Wichs [GW11]. We elaborate on this in Section 2.2.

3. To achieve our final construction of a computational non-malleable extractor (in the CRS model) with negligible error for *low min-entropy* sources, we again use the blueprint from [BHK11], however, this time we use our *computational* 2-source extractor as a building block. To argue security, we prove that the [BHK11] transformation goes through even if we start with a *computational* 2-source extractor. As above, many techincal challanges arise when considering the computational setting.

2.1 From 2-Source Extractors to Non-Malleable Extractors

We begin with the observation that the construction of leaky psuedo-random functions from [BHK11], can be framed more generally as a cryptographic reduction from (information theoretic) 2-source extractors to computational non-malleable extractors in the CRS model. Since we only know information theoretic 2-source extractors (with negligible error) in the high-entropy regime, we obtain a computational non-malleable extractor (in the CRS model) in the high entropy regime.

Moreover, we generalize the [BHK11] blueprint, by showing that one can convert any *computational* 2-source extractor (in the CRS model) to a computational non-malleable extractor (in the CRS model). This introduces several techinical difficulties which we elaborate on in Section 5. This generalization is needed to obtain our final result, of a computational non-malleable extractor (in the CRS model) for sources with low min-entropy (i.e., to achieve Item 3 in the overview above).

We next describe our interpretation of the [BHK11] blueprint for converting any (information theoretic) 2-source extractor into a computational non-malleable one (in the CRS model):

Start with any 2-source extractor

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

with negleigle error (eg., [Bou05, Raz05]).

Assume the existence of the following two cryptographic primitives:

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.

A collision resistant hash family has the guarantee that given a random function $h \leftarrow \mathcal{H}$ it is hard to find two distinct elements $y_1, y_2 \in \{0,1\}^{n_2}$ such that $h(y_1) = h(y_2)$.

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., the image size is very small). It is assumed that it is hard to distinguish between a random injective function and a random lossy function in the family.

We note that both these primitive can be constructed under the DDH assumption, which is a standard cryptographic assumption.[6]

We next show how these cryptographic primites can be used to convert 2Ext into a computational non-malleable 2-source extractor in the CRS model. We start by describing the CRS.

The CRS consists of a random function $h \leftarrow \mathcal{H}$ from the collision-resistant hash family, and consists of $2k$ random injective functions from the lossy function family $\mathcal{F}$, denoted by

$$f_{1,0}, f_{2,0}, \ldots, f_{k,0}$$
$$f_{1,1}, f_{2,1}, \ldots, f_{k,1}$$

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

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

In what follows, we recall the proof idea from [BHK11]. To this end, consider any polynomial size adversary $\mathcal{A}$ that obtains either $(\mathsf{cnm\text{-}Ext}(x, y), y, \mathsf{crs})$ or (U, y, crs), together with an oracle $\mathcal{O}$ that has (x, y, crs) hardwired, and on input y' outputs $\perp$ if $y' = y$, and otherwise outputs $\mathsf{nm\text{-}Ext}(x, y', \mathsf{crs})$. 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 $(\mathsf{crs}, h(y), x)$ and on input y' outputs $\perp$ if $h(y') = h(y)$, and otherwise outputs $\mathsf{cnm\text{-}Ext}(x, y')$.

A key observation is that access to this oracle can be simulated entirely given only $\mathsf{crs}, h(y)$ and $(Z_1, \ldots Z_k)$, where

[6] The DDH assumption asserts that there exists a group G such that (g^a, g^b, g^{ab}) is computationally indistinguishable from (g^a, g^b, g^c), where g is a fixed generator of G, and where a, b, c are chosen randomly from $\mathbb{Z}_q$, where q denotes the order of G.

$$\begin{aligned} Z_k &\triangleq f_{k,1-h(y)_k}(x) \\ Z_{k-1} &\triangleq f_{k-1,1-h(y)_{k-1}}(f_{k,h(y)_k}(x)) \\ &\vdots \\ Z_1 &\triangleq f_{1,1-h(y)_1}(f_{2,h(y)_2}(\ldots f_{k,h(y)_k}(x))) \end{aligned}$$

Since the adversary $\mathcal{A}$ cannot distinguish between random injective functions and random lossy ones, we can 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. By setting k (the size of the output of the hash) to be small enough, we can guarantee that Y has high min-entropy conditioned on $h(y), Z = (Z_1, \ldots, Z_k)$. Moreover, it is easy to seet that X and Y remain independent conditioned on $h(Y), Z$. Thus, we can 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.

This was, of course, a very simplified overview. A careful reader may have observed a circularity in the intuition above: Recall that we sample the crs such that for $b = h(y)$, the functions $f_{1,b_1}, \ldots, f_{k,b_k}$ are injective, whereas $f_{1,1-b_1}, \ldots, f_{k,1-b_k}$ are lossy. Thus, the crs implicitly depends on y (via $b = h(y)$). This results in a circularity, because y is then sampled as a function of this crs, and hence may not satsify that $b = h(y)$. The formal proof requires us to carefully deal with this (and other) dependency issues that arise when formalizing this intuition. In a nutshell, we overcome this circularity by strengthening our assumption to a sub-exponenential one, namely we assume the sub-exponential hardness of DDH as opposed to the (more standard) polynomial hardness of DDH.

In addition, as mentioned above, we prove that this transformation goes through even if the underlying 2-source extractor is a *computational* one (in the CRS model). This introduces various other technical difficulties. We refer the reader to Section 5 for the details.

2.2 Our 2-source extractor.

As mentioned earlier, we construct our computational 2-source extractor by following the blueprint of [BACD+17], which shows how to use a non-malleable seeded extractor to construct a 2-source extractor (in the low entropy regime). However, they need the non-malleable seeded extractor to have the property that the seed length is significantly smaller than $t \log(1/\epsilon)$, where t is the number of tampering functions that the non-malleable extractor is secure against, and where ϵ is the error.[7] Unfortuantely, all known (information theoretic) non-malleable extractors require the seed length to be at least $t \log(1/\epsilon)$

We note that in Section 2.1, we obtained computational non-malleable extractor (in the CRS model) for sources in the high-entropy regime (by using a 2-source extractor from [Bou05, Raz05] as a building block). This extractor, in particular, can be seen as a non-malleable *seeded* extractor. Importantly, it satisfies the requirements of [BACD+17], since in our construction the seed length is independent of t. Thus, one would expect that instantiating the [BACD+17] transformation with our computational non-malleable extractor (in the CRS model), would directly yield a computational 2-source extractor (in the CRS model), with negligible error for low min-entropy sources. However, this turns out not to be the case.

The reason is that the analysis of [BACD+17] crucially requires the underlying non-malleable extractor to be secure against adversaries that run in *unbounded time*. Specifically, even given an

[7] The exact parameters are not relevant to this overview.

efficient adversary that contradicts the security of the 2-source extractor, [BACD+17] obtain an *inefficient* adversary that contradicts the security of the underlying non-malleable extractor. Since our underlying non-malleable extractor is *computational*, it is not clear if this gets us anywhere. Moreover, dealing with sources that can depend on the CRS causes further technical problems. Nevertheless, we show that the construction of [BACD+17] can be instantiated with our computational non-malleable extractor in the CRS model, but with a substantially different (and more technically involved) analysis. In particular, in our analysis we make a novel use the leakage lemma of Gentry and Wichs [GW11].

The blueprint of [BACD+17]. To better understand these technicalities, we begin by describing 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 neighbours of A under Γ is at least K'.

The [BACD+17]-transformation takes a seeded non-malleable extractor

$$\text{nm-Ext} : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^m$$

and a disperser

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

and constructs the following 2-source extractor $\mathsf{2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$, defined by

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

In this work, we instantiate their transformation in the computational setting. In what follows, we first describe the key ideas in the proof from [BACD+17], and then we explain the technical difficulties that arise in the computational setting, and how we resolve them.

Fix any two independent sources X_1 and X_2 with "sufficient" min-entropy. One can argue that

$$(\mathsf{2Ext}(X_1, X_2), X_2) \equiv (U, X_2)$$

as follows:

1. 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

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

   We call a y that satisfies the above property, a good y. By a standard averaging argument one can argue that an overwhelming fraction of y's are good.

2. Fix any x_2 for which there exists an $i \in [t]$ such that $y = \Gamma(x_2, i)$ is good. This means that 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 $\mathsf{2Ext}(X_1, x_2)$ is close to uniform.

3. It thus suffice to show that for $x_2 \leftarrow X_2$, with overwhelming probability there exists an $i \in [t]$ such that $y = \Gamma(x_2, i)$ is good. This can be done by relying on the disperser. Specifically, consider the set of bad x_2's for which $y = \Gamma(x_2, i)$ is not good for all $i \in [t]$. Loosely speaking, if this set occurs with noticeable probability, then one can use the property of the disperser to argue that the support of $\Gamma(x_2, i)$ for $x_2 \in$ bad, $i \in [t]$ covers a large fraction of the y's, and by definition, none of these y's can be good, contradicting the fact that we argued above that an overwhelming fraction of y's must be good.

This completes the outline of the proof in [BACD+17].

The Computational Setting. The intuitive analysis above, while easy to formalize in the information theoretic setting, does not carry over to the computational setting, for various reasons.

1. First, it is not clear that a *computational non-malleable extractor* satisfies the first property used in the [BACD+17] outline. Namely, it is not clear that for an overwhelming fraction of $y \in \{0,1\}^d$, it holds that for all $y'_1, \ldots y'_t$ distinct from y,

$$\begin{aligned}&\left(\mathsf{cnm\text{-}Ext}(X_1, y), \mathsf{cnm\text{-}Ext}(X_1, y'_1), \ldots, \mathsf{cnm\text{-}Ext}(X_1, y'_t)\right) \approx \\ &\left(U, \mathsf{cnm\text{-}Ext}(X_1, y'_1), \ldots, \mathsf{cnm\text{-}Ext}(X_1, y'_t)\right),\end{aligned}$$

   where $\approx$ denotes computational indistinguishability. This is because the computational advantage of an efficient adversary on different y's could cancel out.

2. More importantly, in the computational setting, we would have to construct an *efficient* reduction $\mathcal{R}$ that breaks the non-malleable extractor, given any adversary $\mathcal{A}$ that breaks the 2-source extractor.

   $\mathcal{R}$ obtains input $(\alpha, \widehat{y})$, where $\widehat{y}$ is a random seed and where α is either chosen according to $\mathsf{cnm\text{-}Ext}(X_1, \widehat{y})$ or is chosen uniformly at random. In addition, $\mathcal{R}$ obtains an oracle that outputs $\mathsf{cnm\text{-}Ext}(X_1, y')$ on input $y' \neq \widehat{y}$. The reduction $\mathcal{R}$ is required to *efficiently* distinguish between the case where $\alpha \leftarrow \mathsf{cnm\text{-}Ext}(X_1, \widehat{y})$ and the case where α is chosen uniformly at random.

   In order to use $\mathcal{A}$, $\mathcal{R}$ 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). $\mathcal{R}$ also needs to generate a corresponding x_2 for $\mathcal{A}$, that is sampled according to X_2. How can it generate these values?

   If $\mathcal{R}$ were allowed to be inefficient, then a simple strategy for $\mathcal{R}$ would be the following:

   - Sample $\widehat{x}_2 \leftarrow X_2$ conditioned on the existence of $i \in [t]$ such that $\widehat{y} = \Gamma(\widehat{x}_2, \widehat{i})$.
   - Next, query the oracle on inputs $(y_1, \ldots y_t)$ where for every $i \in [t]$, $y_i = \Gamma(\widehat{x}_2, i)$. As a result, $\mathcal{R}$ obtains $z_i = \mathsf{cnm\text{-}Ext}(x_1, y_i)$ for all $i \in [t] \backslash \widehat{i}$, and sets $z = \left(\bigoplus_{i \in [t]} z_i\right) \oplus \alpha$ (after removing duplicates).
   - It is easy to see that $\widehat{x}_2$ is generated from the distribution X_2. Moreover, if α is the output of cnm-Ext, then z corresponds to $\mathsf{2Ext}(x_1, x_2)$, and otherwise to uniform.
   - At this point, if $\mathcal{A}$ distinguishes z from uniform, $\mathcal{R}$ can echo the output of $\mathcal{A}$ to distinguish α from uniform.

Unfortunately, this does not help us much, because the underlying non-malleable extractor is only guaranteed to be secure against *efficient* adversaries, whereas the adversary $\mathcal{R}$ that we just outlined, crucially needs to invert the disperser. It is not clear that one can build dispersers in our parameter setting that are efficiently invertible. Moreover, even if there was a way to invert the disperser, $\mathcal{R}$ would need to ensure that the inverse $\widehat{x}_2$ is sampled from the correct distribution, and it is unclear whether this can be done efficiently.

3. Our first key idea is to get around this technicality by using the leakage lemma as follows: Since $\mathcal{R}$ on input $\widehat{y}$ cannot find $\widehat{x}_2$ efficiently, we will attempt to view $\widehat{x}_2$ as inefficiently computable *leakage* on $\widehat{y}$, and *simulate* $\widehat{x}_2$ using the following leakage lemma. Informally, this lemma says that any inefficiently computable function that outputs γ bits, can be simulated in time roughly $O(2^\gamma)$ relative to all efficient distinguishers.

   Lemma 2.1. *[GW11, JP14, CLP15] Fix $d, \gamma \in \mathbb{N}$ and fix $\epsilon > 0$. Let $\mathcal{Y}$ be any distribution over $\{0,1\}^d$. Consider any randomized leakage function $\pi : \{0,1\}^d \to \{0,1\}^\gamma$. For every T, there exists a randomized function $\widehat{\pi}$ computable by a circuit of size $\mathrm{poly}\left(2^\gamma \epsilon^{-1} T^{\log T}\right)$ such that for every randomized distinguisher $\mathcal{D}$ that runs in time at most T,*

$$|\Pr[\mathcal{D}(\mathcal{Y}, \pi(\mathcal{Y})) = 1] - \Pr[\mathcal{D}(\mathcal{Y}, \widehat{\pi}(\mathcal{Y})) = 1]| \leqslant \epsilon$$

   By Lemma 2.1, simulating $\widehat{x}_2$ given $\widehat{y}$ would take time roughly $O(2^{|\widehat{x}_2|})$.[8] While this simulator is clearly not as efficient as we would like, one can hope that things still work out if the underlying non-malleable extractor is secure against adversaries running in time $O(2^{|\widehat{x}_2|})$.

   However, any disperser (with our setting of parameter, where t is small) must be compressing, which means that $|\widehat{x}_2| > |\widehat{y}|$. Therefore, the simulator's running time would be more than $O(2^{|\widehat{y}|})$. Howeover, $\widehat{y}$ corresponds to the input of the non-malleable extractor, and recall that our non-malleable extractor applies a (compressing) collision-resistant hash function to its input y. Therefore, the non-malleable extractor is completely *insecure* against adversaries that run in time $O(2^{|\widehat{y}|})$. This creates a circular dependency, and it may appear that this approach is doomed to fail. Nevertheless, we will shortly see how one can apply the leakage lemma in a more sophisticated way. To this end, in what follows, we study our non-malleable extractor in more granular detail.

4. Our next key idea is to split our adversary's running time (for both the 2-source and the non-malleable extractor) into two parts:

   - the time required to sample the sources, and
   - the running time of the distinguisher.

   The reason why this is useful, is that we observe that for the non-malleable extractor, the collision-resistant hash function h only requires the running time of *the distinguisher* be smaller than $O(2^{|\widehat{y}|})$, and allows the sources to be sampled in time significantly higher than $O(2^{|\widehat{y}|})$.

   Armed with this insight, we return to the analysis of our 2-source extractor. Our key idea there, is to also consider two adversaries, one that samples the sources, and the other that distinguishes the output of the 2-source extractor from uniform (as opposed to considering a

[8] Jumping ahead, this is the reason that we end up with a 2-source extractor for unbalanced sources (see Theorem 1.3).

unified reduction $\mathcal{R}$). As a result, we *push* the complexity of simulating an inverse $\widehat{x}_2$ given $\widehat{y}$, to the algorithm that samples the sources for the 2-source extractor. This allows us to complete the leakage argument in the previous bullet, by applying the leakage lemma while sampling the sources, instead of at the time of distinguishing. We therefore get a modular construction of a 2-source extractor, by relying on a non-malleable extractor that is secure for sources sampleable in time $O(2^{|\widehat{x}_2|})$ (as opposed to being secure against adversaries running in time $O(2^{|\widehat{y}|})$).

Roadmap. The rest of this paper is organized as follows. In Section 3, we provide the relevant preliminaries. In Section 4, we provide our new definitions of computational 2-source extractors and non-malleable extractors in the CRS model.

In Section 5 we show how to convert a computational 2-source extractor (in the CRS model) into a computational non-malleable extractor (in the CRS model), with similar min-entropy guarantees. By applying this transformation to the information theoretic 2-source extractors of [Bou05] or [Raz05], we get a computational non-malleable extractor (in the CRS model) for sources in the high min-entropy regime.

Finally, in Section 6 we show how to convert any computatational non-malleable seeded extractor (in the CRS model) into a computational 2-source extractor (in the CRS model) in the low entropy regime. By applying this transformation to the computational non-malleable extractor obtained from Section 5 (which in particular can be thought of as a seeded one), we get a computational 2-source extractor (in the CRS model) in the low entropy regime.

Finally, we obtain our final result which is a computational non-malleable extractor (in the CRS model) in the low entropy regime, by applying the transformation from Section 5 to the computational 2-source extractor that we constructed in Section 6. This is done in Corollary 6.3.

3 Preliminaries

In this section, we discuss some preliminaries needed for the later sections. This includes facts about min-entropy, lossy functions, leakage lemma and dispersers.

Definition 3.1. *A function $\mu : \mathbb{N} \to \mathbb{N}$ is said to be* negligible, *denoted by $\mu = \text{neg}(\lambda)$, if for every polynomial $p : \mathbb{N} \to \mathbb{N}$ there exists a constant $c \in \mathbb{N}$ such that for every $\lambda > c$ it holds that*

$$\mu(\lambda) \leqslant 1/p(\lambda).$$

For any function $T : \mathbb{N} \to \mathbb{N}$, we say that μ is negligible in T, *denoted by $\mu(\lambda) = \text{neg}(T(\lambda))$ if for every polynomial $p : \mathbb{N} \to \mathbb{N}$ there exists a constant $c \in \mathbb{N}$ such that for every $\lambda > c$ it holds that*

$$\mu(\lambda) \leqslant 1/p(T(\lambda)).$$

Definition 3.2. *Two distribution ensembles $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are said to be* $T(\lambda)$-indistinguishable *if for every* $\text{poly}(T)$ *size circuit $\mathcal{A}$,*

$$\left|\Pr_{x\leftarrow X_\lambda}[\mathcal{A}(x) = 1] - \Pr_{y\leftarrow Y_\lambda}[\mathcal{A}(y) = 1]\right| = \text{neg}(T(\lambda))$$

Definition 3.3. *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 min-entropy (and therefore this strengthens our results), and is sometimes more suitable for cryptographic applications.

Definition 3.4. *[DORS08] Let X and Y be two distributions. The average conditional min-entropy of X conditioned on Y, denoted by $H_\infty(X|Y)$[9] 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 3.5. *[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 3.6. *Let X, Y and Z be three distributions, then*

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

Proof of Claim 3.6. Note that

$$\begin{aligned}
&E_{y \leftarrow Y} \max_x \Pr[X = x|Y = y] = \\
&E_{y \leftarrow Y} \max_x \sum_z \Pr[X = x|Y = y, Z = z] \cdot \Pr[Z = z|Y = y] \leqslant \\
&E_{y \leftarrow Y} \sum_z \max_x \Pr[X = x|Y = y, Z = z] \cdot \Pr[Z = z|Y = y] = \\
&\sum_{y,z} \max_x \Pr[X = x|Y = y, Z = z] \cdot \Pr[Z = z|Y = y] \cdot \Pr[Y = y] = \\
&E_{(y,z) \leftarrow (Y,Z)} \max_x \Pr[X = x|Y = y, Z = z]
\end{aligned}$$

Therefore,

$$\begin{aligned}
H_\infty(X|Y) &= -\log E_{y \leftarrow Y} \max_x \Pr[X = x|Y = y] \\
&\geqslant -\log E_{(y,z) \leftarrow (Y,Z)} \max_x \Pr[X = x|Y = y, Z = z] \\
&= H_\infty(X|Y, Z),
\end{aligned}$$

as desired. □

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

3.1 Lossy Functions

Lossy functions were defined by Peikert and Waters in [PW08]. Loosely speaking 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. The injective functions, on the other hand, are injective. It is required that 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 3.7 (Lossy functions). *A function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a (T, n, w)-lossy function family if the following conditions hold:*

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

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

- *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 $s\in\mathrm{Gen}_{\mathrm{inj}}(1^\lambda)$, $f_s\in\mathcal{F}_\lambda$ is injective.*
- *For every $\lambda\in\mathbb{N}$ and every $s\in\mathrm{Gen}_{\mathrm{loss}}(1^\lambda)$, $f_s\in\mathcal{F}_\lambda$ is lossy i.e. its image size is at most $2^{n(\lambda)-w}$.*
- *There is a polynomial time algorithm Eval s.t. $\mathrm{Eval}(s,x) = f_s(x)$ for every $\lambda\in\mathbb{N}$, every s in the support of $\mathrm{Gen}_{\mathrm{inj}}(1^\lambda)\cup\mathrm{Gen}_{\mathrm{loss}}(1^\lambda)$ and every $x\in\{0,1\}^{n(\lambda)}$.*

Modifying the construction in [PW08] (to ensure that the input and output lengths of the functions are the same for every $n = \mathrm{poly}(\lambda)$), [BHK11] gave a construction of a (T, n, w)-lossy function family, for $w = n - n^\epsilon$ (where $\epsilon > 0$ can be any arbitrary small constant), and for every T assuming the DDH assumption holds against $\mathrm{poly}(T)$-size adversaries.

In this work, we use the following lemma.

Lemma 3.8. *[PW08, BHK11] For any constant $\epsilon > 0$ there exists a constant $\delta > 0$ such that for every $\Omega(\lambda) \leqslant n(\lambda) \leqslant \mathrm{poly}(\lambda)$ there exists a (T, n, w)-lossy function family, with $T(\lambda) = 2^{\lambda^\delta}$ and $w = n - n^\epsilon$, assuming the sub-exponential DDH assumption.*

3.2 Leakage Lemma

We make use of the following lemma, which shows that any inefficient leakage function can be simulated efficiently relative to a class of distinguishers.

Lemma 3.9. *[GW11, JP14, CLP15] Fix $d, \gamma\in\mathbb{N}$ and fix $\epsilon > 0$. Let $\mathcal{Y}$ be any distribution over $\{0,1\}^d$. Consider any randomized leakage function $\pi : \{0,1\}^d \to \{0,1\}^\gamma$.*

For every T, there exists a randomized function $\widehat{\pi}$ computable by a circuit of size $\mathrm{poly}\left(2^\gamma\epsilon^{-1}T\right)$ such that for every randomized distinguisher $\mathcal{D}$ that runs in time at most T,

$$|\Pr[\mathcal{D}(\mathcal{Y}, \pi(\mathcal{Y})) = 1] - \Pr[\mathcal{D}(\mathcal{Y}, \widehat{\pi}(\mathcal{Y})) = 1]| \leqslant \epsilon$$

3.3 Dispersers

Definition 3.10. *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 3.11 (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

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

4 Computational Extractors: Definitions

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

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.

At first, it may seem that there is no need to consider leakage explicitely, since one can incorporate the leakage as part of the definition of the min-entropy source; i.e., define the source w.r.t. a fixed leakage value. However, the resulting source may not be efficiently sampleable. For example, if the leakage on a source X is $h(X)$, where h is a collision resistant hash function, then sampling $x \leftarrow X$ conditioned on a given leakage value is computationally hard, due to the collision resistance property of h. Therefore, in the definions below we consider leakage explicitely.

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. We mention that a more general leakage model is one which allows first leakage on Y, then allows leakage on X (that may depend on the initial leakage), and then again allows leakage on Y (that may depend on all the leakage so far), etc. Unfortunately, we do not know how to obtain our results in this more general leakage model.

For technical reasons, we also allow one of the sources (the one which is given to the adversary in the clear, as part of the definition of a strong extractor) to be sampled together with auxiliary information AUX. This auxiliary information depends on the source and on the CRS. As in the leakage case, we need to consider this auxiliary information explicitely, since in our proofs we will use an auxiliary input which is hard to compute given the source and CRS (and therefore cannot generate it while ensuring the security of our underlying hardness assumption). Importantly, it is easy to generate this auxiliary information together with the source, jointly as a function of CRS. As opposed to the case of leakage, the source is not required to have min-entropy conditioned on AUX.

Definition 4.1 (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 $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$, $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, *leakage* $L = \{L_\lambda\}$ *and auxiliary input* $\mathsf{AUX} = \{\mathsf{AUX}_\lambda\}$, *such that for every* $\lambda \in \mathbb{N}$, *the following holds:*

- *For every* $\mathsf{crs} \in \mathsf{Supp}(\mathrm{CRS}_\lambda)$, $\mathsf{Supp}(X_\lambda|\mathsf{crs}) \subseteq \{0,1\}^{n_1(\lambda)}$ *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* $\mathsf{crs} \in \mathsf{Supp}(\mathrm{CRS})$, $(Y, \mathsf{AUX}, L_{\mathsf{init}}|\mathsf{crs})$ *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* $\mathrm{poly}(T)$.
- $H_\infty(X_\lambda|\mathrm{CRS}_\lambda, L_\lambda) \geqslant k_1$ *and* $H_\infty(Y_\lambda|\mathrm{CRS}_\lambda, L_\lambda) \geqslant k_2$.
- *For every* $\mathsf{crs} \in \mathrm{CRS}_\lambda$ *and* $\ell \in \mathsf{Supp}(L_\lambda|\mathsf{crs})$, *the distributions* $(X_\lambda|\mathsf{crs}, \ell)$ *and* $(Y_\lambda, \mathsf{AUX}_\lambda|\mathsf{crs}, \ell)$ *are independent.*
- *For every* $\mathsf{aux} \in \mathsf{Supp}(\mathsf{AUX}_\lambda)$, $|\mathsf{aux}| = O(\log T(\lambda))$[10].

Definition 4.2 (Computational strong 2-source extractors in the CRS model)**.** *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*

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

is said to be a (n_1, n_2, k_1, k_2) strong (T, T')-computational 2-source extractor in the CRS model *if there is an ensemble* $\{\mathrm{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ *where* $\mathrm{CRS}_\lambda \in \{0,1\}^{c(\lambda)}$, *such that the following holds:*

For every T*-admissible leaky* (n_1, n_2, k_1, k_2) *source distribution* (X, Y, L, AUX) *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* $\mathcal{A}$,

$$\left|\Pr\left[\mathcal{A}\left(\mathsf{2Ext}_\lambda(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}\right) = 1\right] - \Pr\left[\mathcal{A}\left(U, y, \mathsf{crs}, \ell, \mathsf{aux}\right) = 1\right]\right| = \nu(T'(\lambda)),$$

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

Definition 4.3 (Computational strong non-malleable extractors in the CRS model)**.** *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, T')-computational non-malleable extractor in the CRS model *if there is an ensemble* $\{\mathrm{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$, *where* $\mathrm{CRS}_\lambda \in \{0,1\}^{c(\lambda)}$, *such that the following holds:*

For every T*-admissible leaky* (n_1, n_2, k_1, k_2) *source distribution* (X, Y, L, AUX) *with respect to* CRS,

[10] We restrict the length of aux to be at most $O(\log T(\lambda))$ for technical reasons.

for every polynomial p, there exists a negligible function $\nu(\cdot)$ such that for every λ and every $p(T'(\lambda))$-size adversary $\mathcal{A}$,

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}\left(\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs}),y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right]-\right.$$
$$\left.\Pr\left[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}\left(U,y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right]\right|=\nu(T'(\lambda)),$$

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 $(\mathsf{crs},x,y,\ell,\mathsf{aux})\leftarrow(\mathsf{CRS}_\lambda,X_\lambda,Y_\lambda,L_\lambda,\mathsf{AUX}_\lambda)$, 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,T')-computational two source (or non-malleable) extractor has error $\mathrm{neg}(T''(\lambda))$ if it satisfies Definition 4.2 (or Definition 4.3), where the adversary's distinguishing advantage is required to be at most $\mathrm{neg}(T''(\lambda))$.

For our constructions, we will 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 4.4. *There exists a (n_1,n_2,k_1,k_2) strong statistical 2-source extractor according to Definition 4.2 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 $\exp^{-\Theta(\min\{k_1,k_2\})}$.*

5 Computational Strong Non-Malleable Extractors in the CRS Model

In this section, we describe the construction of computational non-malleable extractors. Theorem 5.1 describes the reduction from non-malleable extractors to 2-source extractors. Corollary 5.2 instantiates Theorem 5.1 with the known constructions of 2-source extractors with negligible error. We describe the construction in Section 5.1 and analyze it in Section 5.2.

Theorem 5.1. *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, such that $T\geqslant 2^{k_3}$ and the following primitives exist.*

- *A (n_1,n_2,k_1,k_2) strong (T,T')-computational 2-source extractor with error $\mathrm{neg}(2^{k_3})$ in the CRS model, denoted by:*

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

- *A (T,n_1,w)-lossy function family $\mathcal{F}=\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, according to Definition 3.7, where $w=n_1-n_1^\gamma$ for some constant $\gamma\in(0,1)$.*
- *A $\mathrm{poly}(T')$-secure family of collision resistant hash functions $\mathcal{H}=\{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ with $h:\{0,1\}^{n_2}\to\{0,1\}^{k_3}$.*

Then there exists a (n_1,n_2,K_1,K_2) strong (T,T')-computational non-malleable extractor satisfying Definition 4.3 for $K_1=k_1+k_3(n_1-w+1)+1, K_2=k_2+k_3+1$.

Corollary 5.2. *Assume the sub-exponential hardness of DDH, and fix any constant $\epsilon > 0$. Then there exists a constant δ where $0 < \delta < \epsilon$, such that for any parameters (n_1, n_2, K_1, K_2) satisfying*

$$n_1 = \Theta(\lambda), \;\; \Omega(\log^{2/\delta^2} n_1) \leqslant n_2 \leqslant \mathrm{poly}(n_1), \;\; K_1 = n_1^{\epsilon}, \;\; \textit{and } K_2 = 0.51 n_2$$

there exists a (n_1, n_2, K_1, K_2) strong (T, T')-computational non-malleable extractor (satisfying Definition 4.3 and computable in time $\mathrm{poly}(\lambda)$), for $T(\lambda) = 2^{\lambda^\delta}$ and $T'(\lambda) = \lambda^{\log \lambda}$.

Proof of Corollary 5.2. Fix a constant $\epsilon > 0$, and fix $n_1 = n_1(\lambda)$ and $n_2 = n_2(\lambda)$ as in the statement of Corollary 5.2. The sub-exponential hardness of DDH (together with the restrictions on n_1 and n_2) implies that there exists a constant $\delta > 0$ such that the following exist:

- A (T, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ where $T(\lambda) = 2^{\lambda^\delta}$ and w is such that $n_1 - w = n_1^{\epsilon/3}$.

  This follows from Lemma 3.8.

- A collision resistant hash family $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$, where each $h \in \mathcal{H}_\lambda$ is of the form $h : \{0,1\}^{n_2} \to \{0,1\}^{k_3}$, where $k_3 = 0.01 \min\{n_1^{\epsilon/3}, n_2^{\delta/2}\}$, that is secure against $\mathrm{poly}(\lambda^{\log \lambda})$-size adversaries.

  This follows from the following argument: The sub-exponential DDH assumption implies that there exists a constant $\eta > 0$, and there exists such a hash family $\mathcal{H}$ that is collision resistant against adversaries of size $2^{k_3{}^\eta}$. Thus it remains to note that $2^{k_3{}^\eta} \geqslant \lambda^{c \log \lambda}$ for every constant $c \in \mathbb{N}$, which is indeed the case if we set $\delta > 0$ to be small enough. This follows from the definition of k_3, together with the fact that $n_1 \geqslant \Omega(\lambda)$ and $n_2 \geqslant \Omega(\log^{2/\delta^2} n_1)$.

By Theorem 4.4, there exists a (n_1, n_2, k_1, k_2) strong statistical 2-source extractor for $k_1 = n_1^{\epsilon/3}$ and $k_2 = 0.501 n_2$ with error $\exp^{-\Theta(\min(k_1, k_2))} = \mathrm{neg}(2^{k_3})$. In particular, this extractor is a (n_1, n_2, k_1, k_2) strong (T, T')-computational 2-source extractor in the CRS model (where the CRS is empty), with error $\mathrm{neg}(2^{k_3})$.

Therefore as long as $T \geqslant 2^{k_3}$ (which is satisfied for our setting of parameters), then by Theorem 5.1, there exists a (n_1, n_2, K_1, K_2) strong (T, T')-computational non-malleable extractor for $K_1 = k_1 + k_3(n_1 - w + 1) + 1 \leqslant n_1^{\epsilon/3} + 0.01 n_1^{\epsilon/3} \cdot n_1^{\epsilon/3} \leqslant n_1^{\epsilon}$ and $K_2 = k_2 + k_3 + 1 \leqslant 0.501 n_2 + 0.01 n_2 \leqslant 0.51 n_2$, as desired. □

5.1 Construction

In this section, we describe the construction of the non-malleable extractor. We 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 $\mathsf{crs}_{\mathsf{2Ext}}$ for the (n_1, n_2, k_1, k_2) strong (T, T')-computational 2-source extractor with respect to the security parameter 1^λ.
2. Sample $h \leftarrow \mathcal{H}_\lambda$.
3. Sample $b \leftarrow \{0,1\}^{k_3}$.

4. Sample independently k_3 pairs of random injective functions from $\mathcal{F}_\lambda$,

$$f_{1,b_1}, f_{2,b_2}, \ldots, f_{k_3,b_{k_3}} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda).$$

5. 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

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

Our computational non-malleable extractor, $\mathsf{cnm\text{-}Ext} = \{\mathsf{cnm\text{-}Ext}_\lambda\}_{\lambda\in\mathbb{N}}$, is defined as follows: For any $\lambda \in \mathbb{N}$, denote by $c = c(\lambda) = |\mathsf{crs}|$, then

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

where $\forall (x, y, \mathsf{crs}) \in \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^c$, where $\mathsf{crs} = \left(\mathsf{crs}_{\mathsf{2Ext}}, h, \{f_{i,b}\}_{i\in[k_3],b\in\{0,1\}}\right)$

$$\mathsf{cnm\text{-}Ext}_\lambda(x, y, \mathsf{crs}) = \mathsf{2Ext}_\lambda\left(f_{1,h(y)_1} \circ f_{2,h(y)_2} \circ \ldots \circ f_{k_3,h(y)_{k_3}}(x), y, \mathsf{crs}_{\mathsf{2Ext}}\right). \tag{1}$$

5.2 Analysis

In this section, we prove Theorem 5.1; namely, we prove the security of the non-malleable extractor defined in Section 5.1. The proof proceeds in stages. First we replace the oracle $\mathcal{O}^y_{x,\mathsf{crs}}$ with an oracle $\widetilde{\mathcal{O}}^y_{x,\mathsf{crs}}$ which refuses to answer when queried on a y' s.t. the hash values of y and y' match, and we argue that the adversary cannot distinguish between these two oracles. Then in Claim 5.4, we prove that if the adversary succeeds in distinguishing the output of the non-malleable extractor from random, then he can also distinguish even if we condition on the event that $h(y) = b$ (recall that $b \in \{0,1\}^{k_3}$ is used to determine which functions are lossy or injective in the crs). Finally in Claim 5.5, we design a distribution for the 2-source extractor and break it using the supposed adversary for the non-malleable extractor.

Lemma 5.3. *Assuming the conditions in Theorem 5.1 hold,* $\mathsf{cnm\text{-}Ext}$ *is a* (n_1, n_2, K_1, K_2) *strong* (T, T')*-computational non-malleable extractor.*

Proof. We will sometimes suppress the dependence on λ in the notation for convenience.

Fix any T-admissible leaky (n_1, n_2, K_1, K_2) source distribution (X, Y, L, AUX) with respect to CRS. Suppose for the sake of contradiction, that there exists a polynomial p, and a $\mathrm{poly}(T'(\lambda))$-size adversary $\mathcal{A}$, such that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned} &\Pr[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] - \\ &\Pr[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] \geqslant \frac{1}{p(T'(\lambda))}, \end{aligned} \tag{2}$$

where the probabilities are over $(\mathsf{crs}, x, y, \ell, \mathsf{aux}) \leftarrow (\mathrm{CRS}, X, Y, L, \mathsf{AUX})$ and over uniformly distribution $U \leftarrow \{0,1\}^m$.

For any $x \in \{0,1\}^{n_1}$ and $y \in \{0,1\}^{n_2}$, let

$$\begin{aligned} z_{k_3} &\triangleq f_{k_3,1-h(y)_{k_3}}(x) \\ z_{k_3-1} &\triangleq f_{k_3-1,1-h(y)_{k_3-1}}(f_{k_3,h(y)_{k_3}}(x)) \\ &\vdots \\ z_1 &\triangleq f_{1,1-h(y)_1}(f_{2,h(y)_2}(\ldots f_{k_3,h(y)_{k_3}}(x))) \end{aligned}$$

Denote by $z_{x,h(y)} = (z_1, \ldots, z_{k_3})$.

Let $\widetilde{\mathcal{O}}^y_{x,\mathsf{crs}}$ (abusing notation we will call it just $\widetilde{\mathcal{O}}$) be the oracle that on input $y' \in \{0,1\}^{n_2}$, if $h(y') \neq h(y)$ outputs

$$\mathcal{O}^y_{x,\mathsf{crs}}(y') = \mathsf{cnm\text{-}Ext}(x, y', \mathsf{crs}) = \mathsf{2Ext}_\lambda(f_{1,h(y')_1} \circ \ldots \circ f_{k_3,h(y')_{k_3}}(x), y', \mathsf{crs}_{\mathsf{2Ext}}),$$

and otherwise outputs $\perp$. The key observation is that this oracle can be simulated efficiently given only $(h(y), z_{x,h(y)}, \mathsf{crs})$, without any additional information about x or y. This will come in handy later.

Because $\mathcal{H}$ is collision resistant against poly(T')-size adversaries, it follows that $\mathcal{A}$ generates a query $y' \neq y$ such that $h(y') = h(y)$ only with probability neg(T'), and therefore cannot distinguish between oracles $\mathcal{O}^y_{x,\mathsf{crs}}$ and $\widetilde{\mathcal{O}}$ except with probability neg(T'). This, together with Equation (2), implies that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned} &\Pr[\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] - \\ &\Pr[\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] \geqslant \frac{1}{p(T'(\lambda))} - \mathrm{neg}(T'(\lambda)). \end{aligned} \tag{3}$$

where the probabilities are over $(\mathsf{crs}, x, y, \ell, \mathsf{aux}) \leftarrow (\mathsf{CRS}, X, Y, L, \mathsf{AUX})$ and over uniformly distribution $U \leftarrow \{0,1\}^m$. Next, the T-security of the lossy function family, together with the assumption that $T \geqslant 2^{k_3} \geqslant (T')^{\omega(1)}$, 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),

$$\mathrm{neg}(T') \geqslant 2^{-k_3} + \mathrm{neg}(T) \geqslant \Pr[\mathcal{B}(\mathsf{crs}) = b] \geqslant 2^{-k_3} - \mathrm{neg}(T). \tag{4}$$

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

$$\mathrm{neg}(T') \geqslant 2^{-k_3} + \mathrm{neg}(T) \geqslant \Pr\left[h(y) = b\right] \geqslant 2^{-k_3} - \mathrm{neg}(T), \tag{5}$$

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

Claim 5.4. *For infinitely many $\lambda \in \mathbb{N}$,*

$$\begin{aligned} &\Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \\ &- \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \geqslant \frac{1}{4p(T'(\lambda))} \end{aligned} \tag{6}$$

Proof. Towards a contradiction, suppose the claim is not true, that is, for every large enough $\lambda \in \mathbb{N}$,

$$\begin{aligned} &\Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \\ &- \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] < \frac{1}{4p(T'(\lambda))} \end{aligned} \tag{7}$$

We construct a poly(T)-size adversary $\mathcal{B}$ that contradicts the second inequality in Equation (4). On input crs, the adversary $\mathcal{B}$ does the following:

1. Let $N = T' \cdot p^2(T')$.
2. Sample $y \leftarrow (Y|\mathsf{crs})$, and set $z = h(y)$.
3. For every $i \in [N]$, do the following:
   (a) Sample $(x_i, y_i, \ell_i, \mathsf{aux}_i) \leftarrow (X, Y, L, \mathsf{AUX}|\mathsf{crs})$ conditioned on $h(y) = z$[11], and sample $u \leftarrow \{0,1\}^m$ uniformly at random.
   (b) Compute $\delta_i = \mathcal{A}^{\tilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x_i, y_i, \mathsf{crs}), y_i, \mathsf{crs}, \ell_i, \mathsf{aux}_i) - \mathcal{A}^{\tilde{\mathcal{O}}}(u, y_i, \mathsf{crs}, \ell_i, \mathsf{aux}_i)$, by simulating $\tilde{\mathcal{O}}$ for $\mathcal{A}$.

   Set $\delta = \frac{1}{N}\sum_{i=1}^{N} \delta_i$.
4. If $\delta < \frac{1}{2p(T'(\lambda))}$ then output $b' = z$. Else, sample fresh uniformly random $b' \leftarrow \{0,1\}^{k_3}$.

In the rest of the proof, we argue that $\Pr[b' = b] \geqslant 1.5 \cdot 2^{-k_3}$, contradicting the second inequality in Equation (4) (since $T \geqslant 2^{k_3}$).

By Equation (7),

$$\mathbb{E}[\delta | z = b] < \frac{1}{4p(T'(\lambda))}$$

Define $\mathbb{H}$ as the event that $\left(\delta < \frac{1}{2p(T'(\lambda))}\right)$. By a Chernoff bound[12],

$$\Pr[\mathbb{H}|z=b] = 1 - \Pr\left[\left(\delta \geqslant \frac{1}{2p(T'(\lambda)}\right)\middle| z = b\right] \geqslant 1 - \left(e^{-\frac{T' \cdot p^2(T')}{32p^2(T')}}\right) > 1 - \mathsf{neg}(T'(\lambda)) \tag{8}$$

Note that

$$\begin{aligned}
&\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1 \wedge (h(y) \neq b)] - \\
&\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1 \wedge (h(y) \neq b)] \geqslant \\
&\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] - \Pr[h(y) = b] - \\
&\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathsf{aux}) = 1] \geqslant \\
&\frac{1}{p(T'(\lambda))} - \mathsf{neg}(T'(\lambda))
\end{aligned} \tag{9}$$

where the last inequality holds for infinitely many $\lambda \in \mathbb{N}$ by Equations (3) and (4).
This, together with Equation (4), implies that for infinitely many $\lambda \in \mathbb{N}$,

[11] Note that this can be done in time at most $2^{k_3} \leqslant T$.

[12] 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 (8) by setting $\epsilon = \frac{1}{4p(T')}$, $N = T' \cdot p^2(T')$.

$$\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs}),y,\mathsf{crs},\ell,\mathsf{aux})=1|(h(y)\neq b)]-$$
$$\Pr[\mathcal{A}^{\tilde{\mathcal{O}}}(U,y,\mathsf{crs},\ell,\mathsf{aux})=1|(h(y)\neq b)]\geqslant\frac{1}{p(T'(\lambda))}-\mathrm{neg}(T'(\lambda))>\frac{0.99}{p(T'(\lambda))}. \tag{10}$$

where the probabilities are over $\mathsf{crs}\leftarrow\mathrm{CRS}$ (b is used to decide the injective and lossy functions in the crs), and over $(x,y,\ell,\mathsf{aux})\leftarrow(X,Y,L,\mathsf{AUX}|\mathsf{crs})$. This implies that

$$\mathbb{E}[\delta|z\neq b]>\frac{0.99}{p(T'(\lambda))}$$

Recall that $\mathbb{H}$ is the event that $\left(\delta<\frac{1}{2p(T'(\lambda))}\right)$, which implies $\neg\mathbb{H}$ is the event that $\left(\delta\geqslant\frac{1}{2p(T'(\lambda))}\right)$. Therefore, by the Chernoff bound[13],

$$\Pr[\neg\mathbb{H}|z\neq b]=1-\Pr\left[\left(\delta<\frac{1}{2p(T'(\lambda))}\right)\middle|z\neq b\right]\geqslant 1-e^{-\frac{0.49^2}{p^2(T')}\cdot\frac{T'\cdot p(T')^2}{2}}=1-\mathrm{neg}(T'(\lambda)) \tag{11}$$

Furthermore, by construction,

$$\Pr[b'=b|\mathbb{H}\wedge z=b]=1 \tag{12}$$

and

$$\Pr[b'=b|\neg\mathbb{H}\wedge z\neq b]=2^{-k_3} \tag{13}$$

Therefore,

$$\begin{aligned}
&\Pr[b'=b]\\
&\geqslant\Pr[b'=b|\mathbb{H}\wedge z=b]\cdot\Pr[\mathbb{H}|z=b]\cdot\Pr[z=b]\\
&+\Pr[b'=b|\neg\mathbb{H}\wedge z\neq b]\cdot\Pr[\neg\mathbb{H}|z\neq b]\cdot\Pr[z\neq b]\\
&\geqslant 1\cdot\Pr[\mathbb{H}|z=b]\cdot\Pr[z=b]+2^{-k_3}\cdot\Pr[\neg\mathbb{H}|z\neq b]\cdot\Pr[z\neq b]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\big(\text{By substituting with Equations (12) and (13)}\big)\\
&=1\cdot(1-\mathrm{neg}(T'(\lambda)))\cdot\Pr[z=b]+2^{-k_3}\cdot(1-\mathrm{neg}(T'(\lambda)))\cdot\Pr[z\neq b]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\big(\text{By substituting with Equations (8) and (11)}\big)\\
&\geqslant 1\cdot\big(1-\mathrm{neg}(T'(\lambda))\big)(2^{-k_3}-\mathrm{neg}(T(\lambda)))+2^{-k_3}\cdot(1-\mathrm{neg}(T'(\lambda)))\cdot(1-\mathrm{neg}(T'(\lambda)))\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\big(\text{By substituting with Equation (5)}\big)\\
&\geqslant 2^{-k_3}\cdot\big(2-\mathrm{neg}(T'(\lambda))\big)>1.5\cdot 2^{-k_3}.
\end{aligned}$$

This contradicts Equation (4), and thus completes the proof of the claim. □

[13]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,

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

Claim 5.4, together with Equation (5), implies that for infinitely many $\lambda \in \mathbb{N}$:

$$\begin{aligned}
&\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs}),y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\wedge\left(h(y)=b\right)\right]\\
&-\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(U,y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\wedge\left(h(y)=b\right)\right]\\
&=\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs}),y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\middle|\left(h(y)=b\right)\right]\cdot\Pr\left[h(y)=b\right]\\
&-\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(U,y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\middle|\left(h(y)=b\right)\right]\cdot\Pr\left[h(y)=b\right]\\
&\geqslant\frac{1}{4p(T'(\lambda))}\cdot\left(2^{-k_3}-\mathsf{neg}(T(\lambda))\right)\geqslant\frac{1}{p''(2^{k_3})}
\end{aligned}\tag{14}$$

where the last inequality holds for some polynomial $p''(\cdot)$, and it follows from the fact that $T'(\lambda)<2^{k_3}$, which in turn follows from our assumption that the underlying hash function $h:\{0,1\}^{n_2}\to\{0,1\}^{k_3}$ is poly(T')-secure.
Next, substituting

$$\mathsf{cnm\text{-}Ext}(x,y,\mathsf{crs})=\mathsf{2Ext}(f_{1,h(y)_1}\circ f_{2,h(y)_2}\circ\ldots\circ f_{k_3,h(Y)_{k_3}}(x),y,\mathsf{crs}_{\mathsf{2Ext}}),$$

in Equation (14), we conclude that for infinitely many $\lambda\in\mathbb{N}$,

$$\begin{aligned}
&\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(\mathsf{2Ext}(f_{1,h(y)_1}\circ f_{2,h(y)_2}\circ\ldots\circ f_{k_3,h(y)_{k_3}}(x),y,\mathsf{crs}_{\mathsf{2Ext}}),y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\wedge\left(h(y)=b\right)\right]\\
&-\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(U,y,\mathsf{crs},\ell,\mathsf{aux}\right)=1\right)\wedge\left(h(y)=b\right)\right]\geqslant\frac{1}{p''(2^{k_3})}
\end{aligned}\tag{15}$$

We will now use the T-admissible leaky (n_1,n_2,K_1,K_2) source distribution (X,Y,L,AUX) for the non-malleable extractor, to define a new T-admissible leaky (n_1,n_2,k_1,k_2) source distribution (X',Y',L',AUX') for the underlying two-source extractor with CRS distribution $\mathsf{CRS}_{\mathsf{2Ext}}$, where $k_1=K_1-k_3\cdot(n_1-w+1)-1$ and $k_2=K_2-k_3-1$. Then, we will prove that there exists an adversary $\mathcal{A}'$ that breaks the (n_1,n_2,k_1,k_2) strong (T,T')-computational two-source extractor for (X',Y',L',AUX').

Define $(X',Y',L',\mathsf{AUX}'|\mathsf{crs}_{\mathsf{2Ext}})$ as follows:

1. We first define $(Y',L'_{\mathsf{init}},\mathsf{AUX}')|\mathsf{crs}_{\mathsf{2Ext}})$:
   (a) Sample $b\leftarrow\{0,1\}^{k_3}$.
   (b) Sample $f_h=\begin{pmatrix}h, & f_{1,0},f_{2,0},\ldots,f_{k_3,0}\\ & f_{1,1},f_{2,1},\ldots,f_{k_3,1}\end{pmatrix}$ such that $\{f_{i,b_i}\}_{i\in[k_3]}$ are injective and the rest are lossy. Set $\mathsf{crs}=(\mathsf{crs}_{\mathsf{2Ext}},f_h)$.
   (c) Sample $(y,\ell_{\mathsf{init}},\mathsf{aux})\leftarrow(Y,L_{\mathsf{init}},\mathsf{AUX}|\mathsf{crs})$.
   (d) Set $(y',\mathsf{aux}')=(y,\mathsf{aux})$.
   (e) Set $\ell'_{\mathsf{init}}=(d,\ell_{\mathsf{init}},f_h,b)$, where $d=0$ if $h(y)\neq b$ and 1 otherwise.
2. We next define $(X',L'_{\mathsf{final}}|\mathsf{crs}_{\mathsf{2Ext}},\ell'_{\mathsf{init}})$:
   (a) Parse $\ell'_{\mathsf{init}}=(d,\ell_{\mathsf{init}},f_h,b)$, and set $\mathsf{crs}=(\mathsf{crs}_{\mathsf{2Ext}},f_h)$.

(b) Sample $(x, \ell_{\mathsf{final}}) \leftarrow (X, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$. Set $x' = f_{1,b_1} \circ f_{2,b_2} \circ \ldots \circ f_{k_3,b_{k_3}}(x)$ and $\ell'_{\mathsf{final}} = (\ell_{\mathsf{final}}, z_{x,b})$, where

$$z_{x,b} = \{z_1, \ldots, z_{k_3}\} \text{ and for every } i \in [\ell], z_i := f_{i,1-b_i}(f_{i+1,b_{i+1}}(\ldots f_{k_3,b_{k_3}}(x))).$$

Claim 5.5. *$(X', Y', L', \mathsf{AUX}')$ is a T-admissible leaky (n_1, n_2, k_1, k_2) source distribution with respect to $\mathsf{CRS}_{\mathsf{2Ext}}$, where $k_1 = K_1 - k_3 \cdot (n_1 - w + 1) - 1$ and $k_2 = K_2 - k_3 - 1$.*

Proof. First, we observe that $(Y', \mathsf{AUX}', L'_{\mathsf{init}}|\mathsf{crs}_{\mathsf{2Ext}})$ and $(X', L'_{\mathsf{final}}|\mathsf{crs}_{\mathsf{2Ext}}, \ell'_{\mathsf{init}})$ can be sampled in time $\mathrm{poly}(T)$.

Next we note that

$$\begin{aligned}
H_\infty(Y'|\mathsf{CRS}_{\mathsf{2Ext}}, L'_{\mathsf{init}}) &= H_\infty(Y|\mathsf{CRS}_{\mathsf{2Ext}}, D, L_{\mathsf{init}}, f_h, B) \\
&\geqslant H_\infty(Y|\mathsf{CRS}_{\mathsf{2Ext}}, L_{\mathsf{init}}, f_h) - k_3 - 1 \quad \text{(by Claim 3.5)} \\
&= H_\infty(Y|\mathsf{CRS}, L_{\mathsf{init}}) - k_3 - 1 \\
&\geqslant K_2 - k_3 - 1 \quad \text{(by assumption).}
\end{aligned}$$

Similarly,

$$\begin{aligned}
&H_\infty(X'|\mathsf{CRS}_{\mathsf{2Ext}}, L') \\
&= H_\infty(X'|\mathsf{CRS}_{\mathsf{2Ext}}, L_{\mathsf{final}}, Z_{X,B}, D, L_{\mathsf{init}}, f_h, B) \\
&= H_\infty(X|\mathsf{CRS}_{\mathsf{2Ext}}, L_{\mathsf{final}}, Z_{X,B}, D, L_{\mathsf{init}}, f_h, B) \quad \text{(since } f_{i,b_i}\text{'s are injective)} \\
&\geqslant H_\infty(X'|\mathsf{CRS}_{\mathsf{2Ext}}, L_{\mathsf{final}}, L_{\mathsf{init}}, f_h) - k_3 \cdot (n_1 - \omega + 1) - 1 \\
&\qquad \text{(by Claim 3.5 and since } f_{i,1-b_i}\text{'s are lossy)} \\
&= H_\infty(X'|\mathsf{CRS}, L) - k_3 \cdot (n_1 - \omega + 1) - 1 \\
&\geqslant K_1 - k_3 \cdot (n_1 - \omega + 1) - 1 \qquad \text{(by assumption).}
\end{aligned}$$

It remains to argue that for every $\mathsf{crs}_{\mathsf{2Ext}} \in \mathsf{Supp}(\mathsf{CRS}_{\mathsf{2Ext}})$ and every $\ell' \in \mathsf{Supp}(L'(\mathsf{crs}_{\mathsf{2Ext}}))$ it holds that X' is independent of (Y', AUX'). To this end, fix any such $\mathsf{crs}_{\mathsf{2Ext}}$ and ℓ'.

First, we 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 $(Y', \mathsf{AUX}') = (Y, \mathsf{AUX})$ where (Y, AUX) is sampled conditioned on $(\mathsf{crs}, \ell_{\mathsf{init}})$ and on $h(Y) = b$.

The fact that (X, Y, L, AUX) is T-admissible w.r.t. CRS, implies that X and (Y, AUX) 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, AUX) 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, AUX) 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', AUX') are independent conditioned on $(\mathsf{crs}_{\mathsf{2Ext}}, \ell')$, as desired.

This completes the proof of the claim. □

Now, note that Equation (15), together with the definition of $(X', Y', L', \mathsf{AUX}'|\mathsf{crs2Ext})$, implies that there exists a $\mathrm{poly}(T')$-size adversary $\mathcal{A}'$, that simulates the adversary $\mathcal{A}$, as well as its oracle, such that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned}
&\Pr[\mathcal{A}'(\mathsf{2Ext}(X', Y', \mathsf{crs}_{\mathsf{2Ext}}), y', \mathsf{crs}_{\mathsf{2Ext}}, \ell', \mathsf{aux}') = 1] - \Pr[\mathcal{A}'(U, y', \mathsf{crs}_{\mathsf{2Ext}}, \ell', \mathsf{aux}') = 1] \\
&\geqslant 1/\mathrm{poly}(2^{k_3}). \qquad (16)
\end{aligned}$$

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

1. Parse $\ell' = (\ell'_{\mathsf{init}}, \ell'_{\mathsf{final}})$ and further parse $\ell'_{\mathsf{init}} = (d, \ell_{\mathsf{init}}, f_h, h(y))$, $\ell'_{\mathsf{final}} = (\ell_{\mathsf{final}}, z_{x,h(y)})$. and obtain d from ℓ'_{init}.
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 $\mathcal{A}^{\widetilde{\mathcal{O}}}(\alpha, y', \mathsf{crs}, \ell, \mathsf{aux}')$, where the oracle $\widetilde{\mathcal{O}}$ is simulated using $(h(y), z_{x,h(y)}, \mathsf{crs})$.

Equation (15) implies that indeed Equation (16) holds, as desired. This contradicts the fact that 2Ext is a strong (T, T')-computational 2-source extractor for $(X', Y', L', \mathsf{AUX}')$ with error $\mathsf{neg}(2^{k_3})$. □

6 Computational Strong 2-Source Extractors in the CRS Model

In this section, we describe our compiler that converts any computational non-malleable extractor (in the CRS model) with negligible error for sources in the high entropy regime, into a computational 2-source extractor (in the CRS model) with negligible error for sources in the low entropy regime. This construction is essentially identical to that suggested by [BACD+17]. However, the analysis in the computational setting introduces many technical challenges which result from the existence of the CRS, and the necessity of building an efficient reduction. In this section, we formally describe the construction and the techniques that we develop to overcome these challenges. We prove the following theorem.

Here is a brief plan for this section. Theorem 6.1 talks about the reduction from 2-source extractors to non-malleable extractors. Corollary 6.2 instantiates Theorem 6.1 with the non-malleable extractor constructed in Section 5. Corollary 6.3 instantiates Theorem 5.1 with the 2-source extractor from Corollary 6.2 to get a stronger non-malleable extractor. Section 6.1 describes the reduction from 2-source extractors to non-malleable extractors and Section 6.2 analyzes this reduction.

Theorem 6.1. *Let $T, T', n_1, n_2, k_1, k_2, d : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter, such that $T = (T')^{\omega(1)}$ and $T = \lambda^{\Omega(1)}$, $n_2 = O(\log T)$, $k_2 = \omega(\log T')$, and such that the following primitives exist.*

- *A (n_1, d, k_1, d) strong (T, T')-computational non-malleable extractor in the CRS model (according to Definition 4.3), denoted by*

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

- *A $\left(\frac{2^{k_2}}{T' \log T'}, 2^{d-1}\right)$ disperser*

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

with degree $t = \mathsf{poly}(\lambda)$ (according to Definition 3.10).

Then there exists a $(n_1, n_2, k_1, 2k_2)$ strong (T, T')-computational 2-source extractor in the CRS model (according to Definition 4.2).

Theorem 6.1 implies the following corollary.

Corollary 6.2. *Fix any constant $\epsilon > 0$. Then assuming the sub-exponential hardness of the DDH assumption, there exists a constant $\delta > 0$ for which there exists a (n_1, n_2, k_1, k_2) strong (T, T')-computational 2-source extractor (computable in time $\text{poly}(\lambda)$) in the CRS model (satisfying Definition 4.2) for any $n_1, n_2, k_1, k_2, T, T'$ satisfying:*

$$\Omega(\lambda) \leqslant n_1 \leqslant \text{poly}(\lambda), \Omega(\log^{2/\delta^2} n_1) \leqslant n_2 \leqslant O(\lambda^\delta), k_1 = n_1^\epsilon, k_2 = \Omega(\log^{2/\delta^2} n_1), T = 2^{\lambda^\delta} \text{ and } T' = \lambda^{\log \lambda}.$$

Proof. Fix any constant $\epsilon > 0$. By Corollary 5.2, there exists a constant δ where $0 < \delta < \epsilon$, for which there exists a (n_1, d, K_1, d) strong (T, T')-computational non-malleable extractor in the CRS model, for $T = 2^{\lambda^\delta}$ and $T' = \lambda^{\log \lambda}$, and for any n_1, d, K_1 such that

$$\Omega(\lambda) \leqslant n_1 \leqslant \text{poly}(\lambda), \Omega(\log^{2/\delta^2} n_1) \leqslant d \leqslant n_1, K_1 = n_1^\epsilon$$

Fix $T = 2^{\lambda^\delta}$ and $T' = \lambda^{\log \lambda}$, and fix any n_1 such that $\Omega(\lambda) \leqslant n_1 \leqslant \text{poly}(\lambda)$. Let $k_1 = n^\epsilon$.

By Theorem 3.11, there exists a polynomial $t = \text{poly}(\lambda)$ for which there exists a $\left(\frac{2^{k_2}}{T'(\log T')}, 2^{d-1}\right)$ disperser

$$\Gamma : \{0,1\}^{n_1} \times [t] \to \{0,1\}^d$$

for any d, k_2 that satisfy

$$k_2 \geqslant 2d + \log^2 T' = 2d + \log^4 \lambda.$$

Setting $d = \Omega(\log^{2/\delta^2} n_1)$, we have that

$$k_2 \geqslant \Omega(\log^{2/\delta^2} n_1) + \log^4 \lambda = \Omega(\log^{2/\delta^2} n_1),$$

where the latter equation follows from the fact that $n_1 = \Omega(\lambda)$. Therefore, Theorem 6.1 implies that there exists a $(n_1, n_2, k_1, 2k_2)$ strong (T, T')-computational 2-source extractor in the CRS model, as long as $n_2 = O(\log T) = O(\lambda^\delta)$, and as long as $k_2 = \omega(\log T') = \omega(\log^2 \lambda)$, and in particular for $k_2 = \Omega(\log^{2/\delta^2} n_1)$, as desired. □

By using the 2-source extractor obtained as a result of Corollary 6.2 to instantiate the non-malleable extractor in Theorem 5.1, we obtain the following corollary:

Corollary 6.3. *Fix any constant $\epsilon > 0$. Then, assuming the sub-exponential hardness of the DDH assumption, there exists a constant $\delta > 0$ for which there exists a (n_1, n_2, K_1, K_2) strong (T, T')-computational non-malleable extractor (satisfying Definition 4.3) whenever*

$$\Omega(\lambda) \leqslant n_1 \leqslant \text{poly}(\lambda), \Omega(\log^{2/\delta^2} n_1) \leqslant n_2 \leqslant O(\lambda^\delta), K_1 = n_1^\epsilon, k_2 = \Omega(\log^{2/\delta^2} n_1), T = \lambda^{\log \lambda} \text{ and } T' = \lambda.$$

Proof. Fix any constant $\epsilon > 0$. By Corollary 6.2, assuming the sub-exponential hardness of DDH, there exists a constant $\delta > 0$ such that there exists a (n_1, n_2, k_1, k_2) strong (T_1, T_2)-computational two source extractor for $n_1, n_2, k_1, k_2, T_1, T_2$ satisfying

$$\Omega(\lambda) \leqslant n_1 \leqslant \text{poly}(\lambda), \Omega(\log^{2/\delta^2} n_1) \leqslant n_2 \leqslant O(\lambda^\delta), k_1 = n_1^{\epsilon/3}, k_2 = \Omega(\log^{2/\delta^2} n_1), T_1 = 2^{\lambda^\delta} \text{ and } T_2 = \lambda^{\log \lambda},$$

and with error $\text{neg}(T_2)$.

Furthermore, the sub-exponential hardness of DDH, together with the fact that $n_1 \geqslant \Omega(\lambda)$, implies that the following exist:

- A (T, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ where each $f \in \mathcal{F}_\lambda$ is of the form $f : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$, and where $T(\lambda) = \lambda^{\log\lambda}$ and w is such that $n_1 - w = n_1^{\epsilon/3}$.
- A collision resistant hash family $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$, where each $h \in \mathcal{H}_\lambda$ is of the form $h : \{0,1\}^{n_2} \to \{0,1\}^{k_3}$ where $k_3 = (\log\lambda)\cdot(\log\log\lambda)$, that is secure against poly(λ)-size adversaries.

Since $T > 2^{k_3}$, by Theorem 5.1, there exists a (n_1, n_2, K_1, K_2) strong (T, T')-computational non-malleable extractor for $K_1 \geqslant k_1 + k_3(n_1 - w) + 1 = n_1^{\epsilon/3} + n_1^{\epsilon/3}\cdot n_1^{\epsilon/3}$, that is, when $K_1 \geqslant n_1^{\epsilon}$, and $K_2 \geqslant k_2 + k_3 + 1 = \Omega(\log^{2/\delta^2} n_1) + (\log\lambda)\cdot(\log\log\lambda)$, that is $K_2 = \Omega(\log^{2/\delta^2} n_1)$, as desired. □

6.1 Construction.

In this section, we describe the reduction from 2-source extractors to non-malleable extractors.

Fix any parameters $T, T', n_1, n_2, k_1, k_2, d$ according to Theorem 6.1. Fix any (n_1, d, k_1, d) strong (T, T')-computational non-malleable extractor

$$\mathsf{cnm\text{-}Ext} : \{0,1\}^{n_1} \times \{0,1\}^{d} \times \{0,1\}^{c} \to \{0,1\}^{m}$$

and any $\left(\frac{2^{k_2}}{T'\log T'}, 2^{d-1}\right)$ disperser

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

Define

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

by

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

6.2 Analysis

In this section, we prove the security of the reduction described above. The proof proceeds in several steps. We start by assuming an adversary that breaks the 2-source extractor (and also the distribution on which it breaks the extractor). Using this adversary, we define an adversary that is supposed to break the non-malleable extractor (on a distribution to be defined later). Then we define the sets BAD-rand and BAD-seed. These capture the places where the adversary breaks the non-malleable extractor. Claims 6.4 and 6.5 prove that these sets are large. Finally in Claim 6.6 we define the distribution on which the adversary breaks the non-malleable extractor. This relies on the leakage lemma.

Suppose for contradiction that 2Ext is not an $(n_1, n_2, k_1, 2k_2)$ strong (T, T')-computational 2-source extractor. This implies that there exists a T-admissible leaky $(n_1, n_2, k_1, 2k_2)$ source distribution $(X_1, X_2, L, \mathsf{AUX})$, a poly$(T')$-size adversary $\mathcal{B}$, and a polynomial $p'(\cdot)$, such that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{B}(\mathsf{2Ext}(x_1, x_2, \mathsf{crs}), x_2, \mathsf{crs}, \ell, \mathsf{aux}) = 1] - \Pr[\mathcal{B}(U, x_2, \mathsf{crs}, \ell, \mathsf{aux}) = 1] \geqslant \frac{1}{p'(T')} \tag{17}$$

where the probability is over $(\mathsf{crs}, x_1, x_2, \ell, \mathsf{aux}) \leftarrow (\mathsf{CRS}, X_1, X_2, L, \mathsf{AUX})$.
We use $\mathcal{B}$ to build a poly(T')-size adversary $\mathcal{A}$ for the non-malleable extractor. We will define a

T-admissible source distribution $(X', Y', L', \mathsf{AUX}')$ such that $\mathcal{A}$ breaks the non-malleable extractor with respect to this source distribution.

The source distribution $(X', Y', L', \mathsf{AUX}')$ depends on the adversary $\mathcal{A}$, and therefore, we defer the description of this distribution until after we define $\mathcal{A}$. However, we point out that the auxiliary information aux' is either $\perp$, or is of the form (aux, x_2, i) where aux corresponds to the auxiliary information of the underlying distribution $(X_1, X_2, L, \mathsf{AUX})$, and (x_2, i) is such that $\Gamma(x_2, i) = y'$. We start by defining our adversary $\mathcal{A}$.

The adversary $\mathcal{A}$. $\mathcal{A}$ on input $(\alpha, y', \mathsf{crs}, \ell', \mathsf{aux}')$ and with access to oracle $\mathcal{O}$, does the following:

1. If $\mathsf{aux}' = \perp$, output 0. Otherwise, continue.
2. Parse $\mathsf{aux}' = (\mathsf{aux}, x_2, i)$.
3. For every $j \in [t] \backslash \{i\}$, let $v_j = \Gamma(x_2, j)$.
4. Let $v_i = y'$.
5. Set $z = \alpha \oplus \bigoplus_{v : v \neq y' \text{ and there exists } j \in [t]\backslash\{i\} \text{ s.t. } v_j = v} \mathcal{O}(v)$.
6. Output $\mathcal{B}\big(z, x_2, \mathsf{crs}, \ell', \mathsf{aux}\big)$.

Note that $\mathcal{A}$ is a poly(T')-size adversary.

Next, for any $\mathsf{crs} \in \mathsf{Supp}(\mathrm{CRS})$, and any $\ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$, we define the sets $\mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$ and $\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})$. Before formally defining these sets, we provide some intuition to assist the reader.

- The set $\mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$ consists of randomness r such that when sampling $(X_2, \mathsf{AUX}, L_{\mathsf{init}})|\mathsf{crs}$ with randomness r, one obtains $(x_2, \mathsf{aux}, \ell_{\mathsf{init}})$, such that ℓ_{init} is the a priori fixed leakage, and such that for every $i \in [t]$, the adversary $\mathcal{A}$ distinguishes between

  $$\big(\mathsf{cnm\text{-}Ext}(x_1, y'_i, \mathsf{crs}), y'_i, \mathsf{crs}, \ell, \mathsf{aux}'\big) \text{ and } \big(U, y'_i, \mathsf{crs}, \ell, \mathsf{aux}'\big)$$

  with probability at least $\frac{1}{2p'(T')}$, where $y'_i = \Gamma(x_2, i)$, $\mathsf{aux}' = (\mathsf{aux}, x_2, i)$, and where the probability is over $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$ (and $\ell = (\ell_{\mathsf{init}}, \ell_{\mathsf{final}})$).
- The set $\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})$ consists of all y' for which there exists an $r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$ and $i \in [t]$, such that for x_2 sampled conditioned on the crs using randomness r,

  $$y' = \Gamma(x_2, i)$$

Looking ahead, we will prove that with noticeable probability over $\mathsf{crs} \leftarrow \mathrm{CRS}$ and $\ell_{\mathsf{init}} \leftarrow L_{\mathsf{init}}|\mathsf{crs}$, the sets $\mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$ and $\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})$ are "large". We will now formally define these sets.

Defining BAD-rand**.** For any $\mathsf{crs} \in \mathsf{Supp}(\mathrm{CRS})$ and any $\ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$, the set $\mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$ is defined as follows.

$$\mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}}) = \Bigg\{ r : (x_2, \mathsf{aux}, \ell_{\mathsf{init}}) = (X_2, \mathsf{AUX}, L_{\mathsf{init}})(\mathsf{crs}; r) \text{ and}$$
$$\forall i \in [t], \text{ for } y := \Gamma(x_2, i), \Pr\Big[\mathcal{A}^{\mathcal{O}^y_{x_1,\mathsf{crs}}}\big(\mathsf{cnm\text{-}Ext}(x_1, y, \mathsf{crs}), y, \mathsf{crs}, \ell, (\mathsf{aux}, x_2, i)\big) = 1\Big] -$$
$$\Pr\Big[\mathcal{A}^{\mathcal{O}^y_{x_1,\mathsf{crs}}}\big(\mathcal{U}, y, \mathsf{crs}, \ell, (\mathsf{aux}, x_2, i)\big) = 1\Big] \geqslant \frac{1}{2p'(T')} \Bigg\}$$

where the probabilities are over $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}})|(\mathsf{crs}, \ell_{\mathsf{init}})$ and where $\ell = (\ell_{\mathsf{init}}, \ell_{\mathsf{final}})$.

We now prove the following claim about BAD-rand.

Claim 6.4. *With probability at least* $\frac{1}{4p'(T')}$ *over* $\mathsf{crs} \leftarrow \mathrm{CRS}$ *and over the randomness of sampling* $\ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs})$,

$$\Pr_{r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}}\Big[r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})\Big] \geqslant \frac{1}{4p'(T')} \tag{18}$$

Proof. Suppose the claim is not true, then with probability at least $1 - \frac{1}{4p'(T')}$ over $\mathsf{crs} \leftarrow \mathrm{CRS}$ and over the randomness of sampling $\ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs})$,

$$\Pr_{r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}}[r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})] < \frac{1}{4p'(T')}$$

This implies that

$$\Pr_{\mathsf{crs} \leftarrow \mathrm{CRS}, \ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs}), r \leftarrow \{0,1\}^{\mathrm{poly}(\lambda)}}[r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})] < \frac{1}{2p'(T')} \tag{19}$$

By definition, for every $(\mathsf{crs}, \ell_{\mathsf{init}}, r)$ such that $r \notin \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$, and $(x_2, \mathsf{aux}, \ell_{\mathsf{init}}) = (X_2, \mathsf{AUX}, L_{\mathsf{init}})(\mathsf{crs}; r)$, $\exists i \in [t]$ such that for $y = \Gamma(x_2, i)$,

$$\Pr\Big[\mathcal{A}^{\mathcal{O}^y_{x_1,\mathsf{crs}}}\big(\mathsf{cnm\text{-}Ext}(x_1, y, \mathsf{crs}), y, \mathsf{crs}, \ell, (\mathsf{aux}, x_2, i)\big)\Big] -$$
$$\Pr\Big[\mathcal{A}^{\mathcal{O}^y_{x_1,\mathsf{crs}}}\big(U, y, \mathsf{crs}, \ell, (\mathsf{aux}, x_2, i)\big)\Big] < \frac{1}{2p'(T')}$$

where the probability is over the randomness of sampling $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}})|(\mathsf{crs}, \ell_{\mathsf{init}})$.

This implies that for every $(\mathsf{crs}, \ell_{\mathsf{init}}, r)$ such that $r \notin \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})$, and $(x_2, \mathsf{aux}, \ell_{\mathsf{init}}) = (X_2, \mathsf{AUX}, L_{\mathsf{init}})(\mathsf{crs}; r)$,

$$\Pr\Big[\mathcal{B}\big(\mathsf{2Ext}(x_1, x_2, \mathsf{crs}), x_2, \mathsf{crs}, \ell, \mathsf{aux}\big)\Big] -$$
$$\Pr\Big[\mathcal{B}\big(U, x_2, \mathsf{crs}, \ell, \mathsf{aux}\big)\Big] < \frac{1}{2p'(T')} \tag{20}$$

where the probability is over the randomness of sampling $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$.

This, together with Equation (19) implies that

$$\Pr\Big[\mathcal{B}\big(\mathsf{2Ext}(x_1, x_2, \mathsf{crs}), x_2, \mathsf{crs}, \ell, \mathsf{aux}\big)\Big] - \Pr\Big[\mathcal{B}\big(U, x_2, \mathsf{crs}, \ell, \mathsf{aux}\big)\Big] < \frac{1}{p'(T')}$$

where the probability is over $\mathsf{crs} \leftarrow \mathsf{CRS}, (x_1, x_2, \ell, \mathsf{aux}) \leftarrow (X_1, X_2, L, \mathsf{AUX}|\mathsf{crs})$. This contradicts Equation (17) and thus completes the proof of this claim. □

Defining BAD-seed**.** For any $\mathsf{crs} \in \mathsf{CRS}, \ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$, define the set $\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})$ as follows:

$$\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}}) = \Big\{y : \exists r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}}), \exists i \in [t], \text{ such that } y := \Gamma(x_2, i) \text{ for } x_2 := X_2(\mathsf{crs}; r)\Big\}$$

where $x_2 := X_2(\mathsf{crs}; r)$ is a shorthand for the notation $(x_2, \cdot, \cdot) := (X_2, \mathsf{AUX}, L_{\mathsf{init}})(\mathsf{crs}; r)$.

Claim 6.5.

$$\Pr[y \in \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})] \geqslant \frac{1}{10p'(T')}$$

where the probability is over the randomness of sampling $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell_{\mathsf{init}} \leftarrow L_{\mathsf{init}}|\mathsf{crs}$, *and* $y \leftarrow \{0,1\}^d$.

Proof. First, recall that by definition of BAD-seed, for every $\mathsf{crs} \in \mathsf{Supp}(\mathsf{CRS})$ and every $\ell_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$,

$$\Big|\Big\{\Gamma\big(X_2(r, \mathsf{crs}, \ell_{\mathsf{init}}), i\big)\Big\}_{r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}}), i \in [t]}\Big| \subseteq \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}}).$$

Next, recall that $H_\infty(X_2|\mathsf{CRS}, L_{\mathsf{init}}) = 2k_2$, which implies that with probability $1 - 2^{-k_2}$ over choice of $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell \leftarrow L_{\mathsf{init}}$, $H_\infty(X_2|\mathsf{crs}, \ell_{\mathsf{init}}) = (2k_2 - k_2) = k_2$. This, together with Claim 6.4, implies that with probability at least $\frac{1}{5p'(T')}$, over the randomness of sampling $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs})$,

$$\Big|\Big\{X_2(r, \mathsf{crs}, \ell_{\mathsf{init}})\Big\}_{r \in \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell_{\mathsf{init}})}\Big| \geqslant \frac{2^{k_2}}{4p'(T')} > \frac{2^{k_2}}{T' \log T'}$$

By the definition of $(\frac{2^{k_2}}{T' \log T'}, 2^{d-1})$ disperser (Definition 3.10), Γ maps each set of size at least $\frac{2^{k_2}}{T'(\log T')}$ to a set of size at least 2^{d-1}. Therefore, with probability at least $\frac{1}{5p'(T')}$ over sampling $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs})$,

$$|\mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})| \geqslant 2^{d-1},$$

Thus, with probability at least $\frac{1}{5p'(T')}$ over sampling $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs})$,

$$\Pr[y \in \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})] \geqslant \frac{1}{2}$$

where the probability is over the randomness of sampling $y \leftarrow \{0,1\}^d$.

Thus,

$$\Pr[y \in \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell_{\mathsf{init}})] \geqslant \frac{1}{2} \cdot \frac{1}{5p'(T')} > \frac{1}{10p'(T')}$$

where the probability is over the randomness of sampling $\mathsf{crs} \leftarrow \mathsf{CRS}, \ell_{\mathsf{init}} \leftarrow (L_{\mathsf{init}}|\mathsf{crs}), y \leftarrow \{0,1\}^d$, as desired. □

Finally, we define $(X', Y', L', \mathsf{AUX}')$ such that $\mathcal{A}$ breaks the non-malleable extractor on these sources. We note that the CRS for the 2-source extractor is identical to the CRS for the non-malleable extractor.

Defining the sources $(X', Y', L', \mathsf{AUX}'|\mathsf{crs})$ for the non-malleable extractor.

- **Sampling $(L'_{\mathsf{init}}|\mathsf{crs})$.** Sample $(x_2, \mathsf{aux}, \ell_{\mathsf{init}}) \leftarrow (X_2, \mathsf{AUX}, L_{\mathsf{init}}|\mathsf{crs})$, and set $\ell'_{\mathsf{init}} = \ell_{\mathsf{init}}$.
- **Sampling $(Y', \mathsf{AUX}'|\mathsf{crs}, \ell'_{\mathsf{init}})$.**
  1. Sample $y' \leftarrow \{0,1\}^d$.
  2. Compute aux' as a function of $(\mathsf{crs}, \ell'_{\mathsf{init}}, y')$ as follows:
     (a) Define the inefficient function F that on input $(\mathsf{crs}, \ell'_{\mathsf{init}}, y')$ outputs $\overline{\mathsf{aux}}$ computed as follows.
     - If $y' \notin \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell'_{\mathsf{init}})$, set $\overline{\mathsf{aux}} = \bot$.
     - Else output $\overline{\mathsf{aux}} = (\mathsf{aux}, x_2, i)$, where (aux, x_2, i) is computed as follows.

       Sample $r \leftarrow \mathsf{BAD\text{-}rand}(\mathsf{crs}, \ell'_{\mathsf{init}})$ such that $\exists i : \Gamma(X_2(r, \mathsf{crs}, \ell'_{\mathsf{init}}), i) = y'$,

       and set $(x_2, \mathsf{aux}) = (X_2, \mathsf{AUX}|\mathsf{crs}; r)$.

     (b) Note that the output of F consists of $O(\log T)$ bits. This follows from the assumption that $|\mathsf{AUX}| = O(\log T)$, $x_2 = n_2 = O(\log T)$, and $|i| = \log t = O(\log \lambda) = O(\log T)$. Apply Lemma 2.1 with respect to the function F, the input distribution $(\mathrm{CRS}, L'_{\mathsf{init}}, Y')$, and the time bound T', to conclude that there exists a function $\widehat{F}$, computable in time $\mathrm{poly}(T)$, such that no adversary running in time $\mathrm{poly}(T')$ can distinguish between the distributions

$$\Big((X_1, L_{\mathsf{final}}), (\mathrm{CRS}, L'_{\mathsf{init}}, Y'), F(\mathrm{CRS}, L'_{\mathsf{init}}, Y')\Big)$$

and

$$\Big((X_1, L_{\mathsf{final}}), (\mathrm{CRS}, L'_{\mathsf{init}}, Y'), \widehat{F}(\mathrm{CRS}, L'_{\mathsf{init}}, Y')\Big)$$

     with advantage better than $\mathrm{neg}(T')$. The fact that $\widehat{F}$ is computable in time $\mathrm{poly}(T)$ follows from the fact that $T = (T')^{\omega(1)}$ and that the output of F is $O(\log T)$ bits.

     (c) Set $\mathsf{aux}' \leftarrow \widehat{F}(\mathsf{crs}, \ell'_{\mathsf{init}}, y')$.
- **Sampling $(X', L'_{\mathsf{final}}|\mathsf{crs}, \ell'_{\mathsf{init}})$.**
  1. Sample $(x_1, \ell_{\mathsf{final}}) \leftarrow (X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell'_{\mathsf{init}})$.

2. Set $(x' = x_1, \ell'_{\mathsf{final}} = \ell_{\mathsf{final}})$.

Claim 6.6. *$(X', Y', L', \mathsf{AUX}')$ is a T-admissible leaky (n_1, d, k_1, d) source distribution with respect to* CRS.

Proof. We prove that $(X', Y', L', \mathsf{AUX}')$ satisfies the following.

Efficient Sampling. By construction, for every $\mathsf{crs} \in \mathsf{Supp}(\mathrm{CRS})$, $(Y', \mathsf{AUX}', L'_{\mathsf{init}}|\mathsf{crs})$ can be sampled in time $\mathrm{poly}(T(\lambda))$. Additionally, for every $\ell'_{\mathsf{init}} \in \mathsf{Supp}(L_{\mathsf{init}}|\mathsf{crs})$, $(X', L'_{\mathsf{final}}|\mathsf{crs}, \ell'_{\mathsf{init}}) = (X_1, L_{\mathsf{final}}|\mathsf{crs}, \ell_{\mathsf{init}})$, which can also be sampled in time $\mathrm{poly}(T(\lambda))$.

Min-entropy. Recall that Y' is sampled uniformly at random in $\{0,1\}^d$, independently of (crs, ℓ'). Therefore,

$$H_\infty(Y'|\mathrm{CRS}, L') = d.$$

Next, we observe that

$$H_\infty(X'|\mathrm{CRS}, L') = H_\infty(X|\mathrm{CRS}, L) = k_1.$$

Independence. Since aux' is sampled as a function of $(y', \mathsf{crs}, \ell')$ independent of X', it follows that for every $\mathsf{crs} \in \mathrm{CRS}$ and $\ell' \in \mathsf{Supp}(L(\cdot))$, X' and (Y', AUX') are independent conditioned on (crs, ℓ').

This proves that $(X', Y', L', \mathsf{AUX}')$ is a T-admissible leaky (n_1, d, k_1, d) source distribution with respect to CRS. □

Next, we prove that the adversary $\mathcal{A}$ breaks the non-malleable extractor for $(X', Y', L', \mathsf{AUX}')$. To see this, first observe that $(\overline{\mathsf{aux}} \neq \bot) \iff (y' \in \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell'_{\mathsf{init}}))$, which implies

$$\Pr[\overline{\mathsf{aux}} \neq \bot] = \Pr[y' \in \mathsf{BAD\text{-}seed}(\mathsf{crs}, \ell'_{\mathsf{init}})] > \frac{1}{10p'(T')} \tag{21}$$

where the probabilities are over the randomness of sampling $\mathsf{crs} \leftarrow \mathrm{CRS}$, $\ell'_{\mathsf{init}} \leftarrow L'_{\mathsf{init}}|\mathsf{crs}$, and $y' \leftarrow \{0,1\}^d$, and where the equation on the right-hand-side follows by Claim (6.5) together with the fact that $L'_{\mathsf{init}}|\mathsf{crs} = L_{\mathsf{init}}|\mathsf{crs}$.

By the definition of BAD-seed, $\mathcal{A}$ has distinguishing advantage at least $\frac{1}{2p'(\lambda)}$ when $y' \in \mathsf{bad\text{-}seed}(\mathsf{crs}, \ell'_{\mathsf{init}})$, or equivalently, when $\mathsf{aux}' \neq \bot$. This implies that

$$\begin{aligned}&\Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(\mathsf{cnm\text{-}Ext}(x', y', \mathsf{crs}), y', \mathsf{crs}, \ell', \overline{\mathsf{aux}}) = 1 \middle| (\overline{\mathsf{aux}} \neq \bot)\right] - \\ &\Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(U, y', \mathsf{crs}, \ell', \overline{\mathsf{aux}}) = 1 \middle| (\overline{\mathsf{aux}} \neq \bot)\right] \geqslant \frac{1}{2p'(T')}\end{aligned} \tag{22}$$

where the probabilities are over the randomness of sampling $\mathsf{crs} \leftarrow \mathrm{CRS}$, $\ell'_{\mathsf{init}} \leftarrow L'_{\mathsf{init}}|\mathsf{crs}$, $y' \leftarrow \{0,1\}^d$, $(x', \ell'_{\mathsf{final}}) \leftarrow (X', L'_{\mathsf{final}}|(\mathsf{crs}, \ell'_{\mathsf{init}})$, and where $\ell' = (\ell'_{\mathsf{init}}, \ell'_{\mathsf{final}})$.

Moreover, when $\overline{\mathsf{aux}} = \bot$, $\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}$ outputs 0. Combining this fact with Equation (21) and Equation (22), we have that for $(x', y', \ell', \overline{\mathsf{aux}})$ sampled as described above,

$$\Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(\mathsf{cnm\text{-}Ext}(x', y', \mathsf{crs}), y', \mathsf{crs}, \ell', \overline{\mathsf{aux}}) = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(U, y', \mathsf{crs}, \ell', \overline{\mathsf{aux}}) = 1\right] > \frac{1}{20(p'(T'))^2}$$

By Lemma 2.1, we have that no adversary running in time poly(T') can distinguish $(x', y', \mathsf{crs}, \ell', \overline{\mathsf{aux}})$ from $(x', y', \mathsf{crs}, \ell', \mathsf{aux}')$ with advantage better than neg(T'), and therefore,

$$\Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(\mathsf{cnm\text{-}Ext}(x', y', \mathsf{crs}), y', \mathsf{crs}, \ell', \mathsf{aux}') = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}^{y'}_{x',\mathsf{crs}}}(U, y', \mathsf{crs}, \ell', \mathsf{aux}') = 1\right] > \frac{1}{20(p'(T'))^2} - \mathrm{neg}(T')$$

where the probabilities are over the randomness of sampling $\mathsf{crs} \leftarrow \mathrm{CRS}$, $\ell'_{\mathsf{init}} \leftarrow L'_{\mathsf{init}}|\mathsf{crs}$, $(y', \mathsf{aux}') \leftarrow (Y', \mathsf{AUX}'|\mathsf{crs}, \ell'_{\mathsf{init}})$, $(x', \ell'_{\mathsf{final}}) \leftarrow (X', L'_{\mathsf{final}}|(\mathsf{crs}, \ell'_{\mathsf{init}})$, and where $\ell' = (\ell'_{\mathsf{init}}, \ell'_{\mathsf{final}})$.

Thus, the existence of $\mathcal{A}$ contradicts the fact that cnm-Ext is a strong (T, T')-computational non-malleable extractor for T-admissible leaky (n_1, d, k_1, d) source distribution $(X', Y', \mathsf{AUX}', L')$. □

References

[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.

[BCC+13] Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 341–360, 2013. URL: https://doi.org/10.1007/978-3-642-42045-0_18, doi:10.1007/978-3-642-42045-0_18.

[BH19] Joachim Breitner and Nadia Heninger. Biased nonce sense: Lattice attacks against weak ecdsa signatures in cryptocurrencies. Cryptology ePrint Archive, Report 2019/023, 2019. https://eprint.iacr.org/2019/023.

[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.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 285–298. ACM, 2016.

[CL16] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 158–167. IEEE, 2016.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 66–92. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-46494-6_4, doi:10.1007/978-3-662-46494-6_4.

[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.

[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. URL: https://doi.org/10.1109/FOCS.2004.44, 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. URL: https://doi.org/10.1137/060651380, doi:10.1137/060651380.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 601–610, 2009. URL: https://doi.org/10.1145/1536414.1536496, doi:10.1145/1536414.1536496.

[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.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of*

the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011, pages 99–108. ACM, 2011. URL: https://doi.org/10.1145/1993636.1993651, doi:10.1145/1993636.1993651.

[HDWH12] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In *Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012*, pages 205–220, 2012. URL: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger.

[JP14] Dimitar Jetchev and Krzysztof Pietrzak. How to fake auxiliary input. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 566–590. Springer, 2014. URL: https://doi.org/10.1007/978-3-642-54242-8_24, doi:10.1007/978-3-642-54242-8_24.

[KLR09] Yael Tauman Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 617–626. IEEE, 2009.

[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, 2008. URL: https://doi.org/10.1109/FOCS.2008.73, doi:10.1109/FOCS.2008.73.

[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.