Electronic Colloquium on Computational Complexity, Report No. 30 (2016)

Non-Malleable Extractors with Logarithmic Seeds

Gil Cohen*

March 7, 2016

Abstract

We construct non-malleable extractors with seed length $d = O(\log n + \log^3(1/\varepsilon))$ for n-bit sources with min-entropy $k = \Omega(d)$, where ε is the error guarantee. In particular, the seed length is logarithmic in n for $\varepsilon > 2^{-(\log n)^{1/3}}$. This improves upon existing constructions that either require super-logarithmic seed length even for constant error guarantee, or otherwise only support min-entropy $n/\mathrm{polylog}\, n$.

1 Introduction

A non-malleable extractor is a seeded extractor with a very strong guarantee concerning the correlations (or, more precisely, the lack thereof) of the outputs of the extractor when fed with different seeds. The notion of a non-malleable extractor was introduced by Dodis and Wichs [DW09], motivated by the problem of designing privacy amplification protocols against active adversaries. More recently, non-malleable extractors played a key role in the construction of two-source extractors [CZ15].

We turn to give the formal definition of non-malleable extractors. We assume familiarity with standard notions such as min-entropy, statistical distance, and weak-sources, and with standard notation. The unfamiliar reader may consult the Preliminaries.

Definition 1.1 (Non-malleable extractors)**.** *A function* $\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is called a* (k,ε)-non-malleable extractor *if for any* (n,k)*-source* X *and any function* $\mathcal{A}\colon \{0,1\}^d \to \{0,1\}^d$ *with no fixed points, it holds that*

$$(\mathsf{nmExt}(X,Y), \mathsf{nmExt}(X,\mathcal{A}(Y)), Y) \approx_\varepsilon (U_m, \mathsf{nmExt}(X,\mathcal{A}(Y)), Y),$$

where Y *is uniformly distributed over* $\{0,1\}^d$ *independently of* X*. If* nmExt *is a* (k,ε)*-non-malleable extractor, we say that* nmExt *has error guarantee* ε *and that* nmExt *supports min-entropy* k*.*

*Computing and Mathematical Sciences Department, Caltech. Email: coheng@caltech.edu.

ISSN 1433-8092

Computational aspects aside, for any integer n and $\varepsilon > 0$, Dodis and Wichs [DW09] proved the existence of (k, ε)-non-malleable extractors having m output bits and seed length $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$ for any $k > 2m + 2\log(1/\varepsilon) + \log d + O(1)$. Although the mere existence of non-malleable extractors, and with such great parameters, is somewhat surprising (and somewhat non-trivial to prove!), explicit constructions are far more desirable.

Constructing non-malleable extractors gained a significant attention in the literature. Already some of the early constructions [CRS14, DLWZ14, Li12a] have seed length $d = O(\log(n/\varepsilon))$, which is optimal up to a constant factor. However, these constructions could only support min-entropy higher than $n/2$. This restriction was subsequently relaxed to $(1/2 - \alpha) \cdot n$ for some small universal constant $\alpha > 0$ [Li12b]. In a breakthrough result [CGL15], Chattopadhyay, Goyal, and Li constructed a non-malleable extractor with seed length $d = O(\log^2(n/\varepsilon))$ that supports a drastically lower min-entropy $k = \Omega(\log^2(n/\varepsilon))$. Based on the [CGL15] framework, an improved construction of non-malleable extractors was given [Coh15b]. In particular, non-malleable extractors with seed length $d = O(\log(n/\varepsilon) \cdot \log(\log(n)/\varepsilon))$ that support min-entropy $k = \Omega(\log(n/\varepsilon))$. A second, incomparable, construction with seed length $d = O(\log n)$ was given in [Coh15b], though it could only support min-entropy $k = n/\mathrm{polylog}\, n$, for a slightly sub-constant ε.

To summarize, prior to this work, explicit non-malleable extractors with logarithmic seed length could only support high min-entropy ($k = n/\mathrm{polylog}\, n$). To support lower min-entropy (say, $k = \log n$, or $k = \mathrm{polylog}\, n$, or even $k = n^{0.9}$), regardless of the error guarantee, a seed of super-logarithmic length was required. In this work we improve upon existing constructions by devising non-malleable extractors with logarithmic seeds that support logarithmic min-entropy. Further, the error guarantee is sub-constant.

Theorem 1.2. *For any integer n and for any $\varepsilon > 0$ there exists an explicit (k, ε)-non-malleable extractor*

$$\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$$

with seed length $d = O(\log n + \log^3(1/\varepsilon))$ for $k = \Omega(d)$.

To the matter of fact, our construction has a more flexible tradeoff between the different parameters (see Lemma 4.1). Nevertheless, Theorem 1.2 is a clearly presentable instantiation of the more general result in a natural regime of parameters. Note, in particular, that for $\varepsilon > 2^{-(\log n)^{1/3}}$, both the seed length and the supported min-entropy are logarithmic in n.

The non-malleable extractor that is given by Theorem 1.2 is reported to output a single bit. In many scenarios, outputting one bit is not enough. By applying a result from [Coh15b] that allows one to increase the output length of a given non-malleable extractor in a black-box manner, we obtain the following.

Theorem 1.3. *For any integer n and for any $\varepsilon > 0$, there exists an explicit (k, ε)-non-malleable extractor*

$$\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with seed length $d = O(\log n + \log^3(1/\varepsilon) + \log k \cdot \log(1/\varepsilon))$ and $m = \Omega(k/\log(1/\varepsilon))$ output bits, where $k = \Omega(d)$.

Note that the seed length of the non-malleable extractor that is given by Theorem 1.3 is logarithmic in n in the natural regime $k = \operatorname{polylog} n$ when restricting to error guarantee $2^{-(\log n)^{1/3}}$.

2 Proof Overview

The proof of Theorem 1.2 is based on the framework that was introduced by Chattopadhyay, Goyal, and Li [CGL15], as well as on further ideas from [Coh15b], though applied in a more intricate manner. To outline our proof, we believe it is instructive to start by presenting the ideas of [CGL15, Coh15b]. We recall the two main primitives that played a key role in these constructions – *correlation breakers with advice* and *advice generators*.

2.1 Correlation breakers with advice and advice generators

Informally speaking, a correlation breaker with advice is a function that breaks correlations between a "good" random variable and an adversarially correlated random variable, given an "advice", and using an auxiliary weak-source of randomness. Somewhat more formally, a (k, ε)-correlation breaker with advice is a function

$$\mathsf{AdvCB}\colon \{0,1\}^w \times \{0,1\}^\ell \times \{0,1\}^a \to \{0,1\}^m$$

such that for any, arbitrarily correlated, ℓ-bit random variables Y, Y' such that Y is uniform; any arbitrarily correlated w-bit random variables X, X' that are jointly independent of (Y, Y'), and such that X has min-entropy k; and for any distinct a-bit strings $\alpha \neq \alpha'$, it holds that $\mathsf{AdvCB}(X, Y, \alpha)$ is ε-close to uniform in statistical distance even conditioned on $\mathsf{AdvCB}(X', Y', \alpha')$. We think of α as an "advice", and in particular refer to a as the advice length. We refer to X as the auxiliary weak-source of randomness, or simply as the source.

By adopting the construction of local correlation breakers [Coh15a] (that, in turn, was based on ideas from [Li13]), Chattopadhyay *et al.* [CGL15] constructed a correlation breaker with advice for any

$$\ell = \Omega\left(a \cdot \log\left(\frac{aw}{\varepsilon}\right)\right),$$

that supports min-entropy

$$k = \Omega\left(a \cdot \log\left(\frac{a\ell}{\varepsilon}\right)\right)$$

which has $m = \Omega(\ell/a)$ output bits (see Theorem 3.11).

We move to the notion of an advice generator. A (k, ε)-advice generator is a function

$$\mathsf{AdvGen}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^a$$

with the following property. For any (n, k)-source X and for any function $\mathcal{A}\colon \{0,1\}^d \to \{0,1\}^d$ with no fixed points, it holds that

$$\Pr_{(x,y)\sim(X,Y)}\left[\mathsf{AdvGen}(x, y) = \mathsf{AdvGen}(x, \mathcal{A}(y))\right] \leq \varepsilon,$$

where Y is uniformly distributed over d-bit strings independently of X.

2.2 The [CGL15] construction

Clearly, an advice generator is a strictly weaker object than a non-malleable extractor. Indeed, note that a (k, ε)-non-malleable extractor with m output bits is a $(k, \varepsilon + 2^{-m})$ advice generator. Nevertheless, Chattopadhyay *et al.* [CGL15] reduced the problem of constructing a non-malleable extractor to that of constructing an advice generator, at least as long as the advice generator is "nice". With the notations set above, we say that AdvGen is nice if conditioned on $\mathsf{AdvGen}(X, Y)$, $\mathsf{AdvGen}(X, \mathcal{A}(Y))$, the random variables X, Y remain independent and, furthermore, both X and Y have not lost much of their respective min-entropies.

The reduction suggested by [CGL15] can be written as

$$\mathsf{nmExt}(x, y) = \mathsf{AdvCB}(x, y, \mathsf{AdvGen}(x, y)).$$

That is, one uses the source x and the seed y to generate an advice that is then passed to the correlation breaker with advice which, in turn, also operates on x, y. Although a priori it is not clear why such a suggestion is valid, as the advice is correlated with the source and seed (in fact, it is completely determined by them!), this elegant reduction can be shown to work.

Let X be an (n, k)-source, $\mathcal{A} \colon \{0,1\}^d \to \{0,1\}^d$ be a function with no fixed points, and Y a random variable that is uniformly distributed over d-bit strings, independently of X. For ease of notation we write Y' for $\mathcal{A}(Y)$. The analysis proceeds as follows. As AdvGen is a nice advice generator, with high probability over the fixings $\alpha = \mathsf{AdvGen}(X, Y)$ and $\alpha' = \mathsf{AdvGen}(X, Y')$, it holds that $\alpha \neq \alpha'$, and furthermore, X and Y remain independent and have not lost much min-entropy. At this point, except for the fact that Y is no longer uniform but rather has very high min-entropy, all the conditions for applying the correlation breaker with advice are met. Luckily, Y having very high min-entropy is sufficient for the specific implementation of AdvCB being used.

Given this reduction from non-malleable extractors to nice advice generators, and the construction of correlation breakers with advice mentioned above, one can focus on the easier task of devising an advice generator. Note that the shorter the advice length a is, the better the resulting non-malleable will be in terms of seed length and supported min-entropy.

The following advice generator was suggested by [CGL15].[1] First, partition y to two substrings $y = y_1 \circ y_2$ such that y_1 has sufficient length so to be used as a seed for a strong seeded extractor Ext on n-bit strings with error guarantee ε. Using state of the art seeded extractors, $|y_1| = O(\log(n/\varepsilon))$ will do (see Theorem 3.4). We further assume that $|y| \geq 100 \cdot |y_1|$. Let ECC be an error correcting code with relative distance $1 - \varepsilon$, and define

$$\mathsf{AdvGen}(x, y) = y_1 \circ \mathsf{ECC}(y)_{\mathsf{Ext}(x, y_1)}.$$

In the expression above, by $\mathsf{ECC}(y)_{\mathsf{Ext}(x,y_1)}$ we mean the following – we interpret the output of Ext as an index i of the codeword $\mathsf{ECC}(y)$. Then, $\mathsf{ECC}(y)_i$ refers to the content in that i'th entry.

[1] The advice generator that we present in this section is a slightly modified version of the original generator that was given by [CGL15].

With the notations set above, the analysis proceeds as follows. First, note that if $Y_1 \neq Y_1'$ then, as Y_1 is a prefix of $\mathsf{AdvGen}(X, Y)$ and Y_1' is a prefix of $\mathsf{AdvGen}(X, Y')$, we are done. This then "forces" the adversary to set $Y_1 = Y_1'$, which implies that the same index of the codeword is being sampled both for the computation of $\mathsf{AdvGen}(X, Y)$ and for the computation of $\mathsf{AdvGen}(X, Y')$. As these two codewords are distinct (recall that $Y \neq Y'$) and since $\mathsf{Ext}(X, Y_1)$ is ε-close to uniform, the distance of the code guarantees that the suffix of $\mathsf{AdvGen}(X, Y)$ will be different from the respective suffix of $\mathsf{AdvGen}(X, Y')$ with probability $1 - O(\varepsilon)$. This advice generator can be shown to be nice using the assumption $|y_1| \leq 100 \cdot |y|$.

As y_1 is being used as a seed for Ext, its length must be taken to be $\Omega(\log(n/\varepsilon))$. This is the dominating part of the advice length as the suffix can be set to have length $O(\log(1/\varepsilon))$. Recall that the correlation breaker with advice being used requires $\ell = O(a \cdot \log(aw/\varepsilon))$. As $a = \Omega(\log(n/\varepsilon))$ and since $w = n$, the total seed length required for the non-malleable extractor is

$$\ell = O\left(a \cdot \log\left(\frac{aw}{\varepsilon}\right)\right) = O\left(\log^2\left(\frac{n}{\varepsilon}\right)\right).$$

The min-entropy requirement can also be shown to be $k = \Omega(\log^2(n/\varepsilon))$.

2.3 Switching the source and seed

Why did we pay $\log^2(n/\varepsilon)$ in the seed length? Well, one factor of $\log(n/\varepsilon)$ is due to the advice length a while the other is due to the fact that the source fed to AdvCB is X which has length $w = n$. In [Coh15b] it was shown how to save on the second factor by passing as a source to AdvCB not the original source X but rather a much shorter source. In fact, that alternative source for AdvCB is the original *seed* Y. Of course, though, one also needs to supply AdvCB with a uniform string (which before was simply Y) that is independent of the source (which will now be Y). This string will be some function of both X, Y that can be made independent of Y by conditioning on a carefully chosen event. To describe this function we require another, very useful, primitive from the literature.

Raz [Raz05] gave a construction of a strong seeded extractor that is also guaranteed to work with "weak-seeds", namely, seeds that are not required to be uniform, and it suffices that they have sufficient amount of min-entropy. More formally, Raz constructed a function

$$\mathsf{Raz}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m,$$

with $d = O(\log(n/\varepsilon))$, such that for any (n, k)-source X, and for any independent $(d, 0.51d)$-source Y, $(\mathsf{Raz}(X, Y), Y) \approx_\varepsilon (U_m, Y)$ (see Theorem 3.7).

With Raz's extractor in hand, the improved reduction from non-malleable extractors to advice generators suggested by [Coh15b] is defined as follows. Split y to three substrings $y = y_1 \circ y_2 \circ y_3$ with lengths d_1, d_2, d_3, respectively. Again, we take $d_1 = O(\log(n/\varepsilon))$ to be sufficiently large as required by a seed for Ext. We further require $d_2 \geq 100 \cdot d_1$ and $d_3 \geq 100 \cdot d_2$. The improved reduction is then given by

$$\mathsf{nmExt}(x, y) = \mathsf{AdvCB}\left(y_3, \mathsf{Raz}(x, y_2), \mathsf{AdvGen}(x, y)\right).$$

With the notations set above, the analysis proceeds as follows. First, note that conditioned on $\mathsf{AdvGen}(X,Y)$ and on $\mathsf{AdvGen}(X,Y')$, each of the random variables X, Y, which remain independent, loses only $O(\log(n/\varepsilon))$ bits of min-entropy. In particular Y_2 has min-entropy rate larger than 0.51. Thus, as Y_3 is much longer than Y_2, , by conditioning on Y_2 and on Y_2' we have that:

- $\mathsf{Raz}(X, Y_2)$ is close to uniform.
- Y_3 has min-entropy, say, $d/2$.
- The random variables $\mathsf{Raz}(X, Y_2)$, $\mathsf{Raz}(X, Y_2')$ are deterministic functions of X, and thus are jointly independent of the joint distribution (Y_3, Y_3').

Thus, the application of AdvCB is indeed valid (note also that unlike the original reduction [CGL15], no further assumptions on the inner workings of AdvCB are made). What is the required seed length from the resulting non-malleable extractor when using this reduction? First, y_1 is a seed for Ext and y_2 is a seed for Raz. These, however, do not put restriction beyond $\Omega(\log(n/\varepsilon))$ on the seed length. The bottleneck is, again, due to AdvCB. More precisely, as y_3 plays the role of the source to AdvCB, one must have that the min-entropy of Y_3 conditioned on the fixings above, which is $d/2$, is asymptotically bounded below by $a \cdot \log(a\ell/\varepsilon)$, where

$$\ell = \Omega\left(a \cdot \log\left(\frac{ad_3}{\varepsilon}\right)\right) = \Omega\left(\log\left(\frac{n}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right)\right),$$

and so the seed length required by non-malleable extractor is

$$d = \Omega\left(\log\left(\frac{n}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right)\right).$$

Similarly, one can show that the resulting non-malleable extractor supports min-entropy that is equal to the expression in the above equation.

So far we presented the ideas that go into previous works [CGL15, Coh15b]. We now turn to describe the new ideas that allow us to obtain improved non-malleable extractors.

2.4 Improved advice generators via correlation breakers with advice

The "switch" that was described above allows one to reduce one factor of $\log(n/\varepsilon)$ in the seed length of the resulting non-malleable extractor to $\log(\log(n)/\varepsilon)$. Recall that the other factor of $\log(n/\varepsilon)$ in the seed length is due to the advice length. We stress that, computational aspects aside, one can generate an advice of length $\log(1/\varepsilon)$. In particular, one can potentially completely decouple the advice length from the input length n.

The main idea that enables us to obtain improved non-malleable extractors in this work is the construction of an improved advice generator. The key idea for doing so is to use a correlation breaker with advice also for the construction of the advice generator. Thus, two correlation breakers with advice will be used in the construction of the non-malleable extractor – a new one for generating the advice, and the other is for the original purpose of reducing the construction of non-malleable extractors to the construction of advice generators. We now elaborate.

Given a string $y \in \{0,1\}^d$, we partition $y = y_1 \circ \cdots \circ y_5$, with $|y_i| = d_i$. As before, $d_1 = O(\log(n/\varepsilon))$ is chosen sufficiently long so to be used as a seed for a strong seeded extractor. We require that $d_i \geq 100 \cdot d_{i-1}$ for $i = 2,3,4,5$. For a parameter $1 \leq b \leq d_1$ to be chosen later on, set $a_{\mathsf{in}} = d_1/b$ and further partition y_1 to b consecutive equal length substrings, or blocks, $y_1 = y_1^1 \circ \cdots y_1^b$. Note that each y_1^j has length a_{in}. As mentioned, our improved advice generator is based on a correlation breaker with advice

$$\mathsf{AdvCB}_{\mathsf{in}} \colon \{0,1\}^{d_3} \times \{0,1\}^{\ell_{\mathsf{in}}} \times \{0,1\}^{a_{\mathsf{in}}} \to \{0,1\}^{\log(1/\varepsilon)},$$

where

$$\ell_{\mathsf{in}} = \Omega\left(a_{\mathsf{in}} \cdot \log\left(\frac{a_{\mathsf{in}} \cdot d_3}{\varepsilon}\right)\right) = \Omega\left(\frac{1}{b} \cdot \log\left(\frac{n}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right)\right) \tag{2.1}$$

as required by the construction of the correlation breakers with advice that we use. We further make use of a second instantiation of Raz's extractor

$$\mathsf{Raz}_{\mathsf{in}} \colon \{0,1\}^n \times \{0,1\}^{d_2} \times \{0,1\}^{\ell_{\mathsf{in}}}.$$

With these building blocks, our advice generator is given by

$$\mathsf{AdvGen}(x,y) = \mathsf{ECC}(y)_{\mathsf{Ext}(x,y_1)} \circ \bigcirc_{j=1}^{b} \mathsf{AdvCB}_{\mathsf{in}}\left(y_3, \mathsf{Raz}_{\mathsf{in}}(x,y_2), y_1^j\right),$$

where the expression $\bigcirc_{j=1}^{b} s_j$ stands for the concatenation $s_1 \circ \cdots \circ s_j$. In words, instead of setting y_1 as a substring of the advice, as suggested by [CGL15], so to force the adversary to set $Y_1 = Y_1'$, we make use of the different blocks of y_1 as advices to $\mathsf{AdvCB}_{\mathsf{in}}$ applied with "switched" source y_3 and seed $\mathsf{Raz}_{\mathsf{in}}(x,y_2)$. The concatenation of the outputs of these applications of $\mathsf{AdvCB}_{\mathsf{in}}$, in turn, compose part of the advice instead of y_1. The non-malleable extractor is then given by

$$\mathsf{nmExt}(x,y) = \mathsf{AdvCB}_{\mathsf{out}}\left(y_5, \mathsf{Raz}_{\mathsf{out}}(x,y_4), \mathsf{AdvGen}(x,y)\right),$$

for some suitable instantiations of a correlation breaker with advice $\mathsf{AdvCB}_{\mathsf{out}}$ and Raz's extractor $\mathsf{Raz}_{\mathsf{out}}$.

Before delving into the analysis, we remark that the idea above allows us to control the advice length that is generated by AdvGen. Indeed, note that instead of advice length $O(\log(n/\varepsilon))$, now the output of AdvGen is of length $O(b \cdot \log(1/\varepsilon))$. Note that one cannot simply take $b = 1$, hoping to minimize the advice length that is generated by AdvGen, as the advice length passed "internally" to $\mathsf{AdvCB}_{\mathsf{in}}$ is $a_{\mathsf{in}} = d_1/b$, and so by setting $b = 1$ we would gain nothing. So we need to juggle between two advice lengths – the external $O(b \cdot \log(1/\varepsilon))$, that increases with b, and the internal $d_1/b = O(\log(n/\varepsilon)/b)$ which increases with b. Luckily, even by taking all other considerations into account, the best idea is indeed setting these two advices to be of essentially equal lengths by taking

$$b = \sqrt{\frac{\log n}{\log(1/\varepsilon)}}.$$

We now proceed with the analysis. As before, if $Y_1 = Y_1'$, we are done. Otherwise, there must exists some block number $g \in [b]$ such that $Y_1^g \neq (Y')_1^g$ (one should be slightly careful as g is a function of Y_1, Y_1', though we allow ourselves to be somewhat imprecise in this section). Thus, the g'th application of $\mathsf{AdvCB}_{\mathsf{in}}$ when applied to Y is fed with an advice that is different from the advice that is passed to the corresponding application of $\mathsf{AdvCB}_{\mathsf{in}}$ when applied to Y'. By applying similar arguments to those used above when we analyzed the "switch", one can show that

$$\mathsf{AdvCB}\left(Y_3, \mathsf{Raz}_{\mathsf{in}}(X, Y_2), Y_1^g\right) \approx_\varepsilon U_{\log(1/\varepsilon)}$$

even conditioned on $\mathsf{AdvCB}(Y_3', \mathsf{Raz}_{\mathsf{in}}(X, Y_2'), (Y')_1^g)$. Hence,

$$\mathbf{Pr}\left[\mathsf{AdvCB}\left(Y_3, \mathsf{Raz}_{\mathsf{in}}(X, Y_2), Y_1^g\right) = \mathsf{AdvCB}\left(Y_3', \mathsf{Raz}_{\mathsf{in}}(X, Y_2'), (Y')_1^g\right)\right] \leq 2\varepsilon.$$

As we choose $d_5 \gg d_4 \gg d_3$, one can show that even conditioned on the advices, the outer correlation breaker with advice is fed with suitable source Y_5 and a uniform string $\mathsf{Raz}_{\mathsf{out}}(X, Y_4)$.

Now that it has been established that AdvGen, as defined above, is indeed an advice generator which composes nicely with the outer correlation breaker with advice, we turn to analyze the parameters. As mentioned above, since $\mathsf{AdvCB}_{\mathsf{in}}$ has output length $\log(1/\varepsilon)$ and so does the prefix of AdvGen that computes the error correcting code, the advice length generated by AdvGen is $a_{\mathsf{out}} = O(b \cdot \log(1/\varepsilon))$. Thus, the seed length that is required for the reduction from non-malleable extractors to advice generators, using our advice generator, is of order

$$a_{\mathsf{out}} \cdot \log\left(\frac{a_{\mathsf{out}} \cdot \ell_{\mathsf{out}}}{\varepsilon}\right) = b \cdot \log\left(\frac{1}{\varepsilon}\right) \cdot \log\left(\frac{b \cdot \log\log n}{\varepsilon}\right), \tag{2.2}$$

where in the above equation we took ℓ_{out} to be of order

$$a_{\mathsf{out}} \cdot \log\left(\frac{a_{\mathsf{out}} \cdot d_5}{\varepsilon}\right) = b \cdot \log\left(\frac{1}{\varepsilon}\right) \cdot \log\left(\frac{b \cdot \log n}{\varepsilon}\right),$$

as required by the construction we use of correlation breakers with advice.

Equation (2.1) and Equation (2.2) are two constraints on the seed length of the resulted non-malleable extractor. While Equation (2.1) decrease as a function of b, Equation (2.2) increases with b. As it turns out, the best choice for b is given by

$$b = \max\left(1, \sqrt{\frac{\log n}{\log(1/\varepsilon)}}\right).$$

By taking into account the constraint $d = \Omega(\log(n/\varepsilon))$ that follows as we use d_1 as a seed for Ext, one can get away with a non-malleable extractor with seed length $d = O(\log n + \log^3(1/\varepsilon))$.

3 Preliminaries

In this section we recall some standard definition and notations, and state results from the literature that we make use of.

Setting some standard notations. Unless stated otherwise, the logarithm in this paper is always taken base 2. For every natural number $n \geq 1$, define $[n] = \{1, 2, \ldots, n\}$. Throughout the paper, whenever possible, we avoid the use of floor and ceiling in order not to make the equations cumbersome. Whenever we say that a function is efficiently-computable we mean that the corresponding family of functions can be computed by a (uniform) algorithm that runs in polynomial-time in the input length.

Random variables and distributions. We sometimes abuse notation and syntactically treat random variables and their distribution as equal, specifically, we denote by U_m a random variable that is uniformly distributed over $\{0,1\}^m$. Furthermore, if U_m appears in a joint distribution (U_m, X) then U_m is independent of X. When m is clear from context, we omit it from the subscript and write U. The support of a random variable X is denoted by $\mathsf{supp}(X)$.

Statistical distance. The *statistical distance* between two distributions X, Y on the same domain D is defined by

$$\mathsf{SD}\left(X, Y\right) = \max_{A \subseteq D} \left| \, \mathbf{Pr}[X \in A] - \mathbf{Pr}[Y \in A] \, \right|.$$

If $\mathsf{SD}(X, Y) \leq \varepsilon$ we write $X \approx_{\varepsilon} Y$ and say that X and Y are ε-close.

Min-entropy [CG88]. The *min-entropy* of a random variable X, denoted by $H_\infty(X)$, is defined by

$$H_\infty(X) = \min_{x \in \mathsf{supp}(X)} \log_2 \left(\frac{1}{\mathbf{Pr}[X = x]} \right).$$

If X is supported on n-bit strings, we define the *min-entropy rate* of X by $H_\infty(X)/n$. In such case, if X has min-entropy k or more, we say that X is an (n, k)-source. When wish to refer to an (n, k)-source without specifying the quantitative parameters, we sometimes use the standard terms *source* or *weak-source*.

We make further use of a useful generalization of the notion of min-entropy.

Average conditional min-entropy. Let X, W be two random variables. The *average conditional min-entropy* of X given W is defined as

$$\widetilde{H}_\infty(X \mid W) = -\log_2 \left(\mathop{\mathbf{E}}_{w \sim W} \left[2^{-H_\infty(X \mid W = w)} \right] \right).$$

We make frequent use of the following two lemmas.

Lemma 3.1 ([DORS08]). *Let X, Y, Z be random variables such that Y has support size at most 2^ℓ. Then,*

$$\widetilde{H}_\infty(X \mid (Y,Z)) \geq \widetilde{H}_\infty((X,Y) \mid Z) - \ell \geq \widetilde{H}_\infty(X \mid Z) - \ell.$$

In particular, $\widetilde{H}_\infty(X \mid Y) \geq H_\infty(X) - \ell$.

Lemma 3.2 ([DORS08]). *For any two random variables X, Y and any $\varepsilon > 0$, it holds that*

$$\Pr_{y \sim Y}\left[H_\infty(X \mid Y = y) < \widetilde{H}_\infty(X \mid Y) - \log(1/\varepsilon)\right] \leq \varepsilon.$$

Extractors. For our construction, we make use of seeded extractors. We recall the definition of seeded extractors, some standard facts, and relevant results from the literature. For more information, we refer the interested reader to [Sha11, Vad11].

Definition 3.3 (Seeded extractors [NZ96]). *A function $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a (k,ε)-*seeded extractor *if for any (n,k)-source X it holds that $\mathsf{Ext}(X,S) \approx_\varepsilon U_m$, where S is uniformly distributed over $\{0,1\}^d$ independently of X. We say that Ext is a* strong seeded-extractor *if*

$$(\mathsf{Ext}(X,S), S) \approx_\varepsilon U_{m+d}.$$

We refer to k as the supported min-entropy of Ext and to ε as the error guarantee.

Throughout the paper we make use of the strong seeded extractor of Guruswami *et al.* [GUV09].

Theorem 3.4 ([GUV09]). *There exists a universal constant $c_{\mathsf{GUV}} \geq 1$ such that the following holds. For all positive integers n, k, and for any $\varepsilon > 0$, there exists an efficiently-computable (k,ε)-strong seeded-extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ having seed length $d = c_{\mathsf{GUV}} \cdot \log(n/\varepsilon)$ and $m = k/2$ output bits.*

Definition 3.5. *Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-strong seeded extractor. For an (n,k)-source X, we define the set*

$$\mathsf{BadSeeds}(X) = \{y \in \{0,1\}^d \;\mid\; \mathsf{SD}(\mathsf{Ext}(X,y), U) > \sqrt{\varepsilon}\}.$$

An element $y \in \mathsf{BadSeeds}(X)$ is called a bad seed for X (with respect to Ext)*. Otherwise y is called a* good seed for X*.*

The following useful and simple fact readily follows by Markov's inequality.

Fact 3.6. *Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-strong seeded extractor. Then, for any (n,k)-source X, $|\mathsf{BadSeeds}(X)| \leq \sqrt{\varepsilon} \cdot 2^d$.*

We also make frequent use of the following extractor with weak-seeds.

Theorem 3.7 ([Raz05])**.** *There exist universal constants $c_{\mathsf{Raz}}, c'_{\mathsf{Raz}} \geq 1$ such that the following holds. For all integers n, k, d and for any $\varepsilon > 0$ such that $d \geq c_{\mathsf{Raz}} \cdot \log(n/\varepsilon)$ and $k \geq c'_{\mathsf{Raz}} \cdot d$, there exists an efficiently-computable function*

$$\mathsf{Raz}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k/2}$$

with the following property. Let X be an (n,k)-source, and let Y be an independent $(d, 0.51d)$-source. Then, $(\mathsf{Raz}(X,Y), Y) \approx_\varepsilon (U, Y)$.

We also make use of error correcting codes. In particular, of algebraic-geometric codes. We first recall the definition of an error correcting code.

Definition 3.8. *Let Σ be some set. A mapping $\mathsf{ECC}\colon \Sigma^k \to \Sigma^n$ is called an error correcting code with relative-distance δ if for any $x, y \in \Sigma^k$, it holds that the Hamming distance between $\mathsf{ECC}(x)$ and $\mathsf{ECC}(y)$ is at least δn. The rate of the code, denoted by ρ, is defined by $\rho = k/n$. The set Σ is called the* alphabet *of the code.*

Theorem 3.9 ([GS95] (see also [Sti09]))**.** *Let p be any prime number and let m be an even integer. Set $q = p^m$. For every $\rho \in [0,1]$ and for any large enough integer n, there exists an efficiently-computable rate ρ linear error correcting code $\mathsf{ECC}\colon \mathbb{F}_q^{\rho n} \to \mathbb{F}_q^n$ with relative distance δ such that*

$$\rho + \delta \geq 1 - \frac{1}{\sqrt{q} - 1}.$$

Lastly, we give the definition of correlation breakers with advice, and the construction from the literature that we make use of.

Definition 3.10. *A (k, ε)-*correlation-breaker with advice *is a function*

$$\mathsf{AdvCB}\colon \{0,1\}^w \times \{0,1\}^\ell \times \{0,1\}^a \to \{0,1\}^m$$

with the following property. Let X, X' be random variables distributed over $\{0,1\}^w$ such that X has min-entropy k. Let Y, Y' be random variables over $\{0,1\}^\ell$ that are jointly independent of (X, X') such that Y is uniform. Then, for any a-bit strings $\alpha \neq \alpha'$ it holds that

$$(\mathsf{AdvCB}(X, Y, \alpha), \mathsf{AdvCB}(X', Y', \alpha')) \approx_\varepsilon (U_m, \mathsf{AdvCB}(X', Y', \alpha')).$$

The third argument to the function AdvCB is called the advice.

Theorem 3.11 ([CGL15])**.** *There exists a universal constant $c_{\mathsf{ACB}} \geq 1$ such that the following holds. For all integers ℓ, w, a and for any $\varepsilon > 0$ such that*

$$\ell \geq c_{\mathsf{ACB}} \cdot a \cdot \log\left(\frac{aw}{\varepsilon}\right), \tag{3.1}$$

there exists a poly(ℓ, w)*-time computable (k, ε)-correlation-breaker with advice*

$$\mathsf{AdvCB}\colon \{0,1\}^w \times \{0,1\}^\ell \times \{0,1\}^a \to \{0,1\}^{\ell/(2a)}$$

for

$$k \geq c_{\mathsf{ACB}} \cdot a \cdot \log\left(\frac{a\ell}{\varepsilon}\right). \tag{3.2}$$

4 Proof of Theorem 1.2 and Theorem 1.3

In this section we prove Theorem 1.2 and Theorem 1.3. We start with Theorem 1.2. In fact, we prove a somewhat more general result that is given by Lemma 4.1 below.

Lemma 4.1. *For any integer n, any $\varepsilon > 0$, and for any integer $b < \log(n/\varepsilon)$, there exists an efficiently-computable (k, ε)-non-malleable extractor*

$$\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{\log(1/\varepsilon)}$$

for

$$k = \Omega\left(\log\left(\frac{n}{\varepsilon}\right) + \frac{1}{b} \cdot \log\left(\frac{n}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right) + b \cdot \log\left(\frac{1}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right)\right),$$

with seed length

$$d = O\left(\log\left(\frac{n}{\varepsilon}\right) + \frac{1}{b} \cdot \log\left(\frac{n}{\varepsilon}\right) \cdot \log\left(\frac{\log n}{\varepsilon}\right) + b \cdot \log\left(\frac{1}{\varepsilon}\right) \cdot \log\left(\frac{\log\log n}{\varepsilon}\right)\right).$$

Proof. Let $c_{\mathsf{GUV}}, c_{\mathsf{Raz}}$ be the constants that appear in the statement of Theorem 3.4 and Theorem 3.7, respectively. Set $d_1 = \max(c_{\mathsf{GUV}}, c_{\mathsf{Raz}}) \cdot \log(n/\varepsilon)$ and let d_2, d_3, d_4, d_5 be integers such that $d = d_1 + \cdots + d_5$. We assume that $d_i \geq 20 \cdot d_{i-1}$. Note that by our assumption on d, such a sequence of d_i's exists. Given $y \in \{0,1\}^d$, we partition $y = y_1 \circ y_2 \circ \cdots \circ y_5$ such that $|y_i| = d_i$ for $i = 1, \ldots, 5$.

Building blocks. For the construction of our non-malleable extractor we make use of several building blocks from the literature. We now present these components while setting and defining relevant parameters.

- Let q be the least even prime power of 2 that is larger or equal than $5/\varepsilon^2$. Note that $q \leq 20/\varepsilon^2$. Let r be the least integer such that $q^r \geq d$. We identify $[d]$ with an arbitrary subset of $\mathbb{F}_q^r$. Set $v = 2r/\varepsilon$ and let $\mathsf{ECC}\colon \mathbb{F}_q^r \to \mathbb{F}_q^v$ be the error correcting code that is given by Theorem 3.9, set with relative distance $\delta = 1 - \varepsilon$. By Theorem 3.9, an explicit code with these parameters (namely, relative distance $1 - \varepsilon$, rate $2/\varepsilon$, and alphabet size $q \leq 20/\varepsilon^2$) exists.

- Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{\log v}$ be the strong $(2\log v, \varepsilon)$-seeded extractor that is given by Theorem 3.4. Note that d_1 was defined to be of sufficient length so to be used as a seed for Ext. We identify the output of Ext as an element of $[v]$.

- Set $a_{\mathsf{in}} = d_1/b$. Let c_{ACB} be the constant that appears in the statement of Theorem 3.11, and set
$$\ell_{\mathsf{in}} = 2c_{\mathsf{ACB}} \cdot a_{\mathsf{in}} \cdot \log\left(\frac{a_{\mathsf{in}} \cdot d_3}{\varepsilon}\right).$$
Let $\mathsf{Raz}_{\mathsf{in}}\colon \{0,1\}^n \times \{0,1\}^{d_2} \times \{0,1\}^{\ell_{\mathsf{in}}}$ be the $(2\ell_{\mathsf{in}}, \varepsilon)$ extractor with weak seeds that is given by Theorem 3.7. As $d_2 \geq d_1 \geq c_{\mathsf{Raz}} \cdot \log(n/\varepsilon)$, a seed of length d_2 suffices for $\mathsf{Raz}_{\mathsf{in}}$.

- Let

$$\mathsf{AdvCB}_{\mathsf{in}} \colon \{0,1\}^{d_3} \times \{0,1\}^{\ell_{\mathsf{in}}} \times \{0,1\}^{a_{\mathsf{in}}} \to \{0,1\}^{\log(1/\varepsilon)}$$

be the correlation breaker with advice that is given by Theorem 3.11. Note that ℓ_{in} was chosen so to meet the hypothesis of Theorem 3.11. Further, by Theorem 3.11, the output length of $\mathsf{AdvCB}_{\mathsf{in}}$ is $\ell_{\mathsf{in}}/(2a_{\mathsf{in}})$, and is therefore bounded below by $\log(1/\varepsilon)$. Thus, we may truncate the output of the function given by Theorem 3.11 so to have $\log(1/\varepsilon)$ output bits as appears in the definition of $\mathsf{AdvCB}_{\mathsf{in}}$ above.

- Set $a_{\mathsf{out}} = (b+2) \cdot \log(1/\varepsilon) + 5$, and let

$$\ell_{\mathsf{out}} = 2c_{\mathsf{ACB}} \cdot a_{\mathsf{out}} \cdot \log\left(\frac{a_{\mathsf{out}} \cdot d_5}{\varepsilon}\right).$$

Let $\mathsf{Raz}_{\mathsf{out}} \colon \{0,1\}^n \times \{0,1\}^{d_4} \to \{0,1\}^{\ell_{\mathsf{out}}}$ be the $(2\ell_{\mathsf{out}}, \varepsilon)$ extractor with weak seeds that is given by Theorem 3.7. As $d_4 \ge d_1 \ge c_{\mathsf{Raz}} \cdot \log(n/\varepsilon)$, a seed of length d_4 suffices for $\mathsf{Raz}_{\mathsf{in}}$.

- Finally, let

$$\mathsf{AdvCB}_{\mathsf{out}} \colon \{0,1\}^{d_5} \times \{0,1\}^{\ell_{\mathsf{out}}} \times \{0,1\}^{a_{\mathsf{out}}} \to \{0,1\}^{\log(1/\varepsilon)}$$

be the correlation breaker with advice that is given by Theorem 3.11. Note that ℓ_{out} was chosen so to meet the hypothesis of Theorem 3.11. As with $\mathsf{AdvCB}_{\mathsf{in}}$, one can set the output length, as we set above to $\log(1/\varepsilon)$ as $\ell_{\mathsf{out}}/(2a_{\mathsf{out}}) \ge \log(1/\varepsilon)$.

The construction. With the building blocks introduced above, we are now ready to define our non-malleable extractor. We start by defining the function

$$\mathsf{AdvGen} \colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{a_{\mathsf{out}}},$$

which we prove to be an advice generator, as follows. Let $y \in \{0,1\}^d$ and recall our notation $y = y_1 \circ \cdots \circ y_5$, with $|y_i| = d_i$. We further partition y_1 to b consecutive substrings, or blocks, $y_1 = y_1^1 \circ \cdots \circ y_1^b$, each of length $d_1/b = a_{\mathsf{in}}$. Define

$$\mathsf{AdvGen}(x,y) = \mathsf{ECC}(y)_{\mathsf{Ext}(x,y_1)} \circ \bigcirc_{j=1}^{b} \mathsf{AdvCB}_{\mathsf{in}}\left(y_3, \mathsf{Raz}_{\mathsf{in}}(x,y_2), y_1^j\right),$$

where by $\mathsf{ECC}(y)_{\mathsf{Ext}(x,y_1)}$ we mean the following – we interpret the $(\log v)$-bit string $\mathsf{Ext}(x,y_1)$ as an element $i \in [v]$. Then, $\mathsf{ECC}(y)_i$ the content of the i'th entry of the codeword $\mathsf{ECC}(y)$. Further, for strings $s_1, \ldots, s_j$, the expression $\bigcirc_{j=1}^{b} s_j$ stands for the concatenation $s_1 \circ \cdots \circ s_j$. Finally, define

$$\mathsf{nmExt}(x,y) = \mathsf{AdvCB}_{\mathsf{out}}(y_5, \mathsf{Raz}_{\mathsf{out}}(x,y_4), \mathsf{AdvGen}(x,y)).$$

Note that the third argument, $\mathsf{AdvGen}(x,y)$, consists of $\log q + b\log(1/\varepsilon)$ bits. Indeed, the error correcting code ECC is over an alphabet of size q, which can be identified with $\{0,1\}^{\log q}$. Further, the output length of $\mathsf{AdvCB}_{\mathsf{in}}$ was set to $\log(1/\varepsilon)$. As $q \le 20/\varepsilon^2$, the third argument consists of at most $(b+2)\log(1/\varepsilon) + 5 = a_{\mathsf{out}}$ bits, as required by $\mathsf{AdvCB}_{\mathsf{out}}$.

Analysis. With the construction in hand, we turn to the analysis. Let $\mathcal{A}\colon \{0,1\}^d \to \{0,1\}^d$ be a function with no fixed points, and let X an (n,k)-source. Let Y be a random variable that is uniformly distributed over d-bit strings, independently of X. It will be convenient to denote $Y' = \mathcal{A}(Y)$. We start by proving the following claim which, informally speaking, states that AdvGen as defined above is an advice generator.

Claim 4.2.

$$\mathop{\mathbf{Pr}}_{(x,y)\sim(X,Y)}\left[\mathsf{AdvGen}(x,y) = \mathsf{AdvGen}(x,\mathcal{A}(y))\right] = O(\sqrt{\varepsilon}).$$

Proof of Claim 4.2. For $(y_1, y_1') \in \mathsf{supp}(Y_1, Y_1')$ consider the event $(Y_1, Y_1') = (y_1, y_1')$. By Fact 3.6, except with probability $\sqrt{\varepsilon}$ over these fixings, y_1 is a good seed for X with respect to Ext. We proceed by considering two cases.

Case 1 – $y_1 = y_1'$. In this case we follow the analysis of Chattopadhyay *et al.* [CGL15]. Recall that $Y \neq Y'$ and so, as ECC has relative distance $\delta = 1 - \varepsilon$, the codewords $\mathsf{ECC}(Y), \mathsf{ECC}(Y') \in \mathbb{F}_q^v$ agree on at most ε fraction of the coordinates. Let

$$A = \{i \in [v] \mid \mathsf{ECC}(Y)_i = \mathsf{ECC}(Y')_i\}$$

be the random variable that consists of all indices on which the two codewords agree. With probability 1, $|A| \le \varepsilon v$. As $\mathsf{Ext}(X, y_1)$ is $\sqrt{\varepsilon}$-close to uniform for y_1 that is a good seed for X, and since $\mathsf{Ext}(X, y_1)$ is independent of A (as A is a deterministic function of Y), we have that whenever y_1 is a good seed for X,

$$\mathop{\mathbf{Pr}}_{X,Y}\left[\mathsf{Ext}(X, y_1) \in A\right] \le \varepsilon.$$

We can now conclude the proof of the claim for this case by taking back into account the event that y_1 is not a good seed for X.

Case 2 – $y_1 \neq y_1'$. First, note that in this case there exists some $g = g(y_1, y_1') \in [b]$ such that $y_1^g \neq (y')_1^g$. Now, by Lemma 3.1,

$$\widetilde{H}_\infty(Y_2 \mid Y_1, Y_1') \ge d_2 - 2d_1 \ge 0.9 d_2,$$

and so by Lemma 3.2, except with probability ε over the fixings of $(y_1, y_1') \sim (Y_1, Y_1')$, it holds that

$$H_\infty(Y_2) \ge 0.9 d_2 - \log(1/\varepsilon) \ge 0.51 d_2.$$

Thus, except with probability ε over the fixings of Y_1, Y_1', the random variable Y_2 has min-entropy rate 0.51. Further, Y_2 remains independent of X conditioned on these fixings. Thus, by Theorem 3.7, which is applicable as $k = H_\infty(X) \ge 2\ell_{\mathsf{in}}$ and $k \ge c'_{\mathsf{Raz}} d_2$, the random variable $\mathsf{Raz}_{\mathsf{in}}(X, Y_2)$ is ε-close to uniform conditioned on the further fixing of Y_2. Thus, by Markov's inequality, except with probability $O(\sqrt{\varepsilon})$ over $(y_1, y_1', y_2) \sim (Y_1, Y_1', Y_2)$, it holds that $\mathsf{Raz}_{\mathsf{in}}(X, y_2)$ is $O(\sqrt{\varepsilon})$-close to uniform.

At this point we further condition on the fixing of $Y_2' = y_2'$ for $y_2' \sim Y_2'$. By Lemma 3.1,

$$\widetilde{H}_\infty \left(Y_3 \mid Y_1, Y_1', Y_2, Y_2'\right) \geq d_3 - 2(d_1 + d_2) \geq 0.8 d_3.$$

Thus, by Lemma 3.2, except with probability ε over $(y_1, y_1', y_2, y_2') \sim (Y_1, Y_1', Y_2, Y_2')$, it holds that $H_\infty(Y_3) \geq d_3/2$. To summarize, except with probability $O(\sqrt{\varepsilon})$ over the fixings of Y_1, Y_1', Y_2, Y_2' we have that:

- $\mathsf{Raz_{in}}(X, Y_2)$ is $O(\sqrt{\varepsilon})$-close to uniform.
- $H_\infty(Y_3) \geq d_3/2$.
- The joint distribution of the random variables $\mathsf{Raz_{in}}(X, Y_2)$, $\mathsf{Raz_{in}}(X, Y_2')$ is independent of the joint distribution (Y_3, Y_3').

By the above, together with the fact that $y_1^g \neq (y')_1^g$, we can apply Theorem 3.11 to conclude that,

$$\begin{aligned}&\left(\mathsf{AdvCB_{in}}(Y_3, \mathsf{Raz_{in}}(X, y_2), y_1^g), \mathsf{AdvCB_{in}}(Y_3', \mathsf{Raz_{in}}(X, y_2'), (y')_1^g)\right) \approx_{O(\sqrt{\varepsilon})} \\ &\left(U, \mathsf{AdvCB_{in}}(Y_3', \mathsf{Raz_{in}}(X, y_2'), (y')_1^g)\right). \end{aligned} \tag{4.1}$$

We remark that this application of Theorem 3.11 is valid as one can easily verify that Equation (3.1) and Equation (3.2) in the hypothesis of Theorem 3.11 holds. By Equation (4.1), and since $\mathsf{AdvCB_{in}}$ has output length $\log(1/\varepsilon)$, except with probability $O(\sqrt{\varepsilon})$ over the fixings done so far,

$$\mathsf{AdvCB_{in}}(Y_3, \mathsf{Raz_{in}}(X, y_2), y_1^g) \neq \mathsf{AdvCB_{in}}(Y_3', \mathsf{Raz_{in}}(X, y_2'), (y')_1^g),$$

which proves the claim as $\mathsf{AdvCB_{in}}(y_3, \mathsf{Raz_{in}}(x, y_2), y_1^g)$ is a substring of $\mathsf{AdvGen}(x, y)$. □

We proceed to prove the following claim.

Claim 4.3. *Conditioned on any fixing of* $\mathsf{AdvGen}(X, Y), \mathsf{AdvGen}(X, Y')$, *the random variables* X, Y *remain independent. Moreover, except with probability* ε *over the fixing of the variables* $\mathsf{AdvGen}(X, Y)$, $\mathsf{AdvGen}(X', Y)$, *each of* Y_4, Y_5 *has min-entropy rate* 0.6, *and* X *has min-entropy* $k/2$.

Proof of Claim 4.3. We note that by fixing Y_1, Y_2, Y_3, Y_1', Y_2', Y_3' to y_1, y_2, y_3, y_1', y_2', y_3', respectively, the random variables $\mathsf{AdvGen}(X, Y)$, $\mathsf{AdvGen}(X, Y')$ have the form

$$\begin{aligned}\mathsf{AdvGen}(X, Y) &= \mathsf{ECC}(Y)_{\mathsf{Ext}(X, y_1)} \circ \bigcirc_{j=1}^{b} \mathsf{AdvCB_{in}}(y_3, \mathsf{Raz_{in}}(X, y_2), y_1^j), \\ \mathsf{AdvGen}(X, Y') &= \mathsf{ECC}(Y')_{\mathsf{Ext}(X, y_1')} \circ \bigcirc_{j=1}^{b} \mathsf{AdvCB_{in}}(y_3', \mathsf{Raz_{in}}(X, y_2'), (y')_1^j).\end{aligned}$$

We proceed by conditioning on the further fixings of

$$\mathsf{Ext}(X, y_1), \mathsf{Ext}(X, y_1'), \mathsf{Raz_{in}}(X, y_2), \mathsf{Raz_{in}}(X, y_2').$$

Note that these random variables are deterministic functions of X, and so conditioning on them does not introduce any dependencies between X and Y. Conditioned on the fixings done so far, the only non fixed part of $\mathsf{AdvGen}(X,Y)$, $\mathsf{AdvGen}(X,Y')$ are the prefixes $\mathsf{ECC}(Y)_{\mathsf{Ext}(X,y_1)}$ and $\mathsf{ECC}(Y')_{\mathsf{Ext}(X,y_1')}$, which at this point are deterministic functions of Y. Hence, one can further condition on the fixings of $\mathsf{ECC}(Y)_{\mathsf{Ext}(X,y_1)}$, $\mathsf{ECC}(Y')_{\mathsf{Ext}(X,y_1')}$ without introducing dependencies between X, Y, as desired.

As for the moreover part of the claim, note that

$$|Y_1 \circ Y_2 \circ Y_3| + |Y_1' \circ Y_2' \circ Y_3'| + |\mathsf{ECC}(Y)_{\mathsf{Ext}(X,Y_1)}| + |\mathsf{ECC}(Y')_{\mathsf{Ext}(X,Y_1')}| \le 0.3 d_4.$$

Thus, by Lemma 3.1,

$$\widetilde{H}_\infty(Y_4 \mid \mathsf{AdvGen}(X,Y), \mathsf{AdvGen}(X,Y')) \ge 0.7 d_4,$$
$$\widetilde{H}_\infty(Y_5 \mid \mathsf{AdvGen}(X,Y), \mathsf{AdvGen}(X,Y')) \ge 0.7 d_5.$$

Similarly, as the output length of $\mathsf{Raz}_{\mathsf{in}}, \mathsf{Ext}$ is set to ℓ_{in} and $\log v$, respectively, it holds that

$$\widetilde{H}_\infty(X \mid \mathsf{AdvGen}(X,Y), \mathsf{AdvGen}(X,Y')) \ge k - (\ell_{\mathsf{in}} + \log v) \ge 0.6k.$$

The proof then follows by Lemma 3.2 and since d_4, k are a large enough multiple of $\log(1/\varepsilon)$. □

We proceed with the proof of Lemma 4.1. By Claim 4.2 and Claim 4.3, we have that except with probability $O(\sqrt{\varepsilon})$ over $(\alpha, \alpha') \sim (\mathsf{AdvGen}(X,Y), \mathsf{AdvGen}(X,Y'))$ it holds that:

- $\alpha \ne \alpha'$.
- X, Y remain independent.
- $H_\infty(X) \ge k/2$.
- Each of Y_4, Y_5 has min-entropy rate 0.6.

By Theorem 3.7, and since $k/2 \ge 2\ell_{\mathsf{out}}$ and $k/2 \ge c'_{\mathsf{Raz}} d_4$, the random variable $\mathsf{Raz}_{\mathsf{out}}(X, Y_4)$ is $O(\sqrt{\varepsilon})$-close to uniform conditioned on the further fixing of Y_4. Thus, by Markov's inequality, except with probability $O(\sqrt{\varepsilon})$ conditioned on the fixings done so far, it holds that $\mathsf{Raz}_{\mathsf{out}}(X, y_4)$ is $O(\sqrt{\varepsilon})$-close to uniform. We now further condition on the fixing of Y_4'. By Lemma 3.1 and Lemma 3.2, except with probability $O(\sqrt{\varepsilon})$ over the fixings done so far

$$H_\infty(Y_5) \ge 0.6 d_5 - 2 d_4 - \log(1/\varepsilon) \ge d_5/3.$$

To summarize, except with probability $O(\sqrt{\varepsilon})$ over the fixings done so far, we have that:

- $\mathsf{Raz}_{\mathsf{out}}(X, Y_4)$ is $O(\sqrt{\varepsilon})$-close to uniform.
- Y_5 has min-entropy rate at least 1/3.

- The joint distribution of the random variables $\mathsf{Raz}_{\mathsf{out}}(X, Y_4)$, $\mathsf{Raz}_{\mathsf{out}}(X, Y_4')$ is independent of the joint distribution (Y_5, Y_5').

As $\alpha \neq \alpha'$, we can apply Theorem 3.11, whose hypothesis is met by our setting of parameters, to conclude that,

$$(\mathsf{nmExt}(X,Y), \mathsf{nmExt}(X,Y')) \approx_{O(\sqrt{\varepsilon})} (U, \mathsf{nmExt}(X,Y')) \,.$$

This concludes the proof but for the error guarantee which is $O(\sqrt{\varepsilon})$ rather than the stated ε. Clearly, however, one can obtain error ε without affecting the statement of the lemma simply by using building blocks with error $\alpha \cdot \varepsilon^2$ rather than ε for some small enough constant $0 < \alpha < 1$. □

With Lemma 4.1 in hand, we are now ready to prove Theorem 1.2.

Proof of Theorem 1.2. Apply Lemma 4.1 with

$$b = \max\left(1, \sqrt{\frac{\log n}{\log(1/\varepsilon)}}\right).$$

It is easy to see that for $\varepsilon < 1/n$, $b = 1$ and the resulting seed length, and supported min-entropy, are of order $\log^2(1/\varepsilon)$. We turn to consider the case $\varepsilon > 1/n$. By some simple calculations, one can verify that in this case

$$d = O\left(\log n + \sqrt{\log n \cdot \log^3(1/\varepsilon)} + \sqrt{\log n \cdot \log(1/\varepsilon)} \cdot \log\log n\right)$$

As

$$\sqrt{\log n \cdot \log(1/\varepsilon)} \cdot \log\log n \le \max\left(\log n, \sqrt{\log n \cdot \log^3(1/\varepsilon)}\right)$$

for large enough n, and since for every $x, y > 0$, $\sqrt{xy} \le x + y$, we have that

$$d = O\left(\log n + \sqrt{\log n \cdot \log^3(1/\varepsilon)}\right) = O\left(\log n + \log^3(1/\varepsilon)\right).$$

Further, one can easily verify that nmExt supports min-entropy $k = \Omega(d)$. □

To prove Theorem 1.3 we make use of the following result that, informally speaking, gives a black-box algorithm for increasing the output length of a non-malleable extractor.

Theorem 4.4 ([Coh15b]). *There exists a universal constant $\alpha > 0$ such that the following holds. Let*

$$\mathsf{nmExt}\colon \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{\log(1/\varepsilon)}$$

be an explicit (k, ε)-non-malleable extractor with

$$k = \Omega\left(\log n + \log(d_1/\varepsilon) \cdot \log(1/\varepsilon)\right).$$

Then, for any $m < \alpha k / \log(1/\varepsilon)$, there exists an explicit (k, ε')-non-malleable extractor

$$\mathsf{nmExt}' \colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with error guarantee $\varepsilon' = O(\varepsilon^{1/4})$, having seed length

$$d = O\left(d_1 + \log(m/\varepsilon) \cdot \log(1/\varepsilon)\right).$$

Proof of Theorem 1.3. We first apply Theorem 1.2 to obtain a non-malleable extractor nmExt' with seed length $d_1 = O(\log n + \log^3(1/\varepsilon))$. Note that although Theorem 1.2 only guarantees that nmExt' has one output bit, the proof of Theorem 1.2 above implies that nmExt' has $\log(1/\varepsilon)$ output bits. Therefore, one can apply Theorem 4.4 to nmExt' so to obtain a second non-malleable extractor $\mathsf{nmExt} \colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ that supports min-entropy k, has error guarantee $O(\varepsilon^{1/4})$, seed length

$$d = O\left(d_1 + \log(k/\varepsilon) \cdot \log(1/\varepsilon)\right),$$

and $m = \Omega(k/\log(1/\varepsilon))$ output bits. Clearly, one can reduce the error guarantee from $O(\varepsilon^{1/4})$ to ε without changing the statement of Theorem 1.3. □

References

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CGL15] E. Chattopadhyay, V. Goyal, and X. Li. Non-malleable extractors and codes, with their many tampered extensions. *arXiv preprint arXiv:1505.00107*, 2015.

[Coh15a] G. Cohen. Local correlation breakers and applications to three-source extractors and mergers. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 845–862. IEEE, 2015.

[Coh15b] G. Cohen. Non-malleable extractors – new tools and improved constructions. In *Electronic Colloquium on Computational Complexity (ECCC)*, page 183, 2015.

[CRS14] G. Cohen, R. Raz, and G. Segev. Nonmalleable extractors with short seeds and applications to privacy amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.

[CZ15] E. Chattopadhyay and D. Zuckerman. Explicit two-source extractors and resilient functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

[DLWZ14] Y. Dodis, X. Li, T. D. Wooley, and D. Zuckerman. Privacy amplification and non-malleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[DW09] Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the forty-first annual ACM Symposium on Theory of Computing*, pages 601–610. ACM, 2009.

[GS95] A. Garcia and H. Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Inventiones Mathematicae*, 121(1):211–222, 1995.

[GUV09] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20, 2009.

[Li12a] X. Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing*, pages 837–854, 2012.

[Li12b] X. Li. Non-malleable extractors, two-source extractors and privacy amplification. In *IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 688–697, 2012.

[Li13] X. Li. New independent source extractors with exponential improvement. In *Proceedings of the forty-fifth annual ACM Symposium on Theory of Computing*, pages 783–792. ACM, 2013.

[NZ96] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Raz05] R. Raz. Extractors with weak random seeds. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[Sha11] R. Shaltiel. An introduction to randomness extractors. In *Automata, languages and programming*, pages 21–41. Springer, 2011.

[Sti09] H. Stichtenoth. *Algebraic function fields and codes*, volume 254. Springer Science & Business Media, 2009.

[Vad11] S. P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 2011.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de