Low Error Efficient Computational Extractors in the CRS Model

Ankit Garg[1(✉)], Yael Tauman Kalai[2(✉)], and Dakshita Khurana[3]

[1] Microsoft Research India, Bangalore, India
garga@microsoft.com
[2] Microsoft Research New England, Cambridge, USA
yael@microsoft.com
[3] University of Illinois Urbana-Champaign, Champaign, IL, USA
dakshita@illinois.edu

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 [5], 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 [1], we give a transformation converting our computational non-malleable extractor (in the CRS model) into a computational 2-source extractor for sources with low min-entropy (in the CRS model). Our 2-source extractor works for unbalanced sources: specifically, we require one of the sources to be larger than a specific polynomial in the other.
   This transformation does not incur any additional assumptions. Our analysis makes a novel use of the leakage lemma of Gentry and Wichs [18].

A. Canteaut and Y. Ishai (Eds.): EUROCRYPT 2020, LNCS 12105, pp. 373–402, 2020.
https://doi.org/10.1007/978-3-030-45721-1_14

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. In fact, Dodis *et al.* [15] 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 [6], who computed hundreds of Bitcoin private keys by exploiting the fact that the randomness used to generate them was imperfect (other examples include [3,20]).

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 have been 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 $k = \mathsf{polylog}(n)$. 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 $k \geq 0.499n$ [4,26]. Over the last three years, there has been remarkable and exciting progress [2,7–9,11–14,24], 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 $E{:}\{0,1\}^{n_1} \times \{0,1\}^{n_2} \rightarrow \{0,1\}^m$ 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

[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 $x \in \{0,1\}^n$, $\Pr[X = x] \leq 2^{-k}$.

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* $\mathrm{poly}(1/\epsilon)$*!*

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 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 [22,23] 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 comparatively 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).

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

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 could ask why assuming the existence of a CRS is reasonable, 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 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 (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 $n_1 = \Omega(\lambda), n_2 \leq \lambda^{\delta}$ and min-entropy $k_1 = n_1^{\epsilon}, k_2 = \log^{c}(\lambda)$.*

[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 distinguish 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.

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 [17]. 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* (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 $(E(X,Y,\mathrm{CRS}), \mathrm{CRS}, Y)$ 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 $E(X,Y',\mathrm{CRS})$. Clearly, such adversaries can obtain $E(X,Y',\mathrm{CRS})$ 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 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:*

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

Theorem 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:*

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

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. This will be used in Sect. 6 to convert a non-malleable extractor (in the high entropy regime) into a 2-source extractor (in the low entropy regime). We refer the reader to Sects. 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 [5], 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 [1], 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 [1] 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 [18]. We elaborate on this in Sect. 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 [5], however, this time we use our *computational* 2-source extractor as a building block. To argue security, we prove that the [5] transformation goes through even if we start with a *computational* 2-source extractor. As above, many technical challenges 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 [5], 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 [5] 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 technical difficulties which we elaborate on in Sect. 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 [5] 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 negligible error (eg., [4,26]).

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 $\mathsf{2Ext}$.

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 $\mathsf{2Ext}$ 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 [5]. 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 $\text{nm-Ext}(x, y', \mathsf{crs})$. 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

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

[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.

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 $h(y)$ and $Z = (Z_1, \ldots, Z_k)$. In addition, by setting the image of the lossy functions to be small enough, we can guarantee that X also has high min-entropy conditioned on $h(y)$ and $Z = (Z_1, \ldots, Z_k)$. Moreover, it is easy to seet that X and Y remain independent conditioned on $h(Y)$ and Z. Thus, we can use the fact that $\mathsf{2Ext}$ 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 satisfy 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-exponential 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 Sect. 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 [1], 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] Unfortunately, all known (information theoretic) non-malleable extractors require the seed length to be at least $t \log(1/\epsilon)$.

We note that in Sect. 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 [4,26] as a building block). This extractor, in particular, can be seen as a non-malleable *seeded* extractor. Importantly, it satisfies the requirements of [1], since in our construction the seed length is independent of t. Thus, one would expect that instantiating the [1] transformation with our computational non-malleable extractor (in the CRS model), would directly yield a

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

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 [1] crucially requires the underlying non-malleable extractor to be secure against adversaries that run in *unbounded time*. Specifically, even given an efficient adversary that contradicts the security of the 2-source extractor, [1] 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 [1] 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 of the leakage lemma of Gentry and Wichs [18].

The blueprint of [1]. To better understand these technicalities, we begin by describing the transformation of [1]. 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 $\geq K$, it holds that the size of the set of neighbors of A under Γ is at least K'.

The [1]-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 [1], 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

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

   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 statistically close to uniform, even given nm-Ext$(X_1, \Gamma(x_2, j))$ for every $j \in [t] \backslash \{i\}$ such that $\Gamma(x_2, j) \neq y$, which in turn implies that the XOR of these (distinct) values is close to uniform, which implies that $\mathsf{2Ext}(X_1, x_2)$ 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 $x_2 \in \mathsf{bad}, i \in [t]$ 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 [1].

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 of the [1] 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,

$$(\mathsf{cnm\text{-}Ext}(X_1, y), \mathsf{cnm\text{-}Ext}(X_1, y'_1), \ldots, \mathsf{cnm\text{-}Ext}(X_1, y'_t)) \approx (U, \mathsf{cnm\text{-}Ext}(X_1, y'_1), \ldots, \mathsf{cnm\text{-}Ext}(X_1, y'_t)) ,$$

   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 $\mathsf{cnm\text{-}Ext}$) 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 $\widehat{y} = \Gamma(\widehat{x}_2, i)$.
- 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 $\mathsf{cnm\text{-}Ext}$, 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.

Our key ideas. 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 1 [10,18,21]. *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,*

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

By Lemma 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 $O(2^{|\widehat{x}_2|})$.

[8] Jumping ahead, this is the reason that we end up with a 2-source extractor for unbalanced sources (see Theorem 3).

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 manage to apply the leakage lemma in a more sophisticated way. Recall that the adversary outlined above queries its oracle on $\{y_j\}_{j\in[t]\setminus\{i\}}$, where $y_j = \Gamma(\widehat{x}_2, j)$ and where $\widehat{x}_2 \leftarrow X_2$ such that $\widehat{y} = \Gamma(\widehat{x}_2, i)$. Importantly, we show that the elements in $\{y_j\}_{j\in[t]}$ form a hash collision only with negligible probability, assuming the sources for the 2-source extractor are somewhat efficiently sampleable. Otherwise, it would be possible to break the hash function in time proportional to that required to sample sources for the 2-source extractor.

Thus, in order to use the leakage lemma effectively, we prove a stronger form of security of our non-malleable extractor: we show that it is secure against adversaries that potentially run in time larger than the time against which the hash function is secure; as long as these adversaries do not query the oracle of the non-malleable extractor on hash collisions. By setting the parameters appropriately, this allows us to use the leakage lemma, and thus complete the argument outlined above. We therefore get a construction of a 2-source extractor, by relying on a non-malleable extractor that is secure against adversaries running in time $O(2^{|\widehat{y}|})$, as long as they do not make hash collision queries.

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

In Sect. 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 [4] or [26], we get a computational non-malleable extractor (in the CRS model) for sources in the high min-entropy regime.

In Sect. 6 we show how to convert our computatational non-malleable extractor (in the CRS model) into a computational 2-source extractor (in the CRS model) in the low entropy regime. Finally, we obtain a computational non-malleable extractor (in the CRS model) in the low entropy regime, by applying the transformation from Sect. 5 to the computational 2-source extractor that we constructed in Sect. 6.

3 Preliminaries

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

Definition 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*

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

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*

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

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

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

In this paper, we consider sources with average conditional min entropy, as defined in [16] (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 4 [16]**.** *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 [16]. Let X, Y and Z be three distributions, where 2^b is the number of elements in the support of Y. Then,

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

Claim. Let X, Y and Z be three distributions, then

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

We defer the proof of this claim to the full version.

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

3.1 Collision Resistan Hash Functions

In this work we rely on the existence of a collision resistant function family. Our setting of parameters is slightly non-standard, since our input domain may differ from the security parameter.

Definition 5 ($T(\lambda)$-secure collision resistant hash functions). *Let $n, k : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter, and let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of functions where for every $\lambda \in \mathbb{N}$ and every $h \in \mathcal{H}_\lambda$,*

$$h : \{0,1\}^{n(\lambda)} \to \{0,1\}^{k(\lambda)}.$$

This function family is said to be a $T(\lambda)$-secure collision resistant hash family if for every $\mathrm{poly}(T(\lambda))$-size adversary $\mathcal{A}$ there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$,

$$\Pr_{h \leftarrow \mathcal{H}_\lambda}[\mathcal{A}(h) = (x_1, x_2) \ s.t. \ (x_1 \neq x_2) \ \wedge \ h(x_1) = h(x_2)] = \nu(T(\lambda)).$$

Theorem 4. *Assuming sub-exponential hardness of DDH, there exists a constant $\delta > 0$ such that for every pair of polynomials $n, k : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{poly}(\lambda) \geq n(\lambda) > k(\lambda) \geq \Omega(\lambda)$ and for $T(\lambda) = 2^{\lambda^\delta}$, there exists a $T(\lambda)$-secure collision resistant hash family $\mathcal{H}_\lambda$, where for every $h \in \mathcal{H}_\lambda$, $h : \{0,1\}^{n(\lambda)} \to \{0,1\}^{k(\lambda)}$.*

3.2 Lossy Functions

Lossy functions were defined by Peikert and Waters in [25]. 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 6 (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 [25] (to ensure that the input and output lengths of the functions are the same for every $n = \mathrm{poly}(\lambda)$), [5] 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 2 [5,25]**.** *For any constant* $\epsilon > 0$ *there exists a constant* $\delta > 0$ *such that for every* $\Omega(\lambda) \leq n(\lambda) \leq \mathrm{poly}(\lambda)$ *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.*

3.3 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 [10,18,21]**.** *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,

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

3.4 Dispersers

Definition 7. *A function* $\Gamma : [N] \times [t] \to [D]$ *is a* (K, K') *disperser if for every* $A \subseteq [N]$ *with* $|A| \geq K$ *it holds that* $\left|\bigcup_{a \in A, i \in [t]} \{\Gamma(a,i)\}\right| \geq K'$.

We will rely on dispersers which follow from the known constructions of seeded extractors (e.g. [19]).

Theorem 5 (e.g. [19]**).** *There exists a constant* c *such that the following holds. For every* N, K, K', D *such that* $D \leq \sqrt{K}$ *and* $K' \leq D/2$, *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 definitions, 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 explicitly, 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 definitions 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 8 (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 $\{\mathrm{CRS}_\lambda\}_{\lambda\in\mathbb{N}}$ 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 $\mathrm{poly}(T)$.*

- $H_\infty(X_\lambda|\mathrm{CRS}_\lambda, L_\lambda) \geq k_1$ *and* $H_\infty(Y_\lambda|\mathrm{CRS}_\lambda, L_\lambda) \geq k_2$.
- *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.*[10]
- *For every* $\mathrm{aux} \in \mathsf{Supp}(\mathsf{AUX}_\lambda), |\mathrm{aux}| = O(\log T(\lambda))$[11].

Definition 9 (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-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* $\mathcal{A}$*,*

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

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

Definition 10 (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-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*, for every polynomial* p*, there exists a negligible function* $\nu(\cdot)$ *such that for every* λ *and every* $p(T(\lambda))$*-size adversary* $\mathcal{A}$*,*

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

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

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

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

We will occasionally need to impose a different requirement on the error distribution. In such cases we specify the error requirement explicitly. Specifically, we say that a (n_1, n_2, k_1, k_2) strong T-computational two source (or non-malleable) extractor has error $\mathrm{neg}(T'(\lambda))$ if it satisfies Definition 9 (or Definition 10), where the adversary's distinguishing advantage is required to be at most negligible in $T'(\lambda)$.

For our constructions, we will rely on the following theorem from [26] (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 6 [26]**.** *There exists a* (n_1, n_2, k_1, k_2) *strong statistical 2-source extractor according to Definition 9 where* $n_2 = \omega(\log n_1)$*,* $k_1 \geq \log n_1$*, and* $k_2 \geq \alpha n_2$ *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 our construction of computational non-malleable extractors in the CRS model, and prove the following theorem.

Theorem 7. *Let* $T, T', n_1, n_2, k_1, k_2, k_3, w : \mathbb{N} \to \mathbb{N}$ *be functions of the security parameter, where* $T \geq 2^{k_3}$ *and such that the following primitives exist.*

- *A* (n_1, n_2, k_1, k_2) *strong* T*-computational 2-source extractor with 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 6, where* $w = n_1 - n_1^\gamma$ *for some constant* $\gamma \in (0,1)$*.*
- *A* T'*-secure family of collision resistant hash functions* $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ *with* $h : \{0,1\}^{n_2} \to \{0,1\}^{k_3}$*.*

Then there exists a (n_1, n_2, K_1, K_2) *strong* T'*-computational non-malleable extractor, satisfying Definition 10 for* $K_1 = k_1 + k_3(n_1 - w + 1) + 1$*,* $K_2 = k_2 + k_3 + 1$*.*

Before we describe the construction (Sect. 5.1), we point out that the guarantees of the non-malleable extractor from Theorem 7 are not sufficient to instantiate the compiler in Sect. 6. To this end, we prove (Sect. 5.2) that our non-malleable extractor construction satisfies a stronger (yet more technical) property which turns out to be sufficient.

5.1 Construction

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-computational 2-source extractor with respect to the security parameter 1^λ.
2. Sample $h \leftarrow \mathcal{H}_\lambda$.
3. Sample $b = (b_1, \ldots, b_{k_3}) \leftarrow \{0,1\}^{k_3}$.
4. Sample independently k_3 pairs of random injective functions from $\mathcal{F}_\lambda$,

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

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

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

Output

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

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

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

As mentioned above, Theorem 7 is insufficient for instantiating our compiler (in Sect. 6) which converts a non-mallealbe extractor into a 2-source extractor. Rather, we need the non-malleable extractor to have the following more general (and more complex) guarantee, which is tailored to our construction (in Sect. 5.1): If the underlying 2-source extractor $\mathsf{2Ext}$ is T-secure (for $T \geq 2^{k_3}$) then the resulting non-malleable extractor is also T-secure with error $\mathrm{neg}(2^{k_3})$, assuming the adversary (i.e., distinguisher) does not query its oracle on y' such that $h(y) = h(y')$. We next formalize this guarantee, and begin by defining the notion of an $\mathcal{H}$-admissible adversary corresponding to our non-malleable extractor from Sect. 5.1.

Definition 11 ($\mathcal{H}$-Admissible Adversary). *We say that an adversary $\mathcal{A}$ is $\mathcal{H}$-admissible if on any input $(v, y, \mathsf{crs}, \ell, \mathrm{aux})$ (where v is either $\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs})$ or a uniformly random string), it does not query its oracle $\mathcal{O}^y_{x,\mathsf{crs}}$ with y' such that $h(y') = h(y)$, where h is the hash function in crs.*

Theorem 8. *Let $T, n_1, n_2, k_1, k_2, k_3, w : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter, and let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ with $h : \{0,1\}^{n_2} \to \{0,1\}^{k_3}$ be a family of functions. Assume that $T \geq 2^{k_3}$ and the following primitives exist.*

- *A (n_1, n_2, k_1, k_2) strong T-computational 2-source extractor 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 6, where $w = n_1 - n_1^\gamma$ for some constant $\gamma \in (0,1)$.*

Then the extractor constructed in Sect. 5.1 is a (n_1, n_2, K_1, K_2) strong T-computational non-malleable extractor with error $\mathrm{neg}(2^{k_3})$ against $\mathcal{H}$-admissible adversaries, for $K_1 = k_1 + k_3(n_1 - w + 1) + 1, K_2 = k_2 + k_3 + 1$.

Corollary 1 instantiates Theorem 8 with the 2-source extractor from Theorem 6; this corollary will be used in the next section.

Corollary 1. *Let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda\in\mathbb{N}}$ with $h : \{0,1\}^{n_2} \to \{0,1\}^{k_3}$ be a family of functions. Assume the sub-exponential hardness of DDH, and fix any constant $\epsilon > 0$. Then there exists a constant $\delta > 0$ such that for any parameters (n_1, n_2, K_1, K_2) satisfying*

$$\Omega(\lambda) \leq n_1 \leq \mathrm{poly}(\lambda), \quad n_2 = \omega(\log n_1), \quad K_1 = n_1^\epsilon, \quad \text{and } K_2 = 0.51 n_2$$

there exists a (n_1, n_2, K_1, K_2) strong T-computational non-malleable extractor with error $\mathrm{neg}(2^{k_3})$ against $\mathcal{H}$-admissible adversaries (satisfying Definition 10) for $T(\lambda) = 2^{\lambda^\delta}$ and $k_3 \leq \min\{\lambda^\delta, n_1^{\epsilon/2}, n_2^{0.9}\}$.

Proof of Corollary 1. Fix a constant $\epsilon > 0$, and fix $n_1 = n_1(\lambda)$ and $n_2 = n_2(\lambda)$ as in the statement of Corollary 1. By Lemma 2, the sub-exponential hardness of DDH (together with the restrictions on n_1 and n_2) implies that there exists a constant $\delta > 0$ for which there exists 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 $n_1 - w = n_1^{\epsilon/3}$.

By Theorem 6, for $n_2 = \omega(\log n_1)$, 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-computational 2-source extractor in the CRS model (where the CRS is empty).

Note that by our setting of parameters $T \geq 2^{k_3}$. Therefore, by Theorem 8, there exists a (n_1, n_2, K_1, K_2) strong T-computational non-malleable extractor with error $\mathrm{neg}(2^{k_3})$ against $\mathcal{H}$-admissible adversaries, where $K_1 = k_1 + k_3(n_1 - w + 1) + 1 \leq n_1^{\epsilon/3} + n_1^{\epsilon/2} \cdot n_1^{\epsilon/3} + 1 \leq n_1^\epsilon$ and $K_2 = k_2 + k_3 + 1 \leq 0.501 n_2 + n_2^{0.9} + 1 \leq 0.51 n_2$, as desired. □

5.2 Analysis

In this section, we prove Theorem 8; namely, we prove the T-security of the non-malleable extractor against $\mathcal{H}$-admissible adversaries. The proof of Theorem 7 follows from the observation that every adversary $\mathcal{A}$ that runs in time poly(T') on input sources sampled in time poly(T'), cannot query the oracle on hash collisions, except with probability neg(T'), and thus is $\mathcal{H}$-admissible (except with probability neg(T')).

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. Note that since our adversary is assumed to be $\mathcal{H}$-admissible, it cannot distinguish between these two oracles since it never makes such a query. Then 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, we design a distribution for the 2-source extractor and break it using the supposed adversary for the non-malleable extractor.

Proof (of Theorem 8). In this 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 poly(T)-size $\mathcal{H}$-admissible adversary $\mathcal{A}$, such that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned} &\Pr[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathrm{aux}) = 1] - \\ &\Pr[\mathcal{A}^{\mathcal{O}^y_{x,\mathsf{crs}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1] \geq \frac{1}{p(2^{k_3})}, \end{aligned} \tag{2}$$

where the probabilities are over $(\mathsf{crs}, x, y, \ell, \mathrm{aux}) \leftarrow (\mathrm{CRS}, X, Y, L, \mathsf{AUX})$ 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

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

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

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

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.

Since $\mathcal{A}$ is $\mathcal{H}$-admissible, by definition, $\mathcal{A}$ does not generate a query $y' \neq y$ such that $h(y') = h(y)$, and therefore, the oracles are indistinguishable. This, together with Eq. (2), implies that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned} &\Pr[\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathrm{aux}) = 1] - \\ &\Pr[\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1] \geq \frac{1}{p(2^{k_3})} \end{aligned} \tag{3}$$

where the probabilities are over $(\mathsf{crs}, x, y, \ell, \mathrm{aux}) \leftarrow (\mathrm{CRS}, X, Y, L, \mathsf{AUX})$ and over uniformly distribution $U \leftarrow \{0,1\}^m$. Next, the T-security of the lossy function family, together with the assumption that $T \geq 2^{k_3}$, implies that for every poly(T)-size adversary $\mathcal{B}$ (recall $b \in \{0,1\}^{k_3}$ is used to determine which functions are lossy or injective in the crs),

$$2^{-k_3} + \mathrm{neg}(T) \geq \Pr[\mathcal{B}(\mathsf{crs}) = b] \geq 2^{-k_3} - \mathrm{neg}(T). \tag{4}$$

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

$$2^{-k_3} + \mathrm{neg}(T) \geq \Pr\left[h(y) = b\right] \geq 2^{-k_3} - \mathrm{neg}(T), \tag{5}$$

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

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

$$\begin{aligned} &\Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \\ &- \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \geq \frac{1}{4p(2^{k_3})} \end{aligned} \tag{6}$$

The proof of this claim appears in the full version of our paper.

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

$$\begin{aligned} &\Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \wedge \left(h(y) = b\right)\right] \\ &- \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \wedge \left(h(y) = b\right)\right] \\ &= \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(\mathsf{cnm\text{-}Ext}(x, y, \mathsf{crs}), y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \cdot \Pr\left[h(y) = b\right] \\ &- \Pr\left[\left(\mathcal{A}^{\widetilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \middle| \left(h(y) = b\right)\right] \cdot \Pr\left[h(y) = b\right] \\ &\geq \frac{1}{4p(2^{k_3})} \cdot (2^{-k_3} - \mathrm{neg}(2^{k_3})) \geq \frac{1}{p''(2^{k_3})} \end{aligned} \tag{7}$$

where the last inequality holds for some polynomial $p''(\cdot)$.

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 Eq. (7), we conclude that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}\left(\mathsf{2Ext}(f_{1,h(y)_1} \circ f_{2,h(y)_2} \circ \ldots \circ f_{k_3,h(y)_{k_3}}(x), y, \mathsf{crs}_{\mathsf{2Ext}}), y, \mathsf{crs}, \ell, \mathrm{aux}\right) = 1\right) \wedge \left(h(y) = b\right)\right] - \Pr\left[\left(\mathcal{A}^{\tilde{\mathcal{O}}}(U, y, \mathsf{crs}, \ell, \mathrm{aux}) = 1\right) \wedge \left(h(y) = b\right)\right] \geq \frac{1}{p''(2^{k_3})} \tag{8}$$

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

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 $(y, \ell_{\mathsf{init}}, \mathrm{aux}) \leftarrow (Y, L_{\mathsf{init}}, \mathsf{AUX}|\mathsf{crs})$.
   (d) Set $(y', \mathrm{aux}') = (y, \mathrm{aux})$.
   (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. $(X', Y', L', \mathsf{AUX}')$ is a T-admissible leaky (n_1, n_2, k_1, k_2) source distribution with respect to $\mathrm{CRS}_{\mathsf{2Ext}}$, where $k_1 = K_1 - k_3 \cdot (n_1 - w + 1) - 1$ and $k_2 = K_2 - k_3 - 1$.

The proof of this claim appears in the full version of our paper.

We next argue that Equation (8), together with the definition of the distribution $(X', Y', L', \mathsf{AUX}'|\mathsf{crs2Ext})$, implies that there exists a T-size adversary $\mathcal{A}'$, that simulates the adversary $\mathcal{A}$, as well as its oracle, such that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{A}'(\mathsf{2Ext}(X', Y', \mathsf{crs}_{\mathsf{2Ext}}), y', \mathsf{crs}_{\mathsf{2Ext}}, \ell', \mathrm{aux}') = 1] - \Pr[\mathcal{A}'(U, y', \mathsf{crs}_{\mathsf{2Ext}}, \ell', \mathrm{aux}') = 1] \geq 1/\mathrm{poly}(2^{k_3}). \tag{9}$$

The algorithm $\mathcal{A}'$ on input $(\alpha, y', \mathsf{crs}_{\mathsf{2Ext}}, \ell', \mathrm{aux}')$ 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 $\mathcal{A}^{\widetilde{\mathcal{O}}}(\alpha, y', \mathsf{crs}, \ell, \mathrm{aux}')$, where $\widetilde{\mathcal{O}}$ is simulated using $(h(y), z_{x,h(y)}, \mathsf{crs})$.

Equation (8) implies that indeed Eq. (9) holds, as desired. This contradicts the fact that 2Ext is a strong T-computational 2-source extractor for $(X', Y', L', \mathsf{AUX}')$. This completes the proof of Theorem 8.

6 Computational Strong 2-Source Extractors in the CRS Model

In this section, we describe our compiler that converts a 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 [1]. 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. Due to these challenges, our compiler is not as general as the one in the information theoretic setting. In particular, in Theorem 9 below, we use as an ingredient a collision resistant hash family $\mathcal{H}$, and show how to convert a computational non-malleable extractor that is secure against $\mathcal{H}$-admissible adversaries (such as the one from Theorem 8) into a computational 2-source extractor.

Theorem 9. *Let $T, T', n_1, n_2, k_1, k_2, k_3, d : \mathbb{N} \to \mathbb{N}$ be functions of the security parameter, such that $T = (T')^{\omega(1)}, T = \lambda^{\Omega(1)}$, $n_2 = O(\log T), k_2 = \omega(\log T')$, and such that the following primitives exist.*

- *A family of T'-secure collision-resistant hash functions functions $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ with $h : \{0,1\}^d \to \{0,1\}^{k_3}$*
- *A (n_1, d, k_1, d) strong T-computational non-malleable extractor against $\mathcal{H}$-admissible adversaries in the CRS model with error $\mathrm{neg}(2^{k_3})$, where the CRS is generated by sampling $h \leftarrow \mathcal{H}$ and sampling $\mathsf{crs}' \leftarrow \mathrm{CRS}'$, where CRS' is a $\mathrm{poly}(T)$-time sampleable distribution, and setting $\mathsf{crs} = (h, \mathsf{crs}')$. This non-mallealbe extractor is 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 $t = \mathrm{poly}(\lambda)$ (according to Definition 7).

Then there exists a $(n_1, n_2, k_1, 2k_2)$ strong T'-computational 2-source extractor in the CRS model (according to Definition 9).

We defer the construction of the 2-source extractor from Theorem 9 to Sect. 6.1, and defer the analysis to Sect. 6.2. In what follows we present two corollaries. Corollary 2 instantiates Theorem 9 with the non-malleable extractor from Corollary 1.

Corollary 2. *Fix any constant $\epsilon > 0$. Then assuming the sub-exponential hardness of the DDH assumption, there exists a constant $\delta > 0$ such that for any constant $c \geq 1$ and any parameters n_1, n_2, k_1, k_2, T' satisfying*

$$\Omega(\lambda) \leq n_1 \leq \text{poly}(\lambda),\ \lambda^{O(1)} \leq n_2 \leq O(\lambda^\delta),\ k_1 = n_1^\epsilon,\ k_2 = \log^{c/\delta} n_2,\ T' = 2^{\log^c \lambda}$$

there exists a (n_1, n_2, k_1, k_2) strong T'-computational 2-source extractor in the CRS model (satisfying Definition 9).

Proof. Fix any constant $\epsilon > 0$. By Corollary 1, there exists a constant $\delta > 0$ for which there exists a (n_1, d, K_1, d) strong T-computational non-malleable extractor with error $\text{neg}(2^{k_3})$ in the CRS model against $\mathcal{H}$-admissible adversaries, for $\mathcal{H} : \{0,1\}^d \to \{0,1\}^{k_3}$, where $T = 2^{\lambda^\delta}$, $k_3 = \min\{n_1^{\epsilon/2}, d^{0.9}, \lambda^\delta\}$, and for any n_1, d, K_1 such that

$$\Omega(\lambda) \leq n_1 \leq \text{poly}(\lambda),\ d = \omega(\log n_1), K_1 = n_1^\epsilon$$

Moreover, this is the computational non-malleable extractor from Construction 5.1 where the crs is distributed as required in the theorem statement.

Next, fix any n_1 such that $\Omega(\lambda) \leq n_1 \leq \text{poly}(\lambda)$. By Theorem 5, 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, T' that satisfy

$$k_2 \geq 2d + \log^2 T'. \tag{10}$$

Fix any constant $c \geq 1$, let $k_2 = \log^{c/\delta} n_2$ and let $T' = 2^{\log^c \lambda}$. Set $d = k_2/4$. Note that Eq. (10) is satisfied by the definition of d and T'. Also,

$$k_3 = \min\{\lambda^\delta, n_1^{\epsilon/2}, d^{0.9}\} = \Omega((\log \lambda)^{0.9c/\delta}).$$

Therefore, assuming the sub-exponential hardness of DDH, and setting the security parameter in Theorem 4 to be $\kappa = k_3$, we conclude that there exists a constant δ' such that there exists a $2^{k_3^{\delta'}}$-secure collision resistant hash $\mathcal{H} : \{0,1\}^d \to \{0,1\}^{k_3}$. Assume without loss of generality that $\delta \leq 0.9\delta'$ (otherwise, reduce the size of δ). This implies that $T' \leq 2^{k_3^{\delta'}}$.

Theorem 9 implies that there exists a $(n_1, n_2, k_1, 2k_2)$ strong T'-computational 2-source extractor in the CRS model, as long as $n_2 = O(\log T) = O(\lambda^\delta)$, and as long as $k_2 = \omega(\log T')$ and in particular for $k_2 = \log^{c/\delta} n_2$.

By using the 2-source extractor obtained as a result of Corollary 2 to instantiate the non-malleable extractor in Theorem 7, we obtain the following corollary:

Corollary 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'-computational non-malleable extractor satisfying Definition 10 whenever*

$$\Omega(\lambda) \leq n_1 \leq \text{poly}(\lambda),\ \lambda^{O(1)} \leq n_2 \leq O(\lambda^{\delta}),\ K_1 = n_1^{\epsilon},\ K_2 = \log^{1/\delta^2} n_2, T' = \lambda.$$

Proof. Fix n_1, n_2 as in the statement of the corollary. Fix any constant $\epsilon > 0$. By Corollary 2, assuming the sub-exponential hardness of DDH, there exists a constant $\delta > 0$ such that for any constant $c \geq 1$, there exists a (n_1, n_2, k_1, k_2) strong T-computational 2-source extractor for k_1, k_2, T satisfying

$$k_1 = n_1^{\epsilon/3},\ k_2 = \log^{c/\delta} n_2,\ T = 2^{\log^c \lambda}.$$

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

- A (T, n_1, w)-lossy function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ where each $f \in \mathcal{F}_\lambda$ is of the form $f : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$, where $T(\lambda) = 2^{\log^c \lambda}$ as above and w is such that $n_1 - w = n_1^{\epsilon/3}$. This follows from Lemma 2.
- 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 $k_3 = \log^{1/\delta} \lambda$, that is secure against $\text{poly}(\lambda)$-size adversaries (this follows by setting the security parameter to be $\kappa = k_3$ in Theorem 4.[12])

Set $c \geq \frac{1}{\delta}$ which implies that $T \geq 2^{k_3}$. Therefore, by Theorem 7, there exists a (n_1, n_2, K_1, K_2) strong T'-computational non-malleable extractor for $T' = \lambda$ for $K_1 = k_1 + k_3(n_1 - w) + 1 \leq n_1^{\epsilon/3} + \log^{1/\delta} \lambda \cdot n_1^{\epsilon/3} + 1 < n_1^{\epsilon}$, and thus in particular for $K_1 = n_1^{\epsilon}$, and for $K_2 = k_2 + k_3 + 1 = \log^{c/\delta} n_2 + (\log \lambda)^{1/\delta} + 1$, and thus for $K_2 = \log^{c/\delta'} n_2$ for any constant $\delta' < \delta$. The corollary follows by reassigning δ to be δ'.

6.1 Construction

In what follows, we construct the 2-source extractor from Theorem 9. To this, end, fix any parameters $T, T', n_1, n_2, k_1, k_2, d$ according to Theorem 9. Fix any collision-resistant hash function $\mathcal{H}$ and a (n_1, d, k_1, d) strong T-computational non-malleable extractor against $\mathcal{H}$-admissible adversaries in the CRS model

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

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

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

[12] We assume that δ is small enough so that the hash function is $2^{\kappa^{\delta}}$ secure.

Define a 2-source extractor

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

by

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

6.2 Analysis

We prove the security of the 2-source extractor 2Ext described above in several steps. We start by assuming (for contradiction) that there exists an adversary running in time poly(T') that breaks the 2-source extractor 2Ext on a specific $(n_1, n_2, k_1, 2k_2)$ T'-admissible leaky source distribution. Using this adversary, we define an adversary that breaks the non-malleable extractor (on a distribution to be defined later). To this end, we define the sets BAD-rand and BAD-seed. These capture the places where the adversary breaks the non-malleable extractor. Next, we prove that these sets are large. Finally we define the distribution on which the adversary breaks the non-malleable extractor. This relies on the leakage lemma. The complete proof appears in the full version of our paper.

Acknowledgement. We thank Maciej Obremski and João Ribeiro for pointing out a subtle error in an initial draft of this work.

References

1. Ben-Aroya, A., Chattopadhyay, E., Doron, D., Li, X., Ta-Shma, A.: Low-error, two-source extractors assuming efficient non-malleable extractors. In: CCC (2017)
2. Ben-Aroya, A., Doron, D., Ta-Shma, A.: Explicit two-source extractors for near-logarithmic min-entropy. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 23, p. 88 (2016)
3. Bernstein, D.J., et al.: Factoring RSA keys from certified smart cards: Coppersmith in the wild. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 341–360. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_18
4. Bourgain, J.: More on the sum-product phenomenon in prime fields and its applications. Int. J. Number Theory **1**, 1–32 (2005)
5. Braverman, M., Hassidim, A., Kalai, Y.T.: Leaky pseudo-entropy functions. In: Innovations in Computer Science (2011)
6. Breitner, J., Heninger, N.: Biased nonce sense: lattice attacks against weak ECDSA signatures in cryptocurrencies. Cryptology ePrint Archive, Report 2019/023 (2019). https://eprint.iacr.org/2019/023
7. Chattopadhyay, E., Goyal, V., Li, X.: Non-malleable extractors and codes, with their many tampered extensions. In: Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, pp. 285–298. ACM (2016)
8. Chattopadhyay, E., Li, X.: Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In: 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pp. 158–167. IEEE (2016)

9. Chattopadhyay, E., Zuckerman, D.: Explicit two-source extractors and resilient functions. In: Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, pp. 670–683. ACM (2016)
10. Chung, K., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_4
11. Cohen, G.: Local correlation breakers and applications to three-source extractors and mergers. SIAM J. Comput. **45**(4), 1297–1338 (2016)
12. Cohen, G.: Making the most of advice: new correlation breakers and their applications. In: 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pp. 188–196. IEEE (2016)
13. Cohen, G.: Non-malleable extractors-new tools and improved constructions. In: LIPIcs-Leibniz International Proceedings in Informatics, vol. 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
14. Cohen, G.: Two-source extractors for quasi-logarithmic min-entropy and improved privacy amplification protocols. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 23, p. 114 (2016)
15. Dodis, Y., Ong, S.J., Prabhakaran, M., Sahai, A.: On the (im)possibility of cryptography with imperfect randomness. In: Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS 2004), Rome, Italy, 17–19 October 2004, pp. 196–205 (2004). https://doi.org/10.1109/FOCS.2004.44
16. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008). https://doi.org/10.1137/060651380
17. Dodis, Y., Wichs, D.: 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, 31 May–2 June 2009, pp. 601–610 (2009). https://doi.org/10.1145/1536414.1536496
18. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6–8 June 2011, pp. 99–108. ACM (2011). https://doi.org/10.1145/1993636.1993651
19. Guruswami, V., Umans, C., Vadhan, S.: Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. J. ACM (JACM) **56**(4) (2009). Article No. 20
20. Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J.A.: Mining your Ps and Qs: detection of widespread weak keys in network devices. In: Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012, pp. 205–220 (2012). https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger
21. Jetchev, D., Pietrzak, K.: How to fake auxiliary input. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 566–590. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_24
22. Kalai, Y.T., Li, X., Rao, A.: 2-source extractors under computational assumptions and cryptography with defective randomness. In: 2009 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, pp. 617–626. IEEE (2009)
23. Kalai, Y.T., Li, X., Rao, A., Zuckerman, D.: Network extractor protocols. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, 25–28 October 2008, pp. 654–663 (2008). https://doi.org/10.1109/FOCS.2008.73

24. Li, X.: Improved non-malleable extractors, non-malleable codes and independent source extractors. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pp. 1144–1156. ACM (2017)
25. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196 (2008)
26. Raz, R.: Extractors with weak random seeds. In: STOC, pp. 11–20 (2005)