Dispersers for affine sources with sub-polynomial entropy

Ronen Shaltiel*
University of Haifa

October 3, 2011

Abstract

We construct an explicit disperser for affine sources over $\mathbb{F}_2^n$ with entropy $k = 2^{\log^{0.9} n} = n^{o(1)}$. This is a polynomial time computable function $D : \mathbb{F}_2^n \to \{0,1\}$ such that for every affine space V of $\mathbb{F}_2^n$ that has dimension at least k, $D(V) = \{0,1\}$. This improves the best previous construction of Ben-Sasson and Kopparty (STOC 2009) that achieved $k = \Omega(n^{4/5})$.

Our technique follows a high level approach that was developed in Barak, Kindler, Shaltiel, Sudakov and Wigderson (J. ACM 2010) and Barak, Rao, Shaltiel and Wigderson (STOC 2006) in the context of dispersers for two independent general sources. The main steps are:

- Adjust the high level approach to make it suitable for affine sources.
- Implement a "challenge-response game" for affine sources (in the spirit of the two aforementioned papers that introduced such games for two independent general sources).
- In order to implement the game, we construct extractors for affine block-wise sources. For this we use ideas and components by Rao (CCC 2009).
- Combining the three items above, we obtain dispersers for affine sources with entropy larger than $\sqrt{n}$. We use a recursive win-win analysis in the spirit of Reingold, Shaltiel and Wigderson (SICOMP 2006) and Barak, Rao, Shaltiel and Wigderson (STOC 2006) to get affine dispersers with entropy less than $\sqrt{n}$.

*This research was supported by BSF grant 2004329 and ISF grant 686/07.

1 Introduction

This paper continues an active and long line of research that attempts to construct explicit dispersers and extractors for various families of "imperfect random sources". These objects have found many applications in diverse fields of computer science and mathematics. The reader is referred to survey articles [Nis96, Sha02, Vad02, Vad07].

Definition 1.1 (dispersers and extractors)**.** *Let $\mathcal{C}$ be a family of distributions over some set Ω.*

- *$D : \Omega \to \{0,1\}^m$ is a* disperser *for $\mathcal{C}$ with error ϵ if for every $X \in \mathcal{C}$, $|D(\mathsf{Supp}(X))| \geq (1-\epsilon)2^m$ (where $\mathsf{Supp}(X)$ denotes the support of X). In this paper we will be mostly interested in zero-error dispersers for $m = 1$.*
- *$E : \Omega \to \{0,1\}^m$ is an* extractor *for $\mathcal{C}$ with error ϵ if for every $X \in \mathcal{C}$, $E(X)$ is ϵ-close to the uniform distribution. (where two distributions X, Y over the same set Ω are ϵ-close if for every event A, $|\Pr[X \in A] - \Pr[Y \in A]| \leq \epsilon$).*

Throughout the paper we use the term "source" and "distribution" interchangeably. Many families of sources are considered in the literature, and the goal of this research area is to design explicit (that is polynomial time computable) extractors and dispersers for interesting families. The reader is referred to the survey article [Sha11] and to the introductions of [Sha08, GS08] for an overview of the classes of sources considered in the literature.

Dispersers and extractors for affine sources. In this paper we consider affine sources. Let $\mathbb{F}_q$ be a finite field of size q, and set $\Omega = \mathbb{F}_q^n$. A dimension k affine space in $\mathbb{F}_q^n$ is a set $V = \left\{\sum_{1 \leq i \leq k} a_i x_i + x'\right\}$ where $x_1, \ldots, x_k \in \mathbb{F}_q^n$ are linearly independent vectors, $a_1, \ldots, a_k \in \mathbb{F}_q$ are scalars, and $x' \in \mathbb{F}_q^n$ is the "shift vector". An affine source X of dimension k is a distribution that is uniformly distributed over some affine space V of $\mathbb{F}_q^n$ with dimension k. Note that the dimension of an affine source X is also given by $H(X)/\log q$ where H is the Shannon entropy function.

Past work on dispersers and extractors for affine sources considers two different parameter regimes: For "large fields" with $q = \text{poly}(n)$, Gabizon and Raz, [GR08] constructed extractors that extract almost all the entropy from affine sources of dimension k for any $k \geq 1$. Gabizon and Shaltiel [GS08] extended this result to give zero-error dispersers with large output length for the same parameters. DeVos and Gabizon [DG10] give a tradeoff between the dimension k and the field size q in which decreasing q requires increasing k.

The other parameters regime is that of "small fields", and in the extreme, the field $\mathbb{F}_2$. We focus on this setting in the remainder of the paper and identify $\mathbb{F}_2$ with $\{0,1\}$. Note that in this setup the entropy and dimension of affine sources coincide. Constructing dispersers and extractors for affine sources over $\mathbb{F}_2$ turns out to be a challenging problem. Bourgain [Bou07] constructed extractors for affine sources with entropy $k = \delta n$ for every $\delta > 0$. Subsequently, Yehudayoff [Yeh10] and Li [Li11b] constructed extractors for affine sources of entropy $\Omega(n/\sqrt{\log\log n})$. Rao [Rao09b] constructed extractors that can handle much lower entropy ($k = \text{polylog}n$) but only for a restricted sub-family of affine sources called "low-weight sources". Li [Li11a] constructed extractors for two independent affine sources, where each one has entropy slightly larger than $\sqrt{n}$. The best known construction of dispersers for affine sources is by Ben-Sasson and Kopparty [BSK09] and can handle affine sources with entropy $\Omega(n^{4/5})$. We also mention that Ben-Sasson and Zewi [BSZ11] showed that extractors for affine sources can be used to construct 2-source extractors, and that Demenkov and Kulikov [DK11] showed that dispersers for affine source yield certain circuit lower bounds.

Our results. Note that all previous constructions of extractors and dispersers for general affine sources, need entropy at least $\sqrt{n}$. In this paper we construct dispersers for affine sources with entropy k=$n^{o(1)}$.

Theorem 1.2. *There is a polynomial time computable function* $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ *that is a zero-error disperser for affine sources of entropy* $k = 2^{\log^{0.9} n}$.

Theorem 1.2 is stated for extracting a single bit. Some of the aforementioned constructions [Bou07, Yeh10, Li11b] can extract many bits. We remark that using the technique of Gabizon and Shaltiel [GS08], any zero-error disperser for affine sources that extracts $2\log n$ bits, can be "pushed" to extract almost all the entropy with zero-error. It seems that our technique can be extended to show that for every $\delta > 0$ there is a constant $c_\delta > 0$ and an explicit disperser for $k = n^\delta$ which extracts $c_\delta \cdot \log\log n$ bits. However, at this point we do not know how to extract $2\log n$ bits.

1.1 Technique

In order to explain our approach, we need the following notion of affine subsources.

Definition 1.3. *Let X be an affine source. An affine source X' is an affine subsource of X with deficiency d if there exists a linear function L of rank at most d and a possible value for v for $L(X)$ such that X' is uniformly distributed over $\{x \in \mathsf{Supp}(X) : L(x) = v\}$.*

Our starting point is the following trivial observation:

Fact 1.4 (Succeeding on an affine subsource)**.** *Let X be an affine source and let X' be an affine subsource of X. If* Disp *is a disperser for X' then* Disp *is a disperser for X.*

Let $\mathcal{P}$ be some property of affine sources. A paradigm for constructing dispersers for affine sources is to show that:

- Every affine source X with sufficiently large entropy has an affine subsource with property $\mathcal{P}$.
- There are explicit dispersers for affine sources with property $\mathcal{P}$.

Let X be an affine source. We will divide $X \in \{0,1\}^n$ into t blocks $X_1, \ldots, X_t \in \{0,1\}^{n/t}$ and consider properties $\mathcal{P}$ of the form "there exist some block i such that X_i has large entropy" (or more generally that the source is nicely structured in the sense that the entropy distribution between different blocks satisfies some condition). For properties $\mathcal{P}$ as above, we can often show that any affine source X with sufficiently large entropy has an affine subsource $X^{(0)}$ that has property $\mathcal{P}$. Furthermore, it is often possible to construct dispersers (or even extractors) for affine sources that have property $\mathcal{P}$, at least assuming that *we know the index i*. This does not seem helpful as we get only one sample from the source and cannot hope to find i by looking at one sample.

We follow an approach of [BKS+10] and will try to design a procedure $\mathsf{FindBlock}(x)$ with the property that every affine source $X^{(0)}$ that has property $\mathcal{P}$, has a small deficiency affine subsource X' on which $\mathsf{FindBlock}(X')$ identifies index i correctly, with high probability (that is $\Pr[\mathsf{FindBlock}(X') = i] = 1 - o(1)$). Properties $\mathcal{P}$ that we will consider are preserved when conditioning into small deficiency affine subsources, and so X' also has property $\mathcal{P}$ with respect to the *same* index i.

Putting things together, let $\mathsf{Disp}(x,i)$ be a disperser for affine sources with property $\mathcal{P}$ (that needs to know the good index i). We have that $\mathsf{FindBlock}(X')$ correctly identifies the good index i with high probability, and so after computing $i \leftarrow \mathsf{FindBlock}(x)$ we can apply supply i to Disp and compute $\mathsf{Disp}(x,i)$.

We obtain a disperser for X', which by Fact 1.4 implies is also a disperser for the initial general affine source X.

In order to design procedure FindBlock, we follow the high level approach of [BKS+10, BRSW06] and design a "challenge-response game" for affine sources. On the one hand, we have an advantage over [BKS+10, BRSW06] as affine sources are easier to work with than general sources: We can work with Shannon entropy (rather than so called "min-entropy") and therefore have a well behaved notion of conditional entropy. On the other hand, we have only one source (rather than two), and we use a very restrictive notion of subsources (as we only allow affine subsources). This means that we need to preserve affinity whenever the analysis wants to show the existence of a good subsource. This is in contrast to [BKS+10, BRSW06] that work with general sources (that allow distributions over general subsets rather than affine subspaces).[1]

We believe that in addition to the technical contribution of constructing affine dispersers, this paper makes a conceptual contribution in that it demonstrates the usability of the approach of [BKS+10, BRSW06] in other settings. Furthermore, our construction and its analysis are simpler, and require significantly less details and components than that of [BKS+10, BRSW06]. It is our view that presenting the approach in the setting of affine sources, makes it easier to grasp. We hope that this will allow subsequent research to apply the method described above for other explicit construction problems.

2 Preliminaries

We use small letters as much as possible as we reserve capital letters for random variables. Thus for example, for a matrix a that is defined as a function of x, we use A to denote the random variable $a(X)$ for a random variable X that is clear from the context. For a random variable X and an event E with positive probability we use $(X|E)$ to denote the distribution of random variable X when the probability space is conditioned on E. The entropy rate of a source X over $\{0,1\}^n$ is $H(X)/n$. We use $[n]$ to denote $\{1,\ldots,n\}$. For $x \in \{0,1\}^n$ and $s \subseteq [n]$ we define x_s to be the $|s|$ bit string given by restricting x to the indices in s. We say that a matrix is $r \times c$ if it has r rows and c columns, and use $\{0,1\}^{r\times c}$ to denote the space of such matrices. Thus, $\{0,1\}^{r\times c}$, $\{0,1\}^{c\times r}$ and $\{0,1\}^{r\cdot c}$ denote different things. For a matrix x, x_i denotes the i'th row of x, and for $p \leq c$, $\mathsf{slice}_p(x)$ denotes the $r \times p$ matrix obtained by keeping only the first p columns of x.

Somewhere-random sources: A distribution X is $r \times c$, *η-somewhere random* if X is over $\{0,1\}^{r\times c}$ and there exists $i \in [r]$ such that X_i is η-close to uniform. (This is equivalent to saying that X is η-close to a 0-somewhere random distribution). We omit η if it is zero, and omit the term "$r \times c$" if it is clear from the context. An affine somewhere random source is a somewhere-random source which is affine (when interpreted as a distribution over $\{0,1\}^{r\cdot c}$).

Somewhere-random extractors: A function $E : \{0,1\}^n \to \{0,1\}^{r\times c}$ is a somewhere-extractor for a family $\mathcal{C}$ with error η, if for every distribution $X \in \mathcal{C}$, $E(X)$ is η-somewhere random.

Block-wise sources: A distribution X over $\{0,1\}^{b\times n}$ is an affine block-wise source with entropy threshold k if it is an affine source (when interpreted as a distribution over $\{0,1\}^{b\cdot n}$) and for every $i \in [b]$, $H(X_i|X_1,\ldots,X_{i-1}) \geq k$.

[1] We remark that in an early version of [BKS+10] there was a construction of a disperser for affine sources with entropy δn for $\delta > 0$. While that construction is the inspiration for this one, the approach used there is very different form the one used here. More specifically, the idea there is to try to reduce the case of affine sources to that of multiple independent sources. The analysis used is very delicate, because it needs to consider general subsources rather than only affine ones. We also remark that the approach of that paper does not make sense for $k < \sqrt{n}$, and even for $k > \sqrt{n}$ it requires very strong components for general sources (that we do not know to construct for $k = o(n)$).

Linear seeded strong extractors: A linear seeded strong (k, ϵ)-extractor is a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for every $y \in \{0,1\}^d$, $E(\cdot, y)$ is linear, and furthermore for every distribution X that is uniform over a subset of size 2^k, $(E(X,Y), Y)$ is ϵ-close to uniform (where Y is independent for X and is uniform over $\{0,1\}^d$).

Useful properties of affine sources: Throughout the paper we make constant use of the following:

- If X is an affine source with $H(X) \geq k$ and E is a linear seeded strong (k, ϵ)-extractor then for a $(1 - 2\epsilon)$-fraction of $y \in \{0,1\}^d$, $E(X, y)$ is *completely* uniform.
- If X is an affine source with $H(X) \geq k$ and X' is a deficiency d affine subsource of X then $H(X') \geq k - d$. If X over $\{0,1\}^{2 \times n}$ is an affine block-wise source with entropy threshold k then for every possible value v of X_1, $H(X_2 | X_1 = v) \geq k$. Furthermore, if X' is a deficiency d affine subsource of X then X' is an affine block-wise source with entropy threshold $k - d$.

Organization of the paper. In Section 3 we show how to implement a "challenge-response game" for affine sources (assuming we can construct certain somewhere-extractors for affine sources). In Section 4 we show how the challenge-response game can be used to find blocks with large entropy, and to construct dispersers for 2-block affine block-wise sources. We use this to implement a dispersers for affine sources with large entropy. In Section 5 we explain how to construct extractors for affine block-wise sources (the full proof is deferred to Section 8). We also show how extractors for affine block-wise sources can be used to construct somewhere extractors for affine sources (that we need in order to implement the challenge-response game). In Section 6 we give a high level overview of the ideas used to get dispersers for low entropy.

3 Challenge-response games for affine sources

Let X be an affine source over $\{0,1\}^n$ with $H(X) \geq k$ for some parameter k. Let $s \subseteq [n]$ be a subset of size n'. We typically think of X_s as a block of X. Let $0 < k' < k$ be a parameter. We would like to distinguish between two cases:

- $H(X_s) = 0$ (the block X_s is "empty").
- $H(X_s) \geq k'$ (the block X_s has "large entropy").

Obviously, there does not exist a procedure which receives one sample from X and distinguishes the two cases. In this paper, we show how to implement a test that achieves a task with similar flavor. (For this purpose we will require somewhere-extractors for affine sources as an ingredient).

We start with some high level intuition. We imagine the following game: Block X_s tries to "convince" the source X that it has large entropy. For this purpose, block X_s generates a "challenge matrix" $\mathsf{ch}(X_s)$ where ch is a somewhere-random extractor for affine sources entropy k', so that $\mathsf{ch}(X_s)$ is (close to) somewhere random if $H(X_s) \geq k'$. The source X generates polynomially many "response matrices" $r_1(X), \ldots, r_{n^2}(X)$ such that at least one of them is completely uniform. This can be done using a linear seeded strong extractor. The challenge of X_s is responded if there exists a response matrix that equals the challenge matrix. If the challenge is not responded then X_s "wins" (and we say that the test passes). This serves as a demonstration that $H(X_s) \geq k'$.

Intuitively, any of the n^2 response matrices is very unlikely to hit the challenge matrix which is somewhere random. To make this intuition precise we will furthermore show that the challenge matrix is (close to) somewhere random even conditioned on every specific response matrix (and this will hold by slightly decreasing the entropy threshold of the somewhere-extractor ch). The latter property allows us to show that the test passes with high probability if $H(X_s) \geq k'$.

We would like to show that the test fails with high probability if $H(X_s) = 0$. However, this does not hold and indeed we cannot expect to distinguish the two cases using a single sample.

We can however show that the test fails with small positive probability. This is because the random response matrix has positive probability to be equal to the challenge matrix (as the latter is fixed if $H(X_s) = 0$). The crux of the approach is to show that if X is a source with $H(X) \geq k$ and $H(X_s) = 0$, then there exists an affine subsource X' of X such that on a random sample from X', the challenge is responded with probability one. Moreover, we show that X' is a small deficiency affine subsource of X, which implies that $H(X') \approx H(X) \geq k$. The interpretation is that the test indeed *correctly identifies* that the block is empty on the subsource X' (which is of the same type as X as $H(X'_s) = 0$ and $H(X')$ is not much smaller than k).

As explained in the introduction, when constructing dispersers we can imagine that $X = X'$ if we construct a disperser for X'. Thus, the statement above suffices for our purposes and it is helpful to imagine that we do have a test that distinguishes the two cases using one sample.

The construction appears below. We formalize the intuition above in Section 3.1 and demonstrate its usability in Section 4.

Construction 3.1 (Challenge-Response procedure $\mathsf{CR}(x, s)$)**.**

parameters: *integers n, ℓ, $k_{min} \geq (\log n)^a$ and $p \leq p_{max} \leq k_{min}^{\alpha}/\ell$ for some constants $a > 1$, $\alpha > 0$ to be determined later.*

inputs: *$x \in \{0,1\}^n$ and a "block" $s \subseteq [n]$. let $n' = |s|$ to denote the length of the block.*

ingredients:

- *A linear seeded strong $(k_{min}, 1/4)$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\ell \cdot p_{max}}$. By [SU05] there are explicit linear seed strong extractors E which for every $k_{min} > (\log n)^a$ (for some constant $a > 1$) use seed length $2 \cdot \log n$ and output k_{min}^{α} bits for some $\alpha > 0$. We fix E to be this extractor, and note that we previously required that $\ell \cdot p_{max} \leq k_{min}^{\alpha}$.*[2]
- *A function $\mathsf{ch} : \{0,1\}^{n'} \to \{0,1\}^{\ell' \times p_{max}}$ for some $\ell' \leq \ell$. (We use several choices for ch in the final construction and so we think of its as a parameter that is provided to the procedure CR).*

Operation: *of procedure $\mathsf{CR}_{\mathsf{ch},p}(x, s)$*

- *Compute the "challenge matrix" $c = \mathsf{ch}(x_s)$ which is an $\ell' \times p_{max}$ matrix. We pad c with dummy zero rows so that it as an $\ell \times p_{max}$ matrix if necessary.*
- *For all $y \in \{0,1\}^{2 \log n}$ compute $r(y) = E(x, y)$ interpreted as an $\ell \times p_{max}$ "response matrix".*

[2]The construction described below will go over all seeds of E and therefore it is crucial that E has seed $O(\log n)$. The fact that the leading constant in the seed length is 2 is not crucial. We remark that in the construction of [SU05], the constant 2 can be replaced with any constant larger than 1 (yielding better efficiency when going over all seeds). We also remark that if we allow seed length $O(\log n)$ for an arbitrary leading constant, we could also use an alternative construction from [SU05] which has output length $k^{1-\alpha}$ for some constant $\alpha > 0$.

- *We say that "the challenge is responded at* length p*" if there exists a* $y \in \{0,1\}^{2\log n}$ *such that* $\mathsf{slice}_p(c) = \mathsf{slice}_p(r(y))$*, and in that case the output of the procedure* $\mathsf{CR}_{\mathsf{ch},p}(x,s)$ *is defined to be 'fail' and otherwise the output is defined to be 'pass'.*[3]

Construction 3.1 follows the intuitive description above in case $p = p_{max}$ (so that the function slice_p is moot). We think of p as a "confidence" that CR has when it fails. This intuition is captured in the following trivial lemma.

Lemma 3.2. *Let* $p_1 \leq p_2 \leq p_{max}$*. If* $\mathsf{CR}_{\mathsf{ch},p_2}(x,s) = `fail'$ *then* $\mathsf{CR}_{\mathsf{ch},p_1}(x,s) = `fail'$*.*

The confidence parameter plays an important role in the disperser construction (and we explain this role later on). However, at this point, we suggest that the reader ignores parameter p and assume that CR is always applied with the highest confidence parameter $p = p_{max}$.

3.1 Properties of CR

We now show that CR fails on empty blocks (on some subsource, as explained above). This is captured in the second item of the lemma below. We will explain the role of the first item later on. We stress that the Lemma below makes no specific requirements on the function ch (other than being a function of the form specified in Construction 3.1).

Lemma 3.3 (CR fails on empty blocks)**.** *Let* $n, \ell, p, p_{max}, k_{min}$ *and* ch *be as in Construction 3.1.*

1. *For every affine source* X *over* $\{0,1\}^n$ *with* $H(X|X_S) \geq k_{min}$*,* $\Pr[\mathsf{CR}_{\mathsf{ch},p}(X,s) = `fail'] \geq 2^{-\ell \cdot p}$*.*
2. *For every affine source* X *over* $\{0,1\}^n$ *with* $H(X) \geq k_{min}$ *and* $H(X_s) = 0$ *there exists a deficiency* $\ell \cdot p$ *affine subsource* X' *of* X *such that* $Pr[\mathsf{CR}_{\mathsf{ch},p}(X',s) = `fail'] = 1$*.*

Proof. We first note that the first item follows from the second one as for every possible fixing of x' of X_s we have $Y = (X|X_s = x')$ satisfies $H(Y_s) = 0$ and $H(Y) \geq k_{min}$ and we can apply the second item on Y and conclude that CR fails with probability one on an affine subsource Y' which has deficiency $\ell \cdot p$ and thus has measure $2^{-\ell \cdot p}$ according to Y.

We now prove the second item. As $H(X) \geq k_{min}$ and E is a linear seeded strong extractor with entropy threshold k_{min}, there exists a seed y such that $R(y) = E(X,y)$ is uniform. Let T denote the event $\{\mathsf{slice}_p(E(X,y)) = \mathsf{slice}_p(\mathsf{ch}(X_s))\}$. We have that $\mathsf{ch}(X_s)$ is fixed and therefore T has probability $2^{-\ell \cdot p}$. Let $X' = (X|T)$ and note that X' is an affine subsource of X with deficiency $\ell \cdot p$ which satisfies the required property. □

The next lemma captures the intuition that CR passes on blocks with high entropy. In the aforementioned intuitive description, we were planning to use CR with a function ch which is a somewhere-random extractor. Unfortunately, we will not always have an explicit construction of a somewhere-random extractor with suitable parameters. Therefore, we state a weaker requirement on ch that still suffices for our purposes.

Definition 3.4. *Let* X, Y *be random variables. We say that* Y *is* somewhere random *with* error η *and* resiliency r *with respect to* X *if for every linear function* L *of rank* $\leq r$*, and every possible output* v *of* $L(X)$*, the distribution* $(Y|L(X) = v)$ *is* η*-somewhere random.*

[3] We choose to highlight ch, p in the notation above as these parameters will vary in the disperser construction, while the other parameters $n, \ell, p_{max}, k_{min}$ will be the same in all applications of CR. This will simplify the notation later on.

Lemma 3.5 (CR passes on blocks with high entropy)**.** *Let $n, \ell, p, p_{max}, k_{min}$ and ch be as in Construction 3.1. For every affine source X over $\{0,1\}^n$ such that $\mathsf{ch}(X_s)$ is somewhere random with error η and resiliency $\ell \cdot p$ with respect to X, $\Pr[\mathsf{CR}_{\mathsf{ch},p}(X,s) = \text{`}pass\text{'}] \geq 1 - n^2 \cdot (2^{-p} + \eta)$.*

Before proving Lemma 3.5, let us first explain that it is useful. In the informal discussion above we were planning to use CR to test whether $H(X_s) \geq k'$ for some threshold k'. We now explain that a somewhere-extractor ch with suitable parameters indeed meets the requirement of Lemma 3.5 and yields the aforementioned test. More precisely, if $H(X_s) \geq k'$ and ch is a somewhere-extractor for affine sources with entropy $k' - \ell \cdot p$ that has error η, then $\mathsf{ch}(X_s)$ is somewhere random with error η and resiliency $\ell \cdot p$ with respect to X as required in the lemma, and therefore $\mathsf{CR}_{\mathsf{ch},p}(X,s)$ passes with high probability. We will always choose parameters so that $\ell \cdot p = o(k')$ (and note that this dictates having the number of output rows ℓ in the output of ch satisfy $\ell \ll k'$), and then, a somewhere-extractor ch for affine sources with entropy $(1 - o(1)) \cdot k'$ can be used to apply CR and test whether blocks have entropy at least k'.

Loosely speaking, the advantage of the weaker condition in Lemma 3.5 is that rather than requiring that $\mathsf{ch}(Z)$ is somewhere random for all affine sources Z, it only requires that $\mathsf{ch}(X_s)$ is somewhere random on small deficiency affine subsources of the source X that we are interested in. This is helpful as we will be interested in sources X that have special properties and it will be easier to construct a function ch that performs well on such sources.

Proof. (of Lemma 3.5) For every $y \in \{0,1\}^{2\log n}$ of E, the function $L(x) = \mathsf{slice}_p(E(x,y))$ is linear and has rank $\leq \ell \cdot p$. By the guarantee on ch, for every possible value v of $L(X)$, $(\mathsf{ch}(X)|L(X) = v)$ is η-somewhere random. Therefore, $\mathsf{slice}_p(\mathsf{ch}(X))$ is somewhere random after fixing $L(X) = \mathsf{slice}_p(R(y)) = \mathsf{slice}_p(E(X,y))$. It follows that for every y and v, $\Pr[\mathsf{slice}_p(C) = \mathsf{slice}_p(R(y))|L(X) = v] \leq 2^{-p} + \eta$. Thus, for every y, $\Pr[\mathsf{slice}_p(C) = \mathsf{slice}_p(R(y))] \leq 2^{-p} + \eta$, and the Lemma follows by a union bound over the n^2 seeds $y \in \{0,1\}^{2\log n}$. □

4 Using the challenge-response game to construct affine dispersers

4.1 Roadmap for the remainder of the paper

In the previous section we saw how to implement the procedure $\mathsf{CR}(x,s)$ that can (in the sense explained above) distinguish the case that $H(X_s) = 0$ from the case that $H(X_s) = k'$ (where k' is a parameter). To implement CR we require an explicit construction of a function ch, and as discussed above, a somewhere-extractor for affine sources with entropy slightly smaller than k' will do (as long as the number of output rows ℓ is sufficiently smaller than k'). Unfortunately, we do not know how to construct such somewhere-extractors for affine sources with low entropy. In our final construction we will have to use weaker objects, and this creates many complications. We can construct a suitable somewhere-extractor for affine sources with entropy $k' = n^{1-\mu}$ for some constant $\mu > 0$, and this construction is sketched in Section 5 and proven in Section 8. This in turn can be used to construct dispersers for affine sources with entropy $k = n^{1-\rho}$ for some constant $\rho > 0$. We will present the construction of this disperser as a warmup in the remainder of this section. This will allow us to explain the ideas that are used in the final construction in a modular way. Later, in Section 6 we give an overview of the ideas needed to achieve $k < \sqrt{n}$. The final construction and analysis are given in Section 7.

We remark that in Section 7 we make no use of Sections 4,5 and 6 except that we use Definitions 4.1, 4.2, 5.2, and the statement of Theorem 5.1 (that is proven in Section 8). Therefore, the reader may skip the warmup at any time and go directly to the general case presented in Section 7 if he wishes to.

4.2 Nicely structured sources

We will use procedure CR to find good blocks in "nicely structured sources" that we define next.

Definition 4.1 (blocks)**.** *A block s is a subset $s \subseteq [n]$ defined by $s = \{a, \ldots, b\}$. We define $\mathsf{left}(s) = \{1, \ldots, a-1\}$. For $t \in [n]$ we define $s_i^t = \{(i-1) \cdot n/t + 1, \ldots, i \cdot n/t\}$. When t is clear from the context we omit it and use s_i or "block i" to refer to s_i^t.*

Definition 4.2 (nicely structured source)**.** *Let n, k' be parameters. An affine source X over $\{0,1\}^n$ is* nicely structured *for* block s *with* entropy k' *if $H(X_{\mathsf{left}(s)}) = 0$ and $H(X_s) \geq k'$. We say that X is* strongly-nicely structured *if in addition $H(X|X_s) \geq k'$.*

In a nicely structured source all blocks $j < i$ are empty and block i contains entropy. In a strongly nicely structured source, the pair (X_s, X) forms a 2-block affine block-wise source with entropy threshold k. It is easy to see that:

Lemma 4.3 (Existence of nicely structured affine subsources)**.** *For every n, k and $2 \leq t < k$, every affine source X over $\{0,1\}^n$ with $H(X) \geq k$ has an affine subsource $X^{(0)}$ that is nicely structured for some block i^* with entropy $k/4t$. If furthermore, $n/t \leq k/2$ then $X^{(0)}$ is also strongly nicely structured for block i^* with entropy $k/4t$.*

Proof. For every $i \in [t]$, let $k_i = H(X_{s_i}|X_{s_1}, \ldots, X_{s_{i-1}})$. By the chain rule for Shannon entropy, we have that $\sum_{i \in [t]} k_i \geq k$. Let i^* be the smallest i such that $k_i \geq k/4t$. It follows that $\sum_{j<i^*} k_j \leq k/4$. Let $(v_1, \ldots, v_{i^*-1})$ be some value in the support of $(X_{s_1}, \ldots, X_{s_{i^*-1}})$ and set $X^{(0)} = (X|X_{s_1} = v_1, \ldots, X_{s_{i^*-1}} = v_{i^*-1})$. Note that $X^{(0)}$ is an affine subsource of X with deficiency $k/4$ which is nicely structured for block i^* and entropy $k/4t$. We also have that $H(X^{(0)}) \geq H(X) - k/4$. If $n/t \leq k/2$ then $H(X^{(0)}|X^{(0)}_{s_{i^*}}) \geq k - k/4 - n/t \geq k - k/4 - k/2 \geq k/4$ and $X^{(0)}$ is strongly nicely structured. □

Note that the two requirements $t < k$ and $n/t \leq k/2$ (which together imply that $X^{(0)}$ is strongly nicely structured) also imply that $k \geq 2n/t > 2n/k$ which gives $k > \sqrt{2n}$. This is necessary as for $t < k < \sqrt{n}$ there does not necessarily exist an affine subsource of X that is strongly nicely structured, as blocks are of length $n/t > n/k > \sqrt{n} > k$ and so it may be that all blocks except for one are empty.

4.3 Finding good blocks in nicely structured sources

Lemma 4.3 says that every affine source X has an affine subsource that is nicely structured at some block i^*. Thus, by Fact 1.4, in order to construct dispersers for general affine sources, it suffices to construct dispersers for nicely structured affine sources. If the entropy is sufficiently large, it even suffices to construct dispersers for strongly nicely structured affine sources.

We now observe that procedure CR can be used to actually *find* a good block i^* in a nicely structured source. This is achieved by the following procedure.

Procedure FindBlock(x)**.** We are given $x \in \{0,1\}^n$. For every $1 \leq i \leq t$, apply $\mathsf{CR}(x, s_i)$ (we will explain the precise choice of parameters below) and let i' be the minimal i such that $\mathsf{CR}(x, s_i)$ passes.

Loosely speaking, the intuition is that every application of CR on a block $j < i^*$ (that is empty) will fail, while the application on block i^* will pass. More formally, in Lemma 4.4 below, we show that every

affine source X with $H(X) \geq k$ has an affine subsource X' which is nicely structured at some block i^* with entropy $\approx k/t$, and that furthermore, $\Pr[\mathsf{FindBlock}(X') = i^*] \geq 1 - o(1)$.

This reduces the task of constructing dispersers for general affine sources to constructing dispersers for nicely structured sources in which we *know* the good block i^* (and we will consider this problem in the next section).

In order to implement this approach we need to supply procedure CR with a somewhere-random extractor ch. In the lemma below we work out the parameters needed for implementing $\mathsf{FindBlock}$. It turns out that with a sufficiently good ch we can apply $\mathsf{FindBlock}$ even for sources of poly-logarithmic entropy.

Lemma 4.4 (Correctness of $\mathsf{FindBlock}$)**.** *There is a constant $\alpha > 0$ such that the following holds for every $k \geq polylog(n)$, $t \leq k^{1/3}$ and $\mathsf{ch} : \{0,1\}^{n/t} \to \{0,1\}^{\ell \times p_{max}}$ that is a somewhere extractor for affine sources with entropy $k/16t$ and error $2^{-p_{max}}$. Set $p = p_{max}$ and assume that $\ell, p_{max} \leq k^{\alpha}$ and $p_{max} \geq 3 \log n$. For every affine source X over $\{0,1\}^n$ with $H(X) \geq k$, there exists an affine subsource X' that is nicely structured for some block i^* with entropy $k/8t$ and furthermore, $\Pr[\mathsf{FindBlock}(X') = i^*] \geq 1 - 2^{-\Omega(p_{max})}$. Moreover, if $n/t \leq k/2$ then X' is strongly nicely structured for block i^**

Proof. By Lemma 4.3 there exists an affine subsource $X^{(0)}$ of X that is nicely structured for some block i^* with entropy $k/4t$ (and strongly nicely structured if $n/t \leq k/2$). We choose the parameters $k_{min} = k/8t$ and $p = p_{max}$ in all applications of CR and note that the requirements of Construction 3.1 are met. We now iteratively go over all $j < i^*$. For each j we apply Lemma 3.3 (2nd item) with respect to source $X^{(j-1)}$ and block s_j to obtain an affine subsource $X^{(j)}$. To apply the Lemma we need to make sure that $H(X^{(j-1)}_{s_j}) = 0$ which happens for every affine subsource of $X^{(0)}$. We also need to verify that $H(X^{(j-1)}) \geq k_{min}$. This holds initially for $X^{(0)}$ and we will maintain the invariant that $X^{(j)}$ is a deficiency $j \cdot \ell \cdot p_{max} \leq k/8t$ deficiency affine subsource of $X^{(0)}$ which implies that $H(X^{(j-1)}_{s_{i^*}}) \geq H(X^{(0)}_{s_{i^*}}) - j \cdot \ell \cdot p_{max} \geq k/4t - k/8t = k/8t$ and consequently $H(X^{(j-1)}) \geq k/8t = k_{min}$. We therefore can apply Lemma 3.3 and conclude that there exists an affine subsource $X^{(j)}$ of $X^{(j-1)}$ of deficiency $\ell \cdot p_{max}$ such that $\Pr[\mathsf{CR}_{\mathsf{ch},p_{max}}(X^{(j)}, s_j) = `fail'] = 1$.

Let $X' = X^{(i^*-1)}$. X' is a deficiency $t \cdot \ell \cdot p_{max} \leq k/8t$ subsource of $X^{(0)}$, and so we have that $H(X'_{s_{i^*}}) \geq k/4t - k/8t \geq k/8t$ and so it is indeed nicely structured for i^* with entropy $k/8t$. If $X^{(0)}$ is strongly nicely structured, we have that $H(X^{(0)}|X^{(0)}_{s_{i^*}}) \geq k/4t$ and as X' is an affine subsource with deficiency $k/8t$ we have that $H(X'|X'_{s_{i^*}}) \geq k/4t - k/8t = k/8t$. we know that $\mathsf{CR}_{\mathsf{ch},p_{max}}(X', s_j)$ fails with probability one for $j < i^*$. Thus, we only need to show that $\mathsf{CR}_{\mathsf{ch},p_{max}}(X', s_{i^*})$ passes with high probability in order to establish that $\mathsf{FindBlock}$ identifies i^* correctly. This follows from Lemma 3.5 as we have set up the parameters of ch so that $\mathsf{ch}(X'_{s_{i^*}})$ is somewhere random with resiliency $\ell \cdot p_{max}$. □

4.4 Dispersers for 2-block affine block-wise sources

By the previous discussion we can restrict our attention to affine sources X that are nicely structured for a block i^* that we know. Assume that we somehow achieve that the input source X is in fact *strongly* nicely structured. Note for example, that by Lemma 4.4 above this holds if say, $k \geq n^{3/4}$, $t = n^{1/3}$. Jumping ahead we mention that in our final construction, we will be able to arrange things so that we can assume w.l.o.g. that the source X we work with is strongly nicely structured even for low $k = n^{o(1)}$, and will be able to apply the techniques below for low k.

Let $s = s_{i^*}$ and note that by definition (X_s, X) forms a 2-block affine block-wise source with entropy threshold k_{min} for some parameter k_{min}. Thus, we are left with the task of designing dispersers for 2-block affine block-wise sources with entropy threshold k_{min}.

It turns out that procedure CR is versatile beyond the motivation explained in the previous sections. In fact, setting $\mathsf{Disp}_s(x) = \mathsf{CR}(x, s)$ produces a disperser in case (X_s, X) is a 2-block affine block-wise source (by simply interpreting pass/fail as zero/one).

We summarize the precise parameters in Lemma 4.5 below. We stress that when the entropy threshold of the block-wise source is $k_{min} = k/8t$ (as is the case after applying FindBlock) the parameters with which we run CR below are identical to those chosen in FindBlock and so if we have a somewhere-extractor ch to apply in FindBlock we can reuse it.

Lemma 4.5 (disperser for 2-block affine block-wise sources)**.** *Let $s \subseteq [n]$ be a subset of size n'. Let X be an affine source over $\{0,1\}^n$ so that (X_s, X) forms a 2-block affine block-wise source with entropy threshold k_{min}. Let $\mathsf{ch} : \{0,1\}^{n'} \to \{0,1\}^{\ell \cdot p_{max}}$ be a somewhere-random extractor for affine sources with entropy $k_{min} - \ell \cdot p_{max}$ for $\ell \cdot p_{max} \leq k_{min}^{\alpha}$ for some constant $\alpha > 0$ and assume that ch has error $\leq 1/n^3$. For every $3 \log n \leq p \leq p_{max}$ and every $v \in \{`pass', `fail'\}$, $\Pr[\mathsf{CR}_{\mathsf{ch},p}(X, s) = v] \geq 2^{-\ell \cdot p} > 0$.*

Proof. Let us analyze the output of $\mathsf{CR}_{\mathsf{ch},p}(X, X_s)$. We have that block X_s has entropy, and this is exactly the setup in which CR is supposed to pass. Consequently, by Lemma 3.5, it follows that $\Pr[\mathsf{CR}_{\mathsf{ch},p}(X, s) = `pass'] \geq 1 - n^2 \cdot (2^{-p} + \eta)$ which is at least $1/2$. We now observe that by Lemma 3.3 (first item), the fact that $H(X|X_s) \geq k_{min}$ implies that $\Pr[\mathsf{CR}_{\mathsf{ch},p}(X, s) = `fail'] \geq 2^{-\ell \cdot p} > 0$. □

4.5 Putting things together: a basic disperser

Putting things together, we have the following construction of a "basic disperser" $\mathsf{BasicD} : \{0,1\}^n \to \{0,1\}$ for affine sources which are strongly nicely structured at some unknown block i^*.

Basic Disperser: Given input $x \in \{0,1\}^n$

- For every $i \in [t]$ compute $\mathsf{CR}_{\mathsf{ch},p_{max}}(x, s_i)$.
- Apply FindBlock: Namely, let i' denote the smallest i such that the CR passes.
- $\mathsf{BasicD}(x) = \mathsf{CR}_{\mathsf{ch},p}(x, s_{i'})$ where $p \leq p_{max}$ is a parameter that we specify later.

There is a subtlety in making this approach go through. If we use $p = p_{max}$ in the application of CR in the third item, then the two applications of CR on block $X_{s_{i'}}$ are identical. In order for BasicD to output 'fail' we need that the first application outputs 'pass' and the second one outputs 'fail', but this is obviously impossible if the applications are identical.

We overcome this problem by choosing a small confidence parameter $p < p_{max}$. By Lemma 4.4, the probability that the second item fails to identify the good index i^* is at most $2^{-\Omega(p_{max})}$. On the other hand, the probability that the third item outputs a given $v \in \{`pass', `fail'\}$ is at least $2^{-\ell p}$. Thus, taking $p = o(p_{max}/\ell)$ we have that with positive (although small) probability, the block i^* is selected, and the procedure CR fails when applied with confidence level p (and this holds as long as $p \geq 3 \log n$).

Warmup: A disperser for affine sources of entropy $k = n^{1-\rho}$ for some $\rho > 0$. For sufficiently large k, by Lemma 4.4 every affine source with $H(X) \geq k$ has an affine subsource that is strongly nicely structured, and on which BasicD outputs both pass and fail with positive probability. By Fact 1.4 the procedure BasicD yields a disperser for general affine sources (assuming we can construct a somewhere-extractor ch with suitable parameters). In the next section we construct a somewhere-extractor for affine sources with entropy

$k^{1-\mu}$ for some constant $\mu > 0$. Plugging this somewhere-extractor into our machinery we obtain a disperser for affine sources with entropy $k \geq n^{1-\rho}$ for some $\rho > 0$.[4]

5 Somewhere extractors via extractors for affine block-wise sources

We need to construct somewhere-extractors for affine sources in order to apply procedure CR. We will construct somewhere-extractors (as well as weaker objects that still suffice for applying CR) using extractors for affine block-wise sources as a building block.

5.1 An extractor for $O(\frac{\log n}{\log k})$-block affine block-wise source

Theorem 5.1 (extractor for affine block-wise sources)**.** *There exists a constant $a > 1$ such that for every n and $k > (\log n)^a$ there is a polynomial time computable function $\mathsf{BE} : \{0,1\}^{b \times n} \to \{0,1\}^{k^{\Omega(1)}}$ that is an extractor for $b = O(\frac{\log n}{\log k})$-block affine block-wise sources with entropy threshold k, and error $2^{-k^{\Omega(1)}}$.*

The full proof of Theorem 5.1 appears in Section 8. We now provide a brief sketch. Our proof relies on ideas developed in [Rao09a, Rao09b] and an extractor construction of [Rao09b] that works for affine sources which are $k^{\Omega(1)} \times k$ somewhere random. We are given a b-block affine block-wise source $X_1, \ldots, X_b$ for $b = O(\frac{\log n}{\log k})$. Our first step is to transform X_1 into a somewhere random source with $v = n^{O(1)}$ rows. Each row is obtained by $E(X_1, y)$ where y is one of the v seeds of a linear seeded strong $(k, 1/4)$-extractor E with seed length $O(\log n)$ [SU05]. We divide the v rows into $v/k^{\Omega(1)}$ chunks where each chunk contains $k^{\Omega(1)}$ rows. We apply the extractor of [Rao09b] on each chunk, and as one of the chunks is somewhere random we obtain a source (close to) a somewhere random source with $v/k^{\Omega(1)}$ rows $S_1, \ldots, S_{v/k^{\Omega(1)}}$. We cannot apply the extractor of [Rao09b] on $S = (S_1, \ldots, S_{v/k^{\Omega(1)}})$ as it is not affine. Instead, for every $1 \leq i \leq v/k^{\Omega(1)}$ we compute $W_i = E(X_2, S_i)$ where E is a linear seeded strong extractor. If we had that X_1, X_2 are independent then we would be guaranteed that W_i is (close to) somewhere random. At this point, we would have consumed source X_1, and reduced the number of rows in the somewhere random source by a factor of $k^{\Omega(1)}$. Repeating this process, we could consume $b = O(\frac{\log n}{\log k})$ sources and reduce the number of rows from $n^{O(1)}$ to one.

While the sources $X_1, \ldots, X_b$ are not necessarily independent, we can use the fact that E is linear seeded to argue that the analysis above can be extended to the case that $X_1, \ldots, X_b$ form an affine block-wise source. Loosely speaking, this is because in an affine block-wise source (X_1, X_2) with entropy threshold k, X_2 can be represented as a sum of two sources where one is a function of X_1 and the other is independent of X_1. Thus, applying a linear seeded extractor $E(X_2, y)$ where y is a function of X_1 can be imagined to be operating on the independent part of X_2 once the analysis fixes source X_1.

5.2 Somewhere-extractors for affine sources

When we partition a source X into t blocks, we can hope that the partition induces a block-wise source. In fact, we will be happy even if the partition induces a block-wise source with $b < t$ blocks as we are shooting to construct somewhere-extractors and therefore don't mind trying all $\binom{t}{b} \leq t^b$ possibilities. This motivates the following definition and Lemma.

[4] We do not attempt to optimize ρ as this construction is only a warmup for our main construction which achieves $k = n^{o(1)}$. It is clear that the approach we used cannot get $\rho > 1/3$. It seems plausible that one can obtain $\rho \approx 1/4$, but this will require a careful optimization of the constants in some of the components that we use.

Definition 5.2 (hidden block-wise source)**.** *Let $b \leq k' \leq t \leq n$ be parameters. An affine source X over $\{0,1\}^n$ is a t-partition b-block affine hidden block-wise source with entropy threshold k' if there exist $i_1 < \ldots < i_b \in [t]$ such that $X_{s_{i_1}}, \ldots, X_{s_{i_b}}$ form a b-block affine block-wise source with entropy threshold k'. We omit the clause "t-partition" if t is clear from the context.*

Lemma 5.3. *Let $t \leq k' \leq n$ be parameters. Let $n' = n/t$ and $b = O(\frac{\log n'}{\log k'})$ be the number blocks used in Theorem 5.1 for $BE : \{0,1\}^{b \times n'} \to \{0,1\}^{(k')^{\Omega(1)}}$. Consider the function $\mathsf{SR}(x) = (\mathsf{BE}(x_{s_{i_1}}, \ldots, x_{s_{i_b}}))_{i_1 < \ldots < i_b \in [t]}$. If X is an affine source over $\{0,1\}^n$ that is a b-block hidden block-wise source with entropy threshold k', then $\mathsf{SR}(X)$ is $t^b \times (k')^{\Omega(1)}$, $(2^{-(k')^{\Omega(1)}})$-somewhere random.*

Note that SR runs in polynomial time if $t^b = n^{O(1)}$ (which will always hold for our choices of parameters as we will always have that $b = O(\log n / \log t)$). By an argument similar to that of Lemma 4.3, it is not hard to see that any affine source with sufficiently large entropy is an affine hidden block-wise source.

Lemma 5.4. *Let $t \leq k \leq n$ be integers such that $k \geq 2bn/t$ for $b = O(\frac{\log n}{\log k})$ from Lemma 5.3. Every affine source X with $H(X) \geq k$ is a b-block affine hidden block-wise source with entropy threshold $k' = k/4t$.*

The function SR is the component that is missing in the construction for affine dispersers for entropy $n^{1-\rho}$ given in Section 4.5. In the corollary below we set up the parameters for that application.

Corollary 5.5 (Somewhere-extractor for affine sources with large entropy)**.** *For every sufficiently small constant $\alpha > 0$ there is a constant $\mu > 0$ and a polynomial time computable function $\mathsf{ch} : \{0,1\}^n \to \{0,1\}^{n^{\alpha/3} \times n^{\alpha}}$ that is a somewhere-extractor for affine sources with entropy $n^{1-\mu}$, and has error $2^{-n^{\alpha}}$.*

6 High level overview of our construction for $k = n^{o(1)}$

We now outline some of the ideas that will allow us to extend our technique to $k = n^{o(1)}$. The following is an informal explanation and is meant to help the reader in understanding the construction and proof that are given in the next section. The reader may skip this Section if he wishes.

In order to extend our technique to $k = n^{o(1)}$ we need to solve the following two problems:

- The approach we developed can only work for $k > \sqrt{n}$. This is (amongst other things) because affine sources with entropy $k < \sqrt{n}$ are not guaranteed to have affine subsources which are strongly nicely structured.
- While the machinery we developed in Section 3 can potentially work for $k = n^{o(1)}$, we need to construct a somewhere-extractor ch for affine sources with entropy $n^{o(1)}$ in order to apply CR in that parameter regime.

To explain our ideas, it is easier to first assume that we already solved the second problem and can apply CR to distinguish empty blocks from blocks with entropy rate $\Omega(k/t)$. For simplicity of exposition, let us also cheat and assume that CR distinguishes empty blocks from high entropy blocks on the original source given to it (without having to consider subsources). We have already seen that informal arguments presented this way can be formalized by carefully considering affine subsources.

Using a win-win analysis to search for strongly nicely structured sources. We will solve the first problem using a win-win analysis introduced in [RSW06] (a similar high level approach was also used in [BRSW06]). We pick $t = k^{\beta}$ where β is a very small constant (or even slightly sub-constant). We show that any affine source over $\{0,1\}^n$ with entropy k, has a subsource X' that is either (i) strongly nicely structured with entropy $\Omega(k/t)$, or (ii) nicely structured with entropy threshold $\Omega(k)$. In the discussion below, we will once again ignore subsources and assume that the subsource is the original source. We already constructed dispersers for case (i). In case (ii), for the good block i^*, affine source $X_{s_{i^*}}$ is of length n/t and has entropy $\Omega(k)$. This means that $X_{s_{i^*}}$ has entropy rate that is $\Omega(t)$ times the rate of the original source. As X is nicely structured, we can use FindBlock to find block i^* and we now hold a source with better entropy guarantee than the one we started with. We continue this analysis recursively, and at each step either we obtain a strongly nicely structured source, or we multiply the rate by $\Omega(t)$. If this process continues for $O(\log n/\log t)$ steps, then we obtain an affine source of length n' with entropy $k' = (n')^{1-o(1)}$ for which we already constructed dispersers.

Distinguishing the two cases. A subtlety that we ignored in the explanation above is that we need to know which of the two cases happened in order to decide whether to stop and produce an output on $X_{s_{i^*}}$ (as in Section 4.4) or continue recursively on block $X_{s_{i^*}}$. We plan to use CR to distinguish case (i) from case (ii). We apply $\mathsf{CR}(X, s_{i^*})$ with a lower confidence parameter p than any of the confidence parameters that we previously used. We need CR to distinguish the case that $\Omega(k/t) \leq H(X_{s_{i^*}}) \ll k$ from the case where $H(X_{s_{i^*}}) \geq \Omega(k)$. Note that in both cases, the block $X_{s_i^*}$ has relatively high entropy and so CR passes with high probability in any of the two cases. Thus, it is very likely that we decide to continue (which is what we want in the second case). We use an argument that is somewhat similar to that used in Section 4.4 in order to argue that if we hold a strongly nicely structured source, then for any Boolean value v, there is positive (although small) probability that we stop at this point and output v. Thus, we indeed handle both cases correctly.

Computing challenges recursively. In the outline above we assumed that we already constructed a function ch that is a somewhere extractor for affine sources of entropy $n^{o(1)}$ with which we can implement CR. We will construct this function ch using once again, a win-win analysis. We remark that we would stop here and the construction and analysis would be much simpler, if we could construct somewhere extractors for affine sources of low entropy.

We show that every affine source X with entropy k, has an affine subsource that is either, (i) a b-block affine hidden block-wise source with entropy $\Omega(k/t)$, or, (ii) a nicely structured source with entropy threshold $\Omega(k/b) \gg k/t$. In the first case, we can produce a somewhere random matrix by considering all t^b candidate block-wise sources and applying our extractor BE (in a similar way to what is done in Lemma 5.3). In the second case, we can identify the good block using CR and apply ch recursively on a block with significantly higher entropy rate.

This argument may seem circular at first, as we use CR to implement ch and vice-versa. It is important to notice that we apply CR for testing blocks for higher entropy rate than the one we want ch to handle (and this enables us to use recursion). Unlike the previous argument, we do not need to distinguish the two cases (which is crucial as this would give a circular argument if we tried to distinguish the two cases). More precisely, as we are allowed to output a somewhere random source, we simply append the rows of the matrix obtained by the block s_{i^*} that we are considering and the rows of the matrix associated with its chosen sub-block. We know that one of these two matrices is somewhere random and therefore we obtain a somewhere random matrix.

Extending the argument to the real setup with subsources There are subtleties in implementing this recursion when we extend the argument to the true scenario and need to handle subsources. The problem is that we obtain a function ch that is only guaranteed to output a somewhere random distribution on a subsource of the original source that we work with. While this is always good enough when constructing dispersers, it is problematic when constructing extractors (and we want ch to be a somewhere extractor).

To handle this problem, we carefully choose properties of ch that we can maintain recursively, and argue that these properties suffice for applying CR as we go along. We need to be careful in the order in which the analysis goes down to subsources (and this leads to the resiliency property of ch with which Lemma 3.5 is stated).

A little bit more precisely, when given a source X we can first consider an affine subsource $X^{(0)}$ in which the previously described recursion tree, has a node that is strongly nicely structured, and a descendent that is a hidden block-wise source. We can then consider an affine subsource X' of $X^{(0)}$ on which CR fails on all blocks to the left of the blocks that we visit on the way, so that we have a chance to reach the two interesting nodes. This is not a problem, as Lemma 3.3 does not require anything from ch. We now want to apply Lemma 3.5 on the path leading to the two interesting nodes, so that we end up finding the two nodes. The resiliency property stated in Lemma 3.5 can be seen to take care of this problem if we let the resiliency parameter grow with the length of the path.

7 A disperser for affine sources with entropy $k = n^{o(1)}$

In this section we prove Theorem 1.2.

7.1 The construction

Construction 7.1 (Disperser for affine sources)**.** *We construct* $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ *as follows:*

Global parameters: *We use the global parameters defined below:*

- $k = 2^{\log^{0.9} n}$ *the entropy of the input source.*
- $t = 2^{\log^{0.3} n}$ *the partition parameter. We assume that* t *is an integer by taking a ceiling if necessary. Without loss of generality we assume that* n *is a power of* t *(as otherwise we pad input* x *to the nearest power* $\bar{n}$ *of* t *and note that* $k \geq 2^{(1-o(1))\cdot\log^{0.9} n}$ *is essentially unchanged).*

A tree of blocks: *We consider a degree* t *tree whose nodes are labeled by intervals* $s = \{a_1, \ldots, a_2\} \subseteq [n]$*. The tree is defined as follows: The root is* $[n]$*, and every node* s *in the tree has* t *children obtained by taking the interval* s *and partitioning it to* t *equal length intervals. More precisely, at level* z *of the tree there are* t^z *nodes and for* $1 \leq i \leq t^z$ *the* i*'th node in level* z *is the interval* $s = \{(i-1)\cdot n/t^z + 1, \ldots i \cdot n/t^z\}$*. Note that a node* s *is an ancestor of* s' *if and only if* $s' \subseteq s$*. The length of nodes at level* z *is* n/t^z*. We truncate the tree when this length becomes less than* $k^{1/4}$*. Later on, we will use the following: (i) the length of all nodes in the tree is at least* $k^{\Omega(1)} \geq t^{\log^{0.5} n}$*, and (ii) for sufficiently large level, nodes have length less than* $\sqrt{k}$*. Note that the number of nodes in the tree is polynomial in* n*.*

- *Let* $h \leq \frac{\log n}{\log t} \leq \log^{0.7} n$ *be the number of levels in the tree.*

Parameter choices for the extractor BE: *We use the extractor BE from Theorem 5.1 with the following parameters .Let b, p_{max} be parameters that we determine below, We set $\mathsf{BE} : \{0,1\}^{b \times n} \to \{0,1\}^{p_{max}}$. By Theorem 5.1 we can choose the parameters so that BE is an extractor for b-block affine block-wise sources with entropy threshold $\sqrt{k}$ and,*

- $b = O(\frac{\log n}{\log k}) = O(\log^{0.1} n)$,
- $p_{max} = 2^{\log^{0.8} n}$ *(and note that* $p_{max} \leq \sqrt{k}^{o(1)}$ *so that Theorem 5.1 applies),*
- *The error of BE is at most* $2^{-p_{max}}$.

In the construction we will allow ourselves to apply BE on inputs $x_1, \ldots, x_b$ where each one is of length $\leq n$ by first padding them to length n if necessary.

Parameter choices for procedure CR: *Throughout the construction we apply CR from construction 3.1 with the parameters:*

- $k_{min} = \sqrt{k}$.
- $\ell = 2^{\log^{0.5} n}$ *a bound on the number of rows in the challenge matrix.*
- $p_{max} = 2^{\log^{0.8} n}$ *was already chosen above, and note that* $p_{max} \leq k_{min}^{o(1)}/\ell$ *and the requirements in Construction 3.1 are met.*

Recall that whenever we want to apply CR on (x, s) for some node s we need to supply a function $\mathsf{ch} : \{0,1\}^{|s|} \to \{0,1\}^{\ell \times p_{max}}$ and a confidence parameter $p \leq p_{max}$. In the construction it will often be the case that we will have two nodes s, s' where s' is a descendent of s (and note that this means that $s' \subseteq s$) and we will want to apply CR on (x_s, s'). Formally, this can be done by padding x_s with zeroes in indices $i \notin S$ prior to applying CR (so that the first input to CR is of length n).

Active levels: *We set the following additional global parameters:*

- $q = \log^{0.5} n$. *We assume that q is an integer by taking a ceiling if necessary.*
- $h' = h/q \leq \log^{0.2} n$. *We assume that h' is an integer by taking a ceiling if necessary.*

The tree that we defined has h levels. Levels which are multiples of q are called "active" and nodes in active levels are called "active nodes". Thus for example, the root (that is of level 0) and nodes at level $q, 2q, 3q, \ldots$ are active. Note that h' is a bound on the number of active levels. A useful way to think about active nodes is that these nodes form a tree of degree t^q that has h' levels.

Each active level is of the form $z = uq$ for some integer $0 \leq u < h'$. For each such level $z = uq$ we define two confidence parameters $p_u^{(1)}, p_u^{(2)}$ by:

- $p_u^{(a)} = \ell^{6(h'-u)+3a}$ *for* $a \in \{1, 2\}$.

We plan to use these as confidence parameters in CR and so we verify that for every u and a, $p_u^{(a)} \leq \ell^{12h'} \leq 2^{12 \log^{0.7} n} \leq 2^{\log^{0.8} n} = p_{max}$ as required in Construction 3.1. (They were chosen this way so $p_1 \ll \ell p_2$ and that confidence decreases by a factor larger than ℓ as we go down the tree. This is similar in spirit to the choice of the two confidence parameters in Section 4.4). Jumping ahead, we explain that each active node will suggest an output for the disperser and the final output will be chosen in a way specified later by choosing an active node.

Operation of Disp: *On input $x \in \{0,1\}^n$ we do the following.*

1. **Compute challenges.** *In this step, each internal node s computes an $\ell'_s \times p_{max}$ matrix $\mathsf{ch}_s(x_s)$, for some $\ell'_s \leq \ell$. This is done by the following process:*
   - *For a node s whose children $s_1, \ldots, s_t$ are leaves, $\mathsf{ch}_s(x_s) = (\mathsf{BE}(x_{s_{i_1}}, \ldots, x_{s_{i_b}}))_{i_1 < \ldots < i_b \in [t]}$. This indeed gives at most $t^b \leq 2^{O(\log^{0.4} n)} \leq \ell$ rows. (We comment that this is the same idea used in Lemma 5.3).*
   - *For a node s whose children $s_1, \ldots, s_t$ already computed their challenges, we compute $\mathsf{ch}_s(x_s)$ as follows.*
     - *For every $i \in [t]$ we apply $\mathsf{CR}_{\mathsf{ch}_{s_i}, p_{max}}(x_s, s_i)$ and let i' be the smallest i such that CR passes. If no such i exists we pick i' arbitrarily.*
     - *The matrix $\mathsf{ch}_s(x_s)$ is defined by taking the matrix $\mathsf{ch}_{s_{i'}}(x_{s_{i'}})$ and adding more rows to it, where the added rows are: $(\mathsf{BE}(x_{s_{i_1}}, \ldots, x_{s_{i_b}}))_{i_1 < \ldots < i_b \in [t]}$.*
     - *Note that the number of rows in $\mathsf{ch}_s(x_s)$ is the number of rows in $\mathsf{ch}_{s_{i'}}(x_{s_{i'}})$ plus the number of additional rows which is bounded by t^b). Thus, in every step towards the root we increase the number of rows by at most t^b and so the number of rows in any node is bounded by $h \cdot t^b \leq 2^{\log^{0.5} n} = \ell$ as required.*
     - *Note that for every node s, the computation of $\mathsf{ch}_s(x_s)$ above can be seen as applying some function $\mathsf{ch}_s(\cdot)$ on x_s (as required in Construction 3.1).*
2. **Compute path.** *We now specify a path $w' = w'(x)$ from the root to some leaf as follows:*
   - *Assume that a prefix of the path is already defined and has reached node s (initially $s = [n]$ is the root). Let $s_1, \ldots, s_t$ be the children of s.*
   - *For every $i \leq t$, compute $\mathsf{CR}_{\mathsf{ch}_{s_i}, p_{max}}(x_s, s_i)$ and let i' be the smallest i such that CR passes. If no such i exists, we choose i' arbitrarily. We define the next node in the path w' to be $s_{i'}$.*
3. **Compute output.** *We now specify the output of $\mathsf{Disp}(x)$.*
   - *For every active node s that is on the path w' we do the following. The level of s is an active level $z = uq$ for some $0 \leq u < h'$. Let s' be the node on level $(u+1) \cdot q$ that is on the path w'. (If z is the last active level then s' does not exist and we skip the next item).*
   - *For $a \in \{1, 2\}$ we compute $o_s^{(a)}(x) = \mathsf{CR}_{\mathsf{ch}_{s'}, p_u^{(a)}}(x_s, s')$.*
   - *Let s^* be the active node s with the smallest active level on the path w' for which $o_s^{(1)}(x) = \text{`}fail'$. If no such s exists, we stop and output an arbitrary value.*
   - *The final output of the disperser is given by $\mathsf{Disp}(x) = o_{s^*}^{(2)}(x)$.*

Remark 7.2 (The role of the two trees)**.** *One way to think about the construction above is that we use two recursion trees: The tree of all nodes (that has degree t and h levels) and the tree of active nodes (that has degree t^q and $h = h/q$ levels). The first tree is used to compute the challenges, and we want this tree to have low degree t as in each node we apply BE on $t^b \geq t$ candidate block-wise sources, and it is important to keep the number of candidates small so that we don't have a challenge matrix with too many rows.*

The tree of active nodes is used to compute the final output. Here it is important that the number of levels is small (as we need to increase the confidence parameter when moving up the tree and cannot afford

too many levels). In order to reduce the number of levels we increase the degree from t to t^q (which is still much smaller than k and so is not a problem).[5]

7.2 The analysis: Proof of Theorem 1.2

Let Disp be the function from construction 7.1. We prove that Disp is a disperser for affine sources with entropy k. Let X be an affine source over $\{0,1\}^n$ with $H(X) \geq k$. Our plan is to show that there exists an affine subsource X' of X on which $\mathsf{Disp}(X')$ outputs two values. Throughout the proof we will assume that n is sufficiently large so that we can do asymptotic calculations and in particular that whenever $a < b$, $O(\log^a n) < \log^b n$. We will omit the clause "for sufficiently large n" from now on.

7.2.1 Finding nicely structured sources by win-win analysis

The first step is to show that there exists an affine subsource $X^{(0)}$ of X on which there is a root to leaf path that visits a node that is strongly nicely structured, and later visits a node that is a hidden block-wise source. Our high level plan is to try and show that the construction will select this path as w'.

Lemma 7.3. *There exists an affine subsource $X^{(0)}$ of X, and internal nodes g, g', and e such that:*

- *$e \subseteq g' \subseteq g$ (which means that g' is a descendent of g and e is a descendent of g').*
- *g, g' are active nodes on consecutive active levels.*
- *$H(X^{(0)}_g | X^{(0)}_{g'}) \geq k^{3/4}$.*
- *$X^{(0)}_e$ is a b-block hidden affine block-wise source with entropy threshold $k^{3/4}$.*
- *$H(X^{(0)}_{\mathsf{left}(e)}) = 0$.*

We have stated the precise properties of $X^{(0)}$ that will be used in proof later on. Before proving Lemma 7.3 let us relate the technical statement to the informal explanation of Section 6 and explain the overall strategy of the proof of Theorem 1.2.

7.2.2 High level strategy for the proof of Theorem 1.2

We note that the properties of $X^{(0)}$ in Lemma 7.3 imply:

[5] We make an advanced comment that can be safely skipped by the reader at a first reading. In the construction, we compute the path w' in step (2) by using the first tree. However, we could have defined w' to be a path of active nodes. More specifically, in that case, we replace $s_1, \ldots, s_t$ in item (2) with the t^q active nodes that are descendants of active node s on the next active level. In particular, if we had a somewhere extractor ch for affine sources of entropy k that outputs ℓ rows, we could have eliminated the first item of "computing challenges" in addition to the aforementioned modification. In that case, we would no longer need the first tree and the construction will only use the tree of active nodes.

In the proof below, we do rely on the fact that the two trees are "glued together". Specifically, it is useful that if X is nicely structured for some node s it is also nicely structured for every ancestor of s (whether active or not). However, in the proof we have tried to rely "as little as possible" on the fact that we use two trees at the same time (so that the same proof would apply if one can come up with a somewhere extractor ch with suitable parameters). In particular, when arguing that invocations of CR in item (2) pass, we refrain from using the precise way in which the challenge matrices were computed, and instead rely on abstract "resiliency" properties of these matrices.

- For every node $s \neq e$ that is an ancestor of e and every child s' of s that is left of e, $H(X^{(0)}_{s'}) = 0$. (Intuitively, this will allow us to apply Lemma 3.3 and find a subsource on which CR fails when testing the entropy of block s').
- For every node s that is an ancestor of e, $H(X^{(0)}_s) \geq k^{3/4}$. (Intuitively, this will allow us to apply Lemma 3.5 and show that CR passes when testing the entropy of block s).
- $X^{(0)}_g$ is strongly nicely structured at block g' with entropy $k^{3/4}$ and in particular, the pair $(X^{(0)}_{g'}, X^{(0)}_g)$ form a block-wise source with entropy threshold $k^{3/4}$. (Intuitively, this will allow us to use an argument similar to that of Section 4.4 and argue that when CR outputs two different values when applied on $(X^{(0)}_g, g')$).

Our high level plan is to try and show that (on some subsource X' of $X^{(0)}$ to be specified below) the final output of the disperser is given by the active node g by applying CR on (X'_g, g'). The last item above will guarantee that this produces both values with positive probability.

7.2.3 Proof of Lemma 7.3

We now give the proof of Lemma 7.3. The proof works by iteratively applying the ideas in the proof of Lemma 4.3. At each step either we argue that if the node s that we are holding does not have the desired property (strongly nicely structured, or hidden block-wise source) then there is a child of s in which the entropy rate is significantly increased. Details follow.

An iterative process for finding g, g'. We consider the following iterative process. We start with the source X and set $j = 0$ and $s = [n]$. We will maintain the invariant that X is an affine source over $\{0,1\}^n$ that is nicely structured for s with entropy $k/4^j$. Note that this indeed holds in the beginning of the process. In each iteration we apply the following lemma (where the parameter t' in the lemma is set to t^q and the parameter k' in the lemma is set to $k/4^j$).

Lemma 7.4 (win-win analysis for strongly nicely structured sources)**.** *Let $t' < k'$ be integers. Let $s \subseteq [n]$ be an interval and divide s into t' equal length intervals $s_1, \ldots, s_{t'}$. Let X be an affine source over $\{0,1\}^n$ that is nicely structured for s with entropy k'. Then there exists an affine subsource X' of X and $i^* \in [t']$ such that one of the following holds:*

- *X'_s is strongly nicely structured for s_{i^*} with entropy $k'/4t'$.*
- *X' is nicely structured for s_{i^*} with entropy $k'/4$.*

Proof. (of Lemma 7.4) We have that $H(X_s) \geq k'$. Let $k_i = H(X_{s_i}|X_{s_1}, \ldots, X_{s_{i-1}})$. By the chain rule we have that $\sum_{1 \leq i \leq t'} k_i \geq k'$. Therefore, there exists $i \leq t$ such that $k_i \geq k'/4t'$ and let i^* be the smallest such i. Note that $\sum_{j<i^*} k_j \leq k'/4$. Let v be a value in the support of $X_{\mathsf{left}(s_{i^*})}$ and note that $\Pr[X_{\mathsf{left}(s_{i^*})} = v] \geq 2^{-(k'/4)}$. We define $X' = (X|X_{\mathsf{left}(s_{i^*})} = v)$ and this is an affine subsource of X with deficiency $k'/4$. We have that $H(X'_{s_{i^*}}) \geq k_{i^*} \geq k'/4t'$ and $H(X'_{\mathsf{left}(s_{i^*})}) = 0$. It follows that X' is nicely structured for s_{i^*} with entropy $k'/4t'$. If $H(X'_{s_{i^*}}) \geq k/4$ then we have established the second item. If $H(X'_{s_{i^*}}) < k'/4$ then we have that $H(X'_s) \geq H(X_s) - k'/4 \geq k' - k'/4$ and

$$H(X'_s|X'_{s_{i^*}}) \geq H(X'_s) - H(X'_{s_{i^*}}) \geq (k' - k'/4) - k'/4 \geq k'/2 \geq k'/4t'.$$

Thus, we obtained that X'_s is strongly nicely structured at block s_{i^*} with entropy $k'/4t'$. □

When applying Lemma 7.4 we would like the first item to hold. If the first item in Lemma 7.4 does not hold then the second item holds. In that case, after applying the lemma we have that X' is nicely structured for s_{i^*} with entropy $k'/4 = k/4^{j+1}$. We set $X = X'$, $s = s_{i^*}$ and $j = j + 1$ and note that these choices meet the invariant above, so that we can apply the lemma once again. This process stops when the first item holds. We now claim that the process must stop after at most h' steps (and recall that h' is the number of active levels in the tree). This follows because in every iteration the length of s is divided by t^q while the entropy in block s is divided by at most 4. If the process does not hold after h' steps, then the entropy in block s is at least $k/4^{h'} > k^{3/4}$. Note that the last active level in the tree has nodes of length less than $\sqrt{k}$. It is impossible for a node to have entropy larger than its length, and thus the process must stop in less than h' steps.

When the process stops, the first item holds. We set $X^{(1)} = X'$, $g = s$ and $g' = s_{i^*}$. We indeed have that g, g' are active nodes. This is because Lemma 7.4 is applied with $t' = t^q$ and skips nodes in inactive levels. We furthermore have that $H(X^{(1)}_{\mathsf{left}(g')}) = 0$ and X'_g is strongly nicely structured at node g' with entropy at least $\frac{k/4^{h'}}{4t^q} \geq k^{4/5}$ (where the last inequality follows because $4^{h'} \leq t^q = 2^{\log^{0.8} n}$ and $k = 2^{\log^{0.9} n}$. It follows in particular that $H(X^{(1)}_{g'}) \geq k^{4/5}$ and $H(X^{(1)}_g | X^{(1)}_{g'}) \geq k^{4/5} \geq k^{3/4}$.

An iterative process for finding e. We now start another iterative process. We start by setting $j = 0$, $X = X^{(1)}$ and $s = g'$. We will maintain the invariant that X is an affine source over $\{0,1\}^n$ that is nicely structured for node s with entropy $k^{4/5}/(4b)^j$, and that furthermore $H(X_g | X_{g'}) \geq k^{3/4}$. Note that this indeed holds in the beginning of the process. In each iteration we apply the following lemma (where the parameter $k' = k^{4/5}/(4b)^j$).

Lemma 7.5 (win-win analysis for hidden block-wise sources)**.** *Let $t < k'$. Let $s \subseteq [n]$ be an interval such that $s \subseteq g'$ and divide s into t equal length intervals $s_1, \ldots, s_t$. Let X be an affine source that is nicely structured for s with entropy threshold k'. Then one of the following holds:*

- *X_s is an affine hidden block-wise source with entropy threshold $k'/4t$.*
- *There exists $i^* \in [t]$ and an affine subsource X' of X that is nicely structured for s_{i^*} with entropy $k'/4b$, and furthermore, $H(X'_g | X'_{g'}) = H(X_g | X_{g'})$.*

Proof. We have that $H(X_s) \geq k'$. Let $k_i = H(X_{s_i} | X_{s_1}, \ldots, X_{s_{i-1}})$. By the chain rule we have that $\sum_{1 \leq i \leq t} k_i \geq k'$. If there exist b indices $j \in [t]$ for which $k_j \geq k'/4t$ then X_s is a b-block hidden affine block-wise source and we are done. Otherwise, at most $b - 1$ indices $j \in t$ have $k_j > k'/4t$. The sum over all indices j for which $k_j \leq k'/4t$ is bounded by $k'/4$ and therefore, there must exist $i^* \in [t]$ such that

$$k_{i^*} \geq \frac{(k' - k'/4)}{b-1} \geq \frac{k'}{4b}.$$

Let v be a value in the support of $X_{\mathsf{left}(s_{i^*})}$ and define $X' = (X | X_{\mathsf{left}(s_{i^*})} = v)$. We have that $H(X'_{s_{i^*}}) \geq k_{i^*} \geq k'/4b$ and $H(X'_{\mathsf{left}(s_{i^*})}) = 0$. It follows that X' is nicely structured for s_{i^*} with entropy $k'/4b$. Furthermore, the source X' was defined by fixing X at an interval that is contained in g', therefore, $H(X'_g | X'_{g'}) = H(X_g | X_{g'})$. □

When applying Lemma 7.5 we would like the first item to hold. If the first item in Lemma 7.5 does not hold then the second item holds. In that case, after applying the lemma we have that X' is nicely structured for s_{i^*} with entropy $k'/4b = k^{4/5}/(4b)^{j+1}$. We set $X = X'$, $s = s_{i^*}$ and $j = j + 1$ and note that these

choices indeed meet the invariant above, so that we can apply the lemma once again. This process stops when the first item holds. We now claim that the process must stop after at most h steps (and in particular before we reach the leaves of the tree). This follows because in every iteration the length of s is divided by t while the entropy in block s is divided by at most $4b$. If the process does not hold after h steps, then the entropy in block s is at least $k^{4/5}/(4b)^h > k^{3/4}$. Note that the last active level in the tree has nodes of length less than $\sqrt{k}$. It is impossible for a node to have entropy larger than its length, and thus the process must stop before we reach the leaves and in less than h steps.

When the process stops, the first item holds. We set $X^{(0)} = X$ and $e = s$ and note that e is a descendent of g'. We have that $H(X^{(0)}_{\mathsf{left}(e)}) = 0$ and that $X^{(0)}_e$ is an affine hidden block-wise source with entropy threshold at least $\frac{k^{4/5}}{(4b)^h \cdot 4t} \geq k^{3/4}$. Furthermore, by the "furthermore clause" in the second item of Lemma 7.5 the final subsource $X^{(0)}$ inherits the good properties of the initial subsource $X^{(1)}$ and we have that $H(X^{(0)}_g | X^{(0)}_{g'}) = H(X^{(1)}_g | X^{(1)}_{g'}) \geq k^{4/5} \geq k^{3/4}$. This concludes the proof of Lemma 7.3.

7.2.4 Insuring that CR fails on empty blocks

Our next step is to show the existence of an affine subsource X' of $X^{(0)}$ that inherits all the good properties of $X^{(0)}$ (except that the threshold $k^{3/4}$ is replaced by the slightly smaller $k^{2/3}$) and has an additional useful property: the last item in Lemma 7.3 is replaced by the guarantee that for every $s \neq e$ that is an ancestor of e, CR fails on all children s' of s that are left of e. As we explain below, this implies that the path $w'(X')$ selected by the construction cannot reach a node that is left of e with positive probability.

Lemma 7.6. *There exists an affine subsource X' of $X^{(0)}$, and internal nodes g, g', and e such that:*

- *$e \subseteq g' \subseteq g$ (which means that g' is a descendent of g and e is a descendent of g').*
- *g, g' are active nodes on consecutive active levels.*
- *$H(X'_g | X'_{g'}) \geq k^{2/3}$.*
- *X'_e is a b-block hidden affine block-wise source with entropy threshold $k^{2/3}$.*
- *For every node $s \neq e$ that is an ancestor of e and for every child s' of s that is left of e, $\Pr[\mathsf{CR}_{\mathsf{ch}_{s'}, p_{max}}(X'_s, s') = `fail'] = 1$.*

Proof. Let $s \neq e$ be an ancestor of e and let s' be a child of s that is left of e. We would like to use Lemma 3.3 (2nd item) with respect to the source $X^{(0)}_s$ and the block s' to conclude that there exists a deficiency $\ell \cdot p_{max}$ affine subsource X' of $X^{(0)}$ such that $\Pr[\mathsf{CR}_{\mathsf{ch}_{s'}, p_{max}}(X'_s, s') = `fail'] = 1$. (Technically, the lemma only guarantees the existence of an affine subsource X'_s of $X^{(0)}_s$ but this trivially induces an affine subsource X' of $X^{(0)}$ and we will ignore this technicality in the remainder of the proof). We first verify that we indeed meet the conditions of the lemma in the sense that $H(X^{(0)}_s) \geq k_{min} = \sqrt{k}$ and $H(X^{(0)}_{s'}) = 0$. Indeed, we have that $H(X^{(0)}_{\mathsf{left}(e)}) = 0$, and therefore $H(X^{(0)}_{s'}) = 0$. Furthermore, we have that $e \subseteq s$ and $H(X^{(0)}_e) \geq k^{3/4}$. As, $e \subseteq s$ it follows that $H(X^{(0)}_s) \geq k^{3/4} \geq k_{min} = \sqrt{k}$.

Our plan is to apply Lemma 3.3 iteratively for every pair (s, s') as above (and note that there are at most $h \cdot t$ such pairs). Each time we apply the lemma we obtain an affine subsource of the affine source that we are holding. More precisely, we start with the affine source $X^{(0)}$ an at iteration i we obtain a deficiency $\ell \cdot p_{max}$ affine subsource $X^{(i)}$ of $X^{(i-1)}$. Note that the deficiency at each step is $\ell \cdot p_{max} \leq 2^{2\log^{0.8} n}$ and

therefore, even after $ht \leq 2^{\log^{0.4} n}$ iterations, we are losing at most $2^{3\log^{0.8} n}$ bits of entropy, and can argue that each of the intermediate sources $X^{(i)}$, satisfies

$$H(X_e^{(i)}) \geq H(X_e^{(0)}) - 2^{3\log^{0.8} n} \geq k^{3/4} - 2^{3\log^{0.8} n} \geq k^{2/3}.$$

Therefore, in each of the steps we meet the guarantee of the Lemma. Let X' denote the final affine subsource and note that X' is a deficiency $2^{3\log^{0.8} n}$ affine subsource of $X^{(0)}$. In particular, the two guarantees in items 3,4 in the statement of the lemma (that held in Lemma 7.3 with respect to threshold $k^{3/4}$) now hold with entropy threshold $k^{3/4} - 2^{3\log^{0.8} n} \geq k^{2/3}$ as required. □

7.2.5 Roadmap for the remainder of the proof

In the remainder of the proof we will work with the affine subsource X' of Lemma 7.6. Recall that on input x, every internal active node s on the path $w'(x)$, computes two values $o_s^{(1)}(x), o_s^{(2)}(x)$ and the output of $\mathsf{Disp}(x)$ is $o_s^{(2)}(x)$ for the active node s that is closest to the root amongst active nodes with $o_s^{(1)}(x) = `fail'$. We will show that for every $v \in \{0,1\}$, $\Pr[\mathsf{Disp}(X') = v] > 0$. Towards this goal we will be interested in the following events:

1. The path $w'(X')$ visits g'. (So that g "has a shot" to produce an output).
2. $o_g^{(1)}(X') = `fail'$. (So that g "decides" to produce an output).
3. $o_g^{(2)}(X') = v$. (So that the output that g produces is v).
4. For every active node $s \neq g$ that is an ancestor of g, $o_s^{(1)}(X') = `pass'$. (So that the output that g produces is not overruled by some active node that is an ancestor of g).

Note that whenever the four events above occur simultaneously, we indeed have that $\mathsf{Disp}(X') = v$. Our plan is to show that for both choices of $v \in \{0,1\}$, with positive probability, these four events happen simultaneously.

7.2.6 Insuring that CR passes on the path to the good nodes

We start by considering the first event in the roadmap. By the last item of Lemma 7.6 we have that with probability one, the path $w'(X')$ cannot reach a node that is left of e (and therefore cannot reach a node that is left of g'). (We remark that we were able to obtain this guarantee without considering specific properties of the challenge matrices computed in the construction, and this is because Lemma 3.3 does not require any specific properties from the challenge matrices). As we know that the path $w'(X')$ cannot reach a node that is left of g', in order to show that it visits g', it is sufficient to show that for every consecutive nodes s, s' on the path to g', $\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s')$ passes. For this purpose we plan to show that the probability that $\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s')$ fails for a fixed pair (s, s') is very small so that we can do a union bound over all nodes on the path to g'. By Lemma 3.5 we can indeed show that $\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s')$ passes with high probability if we can guarantee that the challenge matrix $\mathsf{ch}_{s'}(X'_{s'})$ is somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X'_s. We will therefore start with showing that this is the case.

Before doing so, we first observe that it is sufficient to replace the term "with respect to X'_s" in the previous sentence by "with respect to X'. This follows by the following trivial general observation (that is imediately implied by the definition of resiliency).

Lemma 7.7. *Let X, Y be random variables such that X is over $\{0,1\}^n$ and Y is η-somewhere random with resiliency r with respect to X. For every $s \subseteq [n]$, Y is also η-somewhere random with resiliency r with respect to X_s.*

We now implement the plan above. In the next lemma we argue that for every node s on the path to e (which includes the path to g'), $\mathsf{ch}_s(X'_s)$ is somewhere random with "high" resiliency with respect to X'. Note that this is sufficient for the discussion above (except that in the discussion above the node we now refer to as s played the role of s'). The proof is by induction on the distance between s and e and so we will be very particular about how the parameters evolve during the induction. Later on, we will be less picky and it will be sufficient that in all nodes along the path from the root to g', the error parameter is $2^{-\Omega(p_{max})}$, and, the resiliency parameter is at least $\ell \cdot p_{max}$.

Lemma 7.8. *Let s be an ancestor of e and assume that the path from s to e has a edges. Then, $\mathsf{ch}_s(X'_s)$ is $n^{4a} \cdot 2^{-p_{max}}$-somewhere random with resiliency $(h-a+1)\ell \cdot p_{max}$ with respect to X'.*

Proof. (of Lemma 7.8) We prove the lemma by induction on a. The base case is $a = 0$ which means that $s = e$. We have that X'_e is a b-block hidden affine block-wise source with entropy threshold $k^{2/3}$. We have set the entropy threshold of BE to $\sqrt{k}$. It follows that for every linear function L of rank $(h+1) \cdot \ell \cdot p_{max}$ and every value v, $(X'|L(X') = v)$ is a b-block hidden affine block-wise source with entropy threshold $k^{2/3} - (h+1) \cdot \ell \cdot p_{max} \geq k^{1/2}$. Node e tries all t^b candidate block-wise sources and so $(\mathsf{ch}_e(X'_e)|L(X') = v)$ is $2^{-p_{max}}$-somewhere random. By definition 3.4, this means that $\mathsf{ch}_e(X'_e)$ is $2^{-p_{max}}$-somewhere random with resiliency $(h+1)\ell \cdot p_{max}$ with respect to X' as required.

For $a > 0$, let s be the ancestor of e such that the path from s to e has a edges. Let s' be the child of s on the path to e. By induction we have that $\mathsf{ch}_{s'}(X'_{s'})$ is $n^{4(a-1)} \cdot 2^{-p_{max}}$-somewhere random with resiliency $(h-a+2)\ell \cdot p_{max}$ with respect to X'. We need to show that $\mathsf{ch}_s(X'_s)$ is $n^{4a} \cdot 2^{-p_{max}}$-somewhere random with resiliency $(h-a+1)\ell \cdot p_{max}$ with respect to X'. Consider any linear function L with rank $(h-a+1)\ell \cdot p_{max}$ and a possible output v of $L(X')$ and let $X'' = (X'|L(X') = v)$. Note that by definition, $\mathsf{ch}_{s'}(X''_{s'})$ is $n^{4(a-1)} \cdot 2^{-p_{max}}$-somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X''. (This is because $\mathsf{ch}_{s'}(X'_{s'})$ is resilient to first conditioning into X'' and even a further conditioning into a deficiency $\ell \cdot p_{max}$ affine subsource). By Lemma 7.7 we can replace the clause "with respect to X'' by "with respect to X''_s"". We can therefore apply Lemma 3.5 in X'' and conclude that

$$\Pr[\mathsf{CR}_{\mathsf{ch}_{s'}, p_{max}}(X''_s, s') = `pass'] \geq 1 - n^2 \cdot (2^{-p_{max}} + n^{4(a-1)} \cdot 2^{-p_{max}}) \geq 1 - n^{4a-1} \cdot 2^{-p_{max}}$$

We will now show that this implies that with at least this probability, in X'' the node s selects node s' as the node from which to take the challenge matrix. Indeed, for every child s'' of s that is left of s, s'' is also left of e and by Lemma 7.6 (last item) we have that

$$\Pr[\mathsf{CR}_{\mathsf{ch}_{s''}, p_{max}}(X''_s, s'' = `fail'] = 1$$

Therefore, in X'' with probability at least $1 - n^{4a-1}2^{-p_{max}}$ node s chooses node s' as the child from whom the challenge matrix is taken, and the matrix $\mathsf{ch}_s(X''_s)$ includes the rows of the matrix $\mathsf{ch}_{s'}(X''_{s'})$ which is $n^{4(a-1)} \cdot 2^{-p_{max}}$-somewhere random. By summing up the two errors, we conclude that $\mathsf{ch}_s(X''_s)$ is $n^{4a} \cdot 2^{-p_{max}}$-somewhere random. Thus, we showed that $\mathsf{ch}_s(X'_s)$ is $n^{4a} \cdot 2^{-p_{max}}$-somewhere random with resiliency $(h-a+1)\ell \cdot p_{max}$ with respect to X'. □

Having established Lemma 7.8 we state the following loose corollary.

Corollary 7.9. *Let s be an ancestor of e and assume that the path from s to e has a edges. Then, $\mathsf{ch}_s(X'_s)$ is $2^{-\Omega(p_{max})}$-somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X'.*

Now that we have established that challenges satisfy the properties needed in Lemma 3.5 we can apply it on every node s that is an ancestor of e and conclude that:

Lemma 7.10. *For every two nodes s, s' such that s is the father of s' and s' is an ancestor of e, $\Pr[\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s') = \text{`}pass\text{'}] \geq 1 - 2^{-\Omega(p_{max})}$.*

Proof. For every such two nodes s, s' we have by Corollary 7.9 that $\mathsf{ch}_{s'}(X'_{s'})$ is $2^{-\Omega(p_{max})}$-somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X'. By Lemma 7.7 we have that $\mathsf{ch}_{s'}(X'_{s'})$ is $2^{-\Omega(p_{max})}$-somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X'_s. We can therefore apply Lemma 3.5 with respect to $\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s')$ and conclude that $\Pr[\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s') = \text{`}pass\text{'}] \geq 1 - n^2 \cdot 2^{-\Omega(p_{max})} \geq 1 - 2^{-\Omega(p_{max})}$. □

We now show that with high probability the chosen path $w'(X')$ visits g'. (The same argument in fact shows that the path also visits e, but we won't need this later on).

Lemma 7.11. $\Pr[w'(X') \textit{ visits } g\text{'}] \geq 1 - 2^{-\Omega(p_{max})}$.

Proof. We consider the path from the root to g'. For every node $s \neq g'$ on this path and every child s' of s that is left of g', s' is also left of e and therefore by Lemma 7.6, $\Pr[\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s') = \text{`}fail\text{'}] = 1$. Consequently, to show that the path $w'(X')$ reaches g' it is sufficient to show that the event "For every two nodes s, s' such that s is the father of s' and s' is an ancestor of g', $\mathsf{CR}_{\mathsf{ch}_{s'},p_{max}}(X'_s, s')$ passes" holds. This event has probability $1 - 2^{-\Omega(p_{max})}$ by a union bound over the at most h nodes s using Lemma 7.10. □

7.2.7 Showing that active node g can produce two values

We now continue according to the roadmap of Section 7.2.5. We have that $w'(X')$ visits g' with probability almost one. We will now consider the second and third events in the roadmap. These events discuss random variables $o_g^{(a)}(X')$ for $a \in \{1, 2\}$. It will be more convenient to first consider slightly different variables that we define next. Recall that g is active and let $z = uq$ be the level of g in the tree. For $x \in \{0,1\}^n$ and $a \in \{1, 2\}$ we define:

$$r^{(a)}(x) = \mathsf{CR}_{\mathsf{ch}_{g'},p_u^{(a)}}(x_g, g')$$

Note that for every x such that $w'(x)$ visits g', the active node g uses active node g' to compute $o_g^{(a)}$ and therefore, $r^{(a)}(x) = o_g^{(a)}(x)$. This means that to understand the behavior of $o_g^{(a)}(X')$ it is enough to analyze $r^{(a)}(X')$.

Lemma 7.12. *Let $z = uq$ be the level of g. For every $v \in \{\text{`}pass\text{'}, \text{`}fail\text{'}\}$, $\Pr[r^{(1)}(X') = \text{`}fail\text{'} \textit{ and } r^{(2)}(X') = v] \geq 2^{-\ell^{6(h'-u)+7}}$.*

Proof. By Lemma 7.6 we have that $H(X'_g | X'_{g'}) \geq k^{2/3} \geq k_{min}$. By Lemma 3.3 (first item) we can conclude that for every $a \in \{1, 2\}$,

$$\Pr[r^{(a)}(X') = \text{`}fail\text{'}] = \Pr[\mathsf{CR}_{\mathsf{ch}_{g'},p_u^{(a)}}(X'_g, g') = \text{`}fail\text{'}] \geq 2^{-\ell \cdot p_u^{(a)}}.$$

We start with the case that $v = `fail'$. As confidence level $p_u^{(2)} \geq p_u^{(1)}$ then by Lemma 3.2 whenever $\mathsf{CR}$ fails when applied with confidence $p_u^{(2)}$ then it also fails when applied with confidence $p_u^{(1)}$. More formally, the event $\left\{r^{(2)}(X') = `fail'\right\}$ is contained in the event $\left\{r^{(1)}(X') = `fail'\right\}$. We conclude that:

$$\Pr[r^{(1)}(X') = `fail' \text{ and } r^{(2)}(X') = `fail'] = \Pr[r^{(1)}(X') = `fail'] \geq 2^{-\ell \cdot p_u^{(2)}} = 2^{-\ell^{6(h'-u)+7}}.$$

We now consider the case that $v = `pass'$. We consider the event

$$\left\{r^{(2)}(X') = `pass'\right\} = \left\{\mathsf{CR}_{\mathsf{ch}_{g'}, p_u^{(2)}}(X'_g, g') = `pass'\right\}$$

and apply Lemma 3.5 to show that it has high probability. We first verify that the conditions of the lemma are met. Indeed, by Corollary 7.9 and Lemma 7.7, $\mathsf{ch}(X'_{g'})$ is $2^{-\Omega(p_{max})}$-somewhere random for X'_g with resiliency at least $\ell \cdot p_{max} \geq \ell \cdot p_u^{(2)}$. We conclude that

$$\Pr[r^{(2)}(X') = `pass'] = \Pr[\mathsf{CR}_{\mathsf{ch}_{g'}, p_u^{(2)}}(X'_g, g') = `pass'] \geq 1 - n^2 \cdot (2^{-p_u^{(2)}} + 2^{-\Omega(p_{max})}) \geq 1 - 2^{-\Omega(p_u^{(2)})}.$$

Therefore,

$$\begin{aligned}\Pr[r^{(1)}(X') = `fail' \text{ and } r^{(2)}(X') = `pass'] &\geq \Pr[r^{(1)}(X') = `fail'] - \Pr[r^{(2)}(X') = `fail'] \\ &\geq 2^{-\ell \cdot p_u^{(1)}} - 2^{-\Omega(p_u^{(2)})} \geq 2^{-\ell^{6(h'-u)+4}} - 2^{-\Omega(\ell^{6(h'-u)+6})} \geq 2^{-\ell^{6(h'-u)+7}}\end{aligned}$$

□

7.2.8 Showing that Disp selects active node G and outputs two values

We are now finally ready to show that Disp outputs two values. The following Lemma concludes the proof of Theorem 1.2.

Lemma 7.13. *For every $v \in \{`pass', `fail'\}$, $\Pr[\mathsf{Disp}(X') = v] > 0$.*

Proof. Let $z = uq$ be the level of g. Let $v \in \{`pass', `fail'\}$ be some value. We consider the following events:

- Event A_1 defined as "$w'(X')$ visits g'".
- Event A_2 defined as "$r^{(1)}(X') = `fail'$ and $r^{(2)}(X') = v$".
- Event A_3 defined as "For every active node $s \neq g$ that is an ancestor of g, and s' that is the next active node on the path to g, $\mathsf{CR}_{\mathsf{ch}_{s'}, p_{u'}^{(1)}}(X'_s, s') = `fail'$".

We claim that the event $A_1 \cap A_2 \cap A_3$ is contained in $\{\mathsf{Disp}(X') = v\}$. This is because A_1 implies that for every active node s that is an ancestor of g, if we use $z = u'q$ to denote the level of s and let s' be the next active node on the path to g', then for every $a \in \{1, 2\}$, $o_s^{(a)}(X') = \mathsf{CR}_{\mathsf{ch}_{s'}, p_{u'}^{(a)}}(X'_s, s')$. In particular, for $a \in \{1, 2\}$, $o_g^{(a)}(X') = r^{(a)}(X')$. It follows that in $A_1 \cap A_2 \cap A_3$, the active node g is the active node s closest to the root for which $o_s^{(1)}(X') = `fail'$. This indeed implies that the event $A_1 \cap A_2 \cap A_3$ is contained in the

event $\left\{\mathsf{Disp}(X') = o_g^{(2)}(X') = v\right\}$. We conclude that it is sufficient to prove that $\Pr[A_1 \cap A_2 \cap A_3] > 0$. We have already analyzed the probability of A_1 and A_2. We therefore consider A_3.

Let $s \neq g$ be an active node that is an ancestor of g and denote its level by $z' = u'q$ and note that $u' < u$. Let s' be the next active node on the path to g. Every such node s' is an ancestor of g and therefore an ancestor of e. It follows by Corollary 7.9 and Lemma 7.7 that $\mathsf{ch}(X'_{s'})$ is $2^{-\Omega(p_{max})}$-somewhere random with resiliency $\ell \cdot p_{max}$ with respect to X'_s. This allows us to apply Lemma 3.5 with respect to $\mathsf{CR}_{\mathsf{ch}_{s'},p_{u'}^{(1)}}(X'_s, s')$ and conclude that

$$\Pr[\mathsf{CR}_{\mathsf{ch}_{s'},p_{u'}^{(1)}}(X'_s, s') = \text{`}pass'] \geq 1 - n^2 \cdot (2^{-p_{u'}^{(1)}} + 2^{-\Omega(p_{max})}) \geq 1 - 2^{-\Omega(p_{u'}^{(1)})} \geq 1 - 2^{-\Omega(p_{u-1}^{(1)})}$$

By a union bound over all active nodes s on the path to g we have that:

$$\Pr[\bar{A_3}] \leq h' \cdot 2^{-\Omega(p_{u-1}^{(1)})} \leq 2^{-\ell^{6(h'-(u-1))+2}} = 2^{-\ell^{6(h'-u)+8}}$$

Finally, we observe that:

$$\Pr[A_1 \cap A_2 \cap A_3] \geq \Pr[A_2] - \Pr[\bar{A_1}] - \Pr[\bar{A_3}]$$

By Lemma 7.12 we have that $\Pr[A_2] \geq 2^{-\ell^{6(h'-u)+7}}$. By Lemma 7.11, $\Pr[\bar{A_1}] \leq 2^{-\Omega(p_{max})} = 2^{-\Omega(\log^{0.8} n)}$. Therefore,

$$\Pr[A_1 \cap A_2 \cap A_3] \geq 2^{-\ell^{6(h'-u)+7}} - 2^{-\ell^{6(h'-u)+8}} - 2^{-\Omega(2^{\log^{0.8} n})} > 0$$

where the last inequality follows because $\ell^{h'} = 2^{\log^{0.7} n}$. □

This concludes the proof of Theorem 1.2.

7.2.9 On the behavior of the parameters in the proof

The proof of Theorem 1.2 is tailored to achieve $k = 2^{\log^{0.9} n}$. It is natural to ask whether the same approach can achieve a disperser for smaller k. We did not try to optimize the constant 0.9. This is because our approach used cannot achieve $k \leq 2^{\log^{0.5} n}$. There are several reasons for this.

The most obvious one is the losses suffered during the win-win analysis in Lemma 7.3 when we are looking for a hidden b-block block-wise source. In every step of the proof we consider a node s and its children $s_1, \ldots, s_t$. We are holding an affine source X with the guarantee that $H(X_s)$ is large (say at least k') and would like to have that X_s is a b-block block-wise source. It may be be the case that $H(X_{s_1}) = H(X_{s_2}|X_{s_1}) = k'/2$ and $H(X_{s_j}) = 0$ for $j > 2$. Assuming that $b > 2$, this is not good enough and X_s is not a hidden b-block block-wise source. Therefore, we will have to continue the win-win analysis on the block s_1. The crucial observation is that $H(X_{s_1}) \leq H(X_s)/2$ and we have lost half the entropy. Therefore, if we want to have entropy larger than 1 in the final block, we must have that $k \geq 2^h$ where h is the number of iterations. Let us now turn our attention to h. The parameter h is set to be the number of levels in the tree. It is induced by the degree of the tree which is t. We need the tree to contain nodes of length less than k and so the number of levels is at least $\frac{\log(n/k)}{\log t} = \Omega(\frac{\log n}{\log t})$ as we are interested in getting $k = n^{o(1)}$. We must choose $t < k$, this is because otherwise, a source X with entropy k may have the entropy split evenly between t blocks, giving that each block has entropy less than one which is useless. Thus, we have that $h \geq \frac{\log n}{\log k}$ and the requirement that $k \geq 2^h$ dictates $k \geq 2^{\log^{0.5} n}$.

An additional limitation on k is that our analysis requires that confidence levels increase by a factor of at least ℓ from one active level to the next one. Note that all confidence levels must be smaller than k (as

dictated in Construction 3.1). In our construction $\ell \geq t^b$. However, even if we had that $\ell = 2$ (which is the best we can hope for), then the largest confidence level is at least $\ell^{h'}$ where h' is the number of active levels. It follows that $k \geq \ell^{h'} \geq 2^{h'}$. The choice of h' depends on the choice of q as $h' = h/q$. We have already seen that $h \geq \Omega(\log n / \log k)$. Therefore, if we want $k \leq 2^{\log^{0.5} n}$ we must set q to be large so that h' is much smaller than h. However, we cannot set q to be too large. This is because in the win-win analysis for finding the good node g, we are partitioning the current source into t^q parts and must have that $k > t^q$ if we are hoping to get more than one bit of entropy in the good block. Summing up, we must satisfy $k > 2^{h'}$ and $k > t^q > 2^q$ for h', q such that $h' \cdot q = h \geq \Omega(\frac{\log n}{\log k})$. This dictates $k \geq 2^{\sqrt{\frac{\log n}{\log k}}}$ which implies $k \geq 2^{\log^{\frac{1}{3}} n}$. (We remark that a tighter analysis in which we use $\ell > t$ would show a limitation of $k \geq 2^{\log^{0.5} n}$.

Summing up, it seems that beating $k = 2^{\log^{0.5} n}$ would require new ideas. We do not have concrete suggestions at this point.

8 Extractor for affine block-wise sources with $O(\frac{\log n}{\log k})$ blocks

In this section we construct an extractor for affine block-wise sources and prove Theorem 5.1. We will utilize some nice properties of affine block-wise sources that are summarized next.

8.1 Structure of affine block-wise sources

We will make use of the following simple lemma. A special case of this lemma was stated in [Rao09b] and the lemma as stated follows from a more general case was stated in [Li11b]. We give a full proof for completeness.

Lemma 8.1. *Let $X = (X_1, X_2)$ be an affine source, and assume that $H(X_2|X_1) \geq k$. There is a linear function L such that affine sources $X_2^1 = L(X_1)$ and $X_2^2 = X_2 - L(X_1)$ satisfy:*

- $H(X_2^2) \geq k$
- *X_2^2 is independent of X_1.*
- $X_2 = X_2^1 + X_2^2$.

Proof. We assume that X is uniform over a linear space (meaning that the shift vector is zero). This is without loss of generality as the general case follows from the special case by simply adding the shift vector in the end. Let $k_1 = H(X_1)$ and $k_2 = H(X_2|X_1)$. Note that $H(X) = k_1 + k_2$. Let $Z = (X|X_1 = 0)$ be an affine subsource of X and note that $H(Z) = k_2$. Let $(a^1, \ldots, a^{k_2})$ be a basis for Z. Let $(b^1, \ldots, b^{k_1})$ be a continuation of the former basis such that together $B = (a^1, \ldots a^{k_2}, b^1, \ldots, b^{k_1})$ form a basis for X. Note that the vectors in $C = (b_1^1, \ldots, b_1^{k_1})$ span X_1, and as their number is k_1, C is a basis for X_1. The random variable X can be sampled by uniformly choosing a sequence of coefficients for the basis B and taking the linear span. More precisely, by X can be expressed as the following experiment: sample uniform and independent $R^1 \in \{0,1\}^{k_1}$, $R^2 \in \{0,1\}^{k_2}$ and set

$$X = \sum_{1 \leq i \leq k_1} R_i^1 \cdot b^i + \sum_{1 \leq i \leq k_2} R_i^2 \cdot a^i.$$

We have that $a_1^i = 0$ for every $1 \leq i \leq k_2$ and therefore:

$$X_1 = \sum_{1 \leq i \leq k_1} R_i^1 \cdot b_1^i.$$

We now define two affine sources X_2^1, X_2^2 that are defined as a function of R^1, R^2.

$$X_2^1 = \sum_{1 \leq i \leq k_1} R_i^1 \cdot b_2^i.$$

$$X_2^2 = \sum_{1 \leq i \leq k_2} R_i^2 \cdot a_2^i.$$

By definition $X_2 = X_2^1 + X_2^2$. We have that C is a basis for X_1 and therefore R^1 can be expressed as a linear function of X_1. By definition, X_2^1 is a linear function of R^1. It follows that X_2^1 is indeed a linear function of X_1. By definition, X_2^2 is a function of R^2 and therefore X_2^2 is independent of R^1 (and therefore of X_1). We also have that $H(X_2^2) \geq k_2 \geq k$ as X_2^2 is sampled by uniformly choosing a linear combination of k_2 linearly independent vectors. □

The usefulness of Lemma 8.1. Lemma 8.1 gives that when (X_1, X_2) form an affine 2-block block-wise source, we can divide X_2 to a part X_2^1 that depends on X_1 and a part X_2^2 with $H(X_2^2) \geq k$ that is independent of X_1. This is useful as it sometimes allows us to analyze X_1, X_2 as if they are independent.

Let us explain this idea more formally. Let $E(x_1, x_2)$ be a function that is linear in x_2 and has the additional property that for every two independent affine sources X_1, X_2 of "sufficient" entropy, for "most" choices of $x_1 \leftarrow X_1$, $E(x_1, X_2)$ is close to uniform. (In the extractor jargon, this property is called "strongness in X_1"). We now consider a block-wise source (X_1, X_2) in which X_1, X_2 can be correlated. We can use the lemma to show that the same conclusion holds, and for most choices of $x_1 \leftarrow X_1$, $(E(x_1, X_2)|X_1 = x_1)$ is close to uniform (and in particular that $E(X_1, X_2)$ is close to uniform).
Indeed, consider a fixed value x_1, and let $X' = (X|X_1 = x_1)$. Note that

$$X_2' = ((X_2^2 + L(x_1))|X_1 = x_1) = X_2^2 + L(x_1)$$

where the last equality relies on the independence of X_1 and X_2^2. We have that

$$(E(x_1, X_2)|X_1 = x_1) = E(x_1, X_2') = E(x_1, X_2^2 + L(x_1)) = E(x_1, X_2^2) + E(x_1, L(x_1)).$$

By using the guarantee of E on the independent sources X_1, X_2^2 we have that $E(x_1, X_2^2)$ is close to uniform for most x_1, and we obtain the required conclusion.

8.2 The construction

The construction of BE makes use of the following ingredients.

- A linear seed strong $(k, 2^{-k^\alpha})$ extractor $E : \{0,1\}^n \times \{0,1\}^{k^\mu/2} \to \{0,1\}^{k^\mu}$. Such an explicit construction is given in [Tre01] for some constants $0 < \alpha, \mu \leq 1$.
- A linear seed strong $(k, 1/4)$ extractor $E' : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{k^\mu}$. Such an explicit construction is given in [SU05] for a constant $\mu > 0$, where the constant hidden in the $O(\cdot)$ is universal.
- In [Rao09b] there is an explicit construction of an extractor for affine $k^\gamma \times k$ somewhere random sources (where $\gamma > 0$ is some constant). We apply this extractor for setting k to k^μ and conclude that there is a polynomial time computable function $\mathsf{Rao} : \{0,1\}^{k^{\mu \cdot \gamma} \times k^\mu} \to \{0,1\}^{k^\mu/2}$ that is an extractor for somewhere random sources with error $2^{-k^{\Omega(1)}}$. We can w.l.o.g. assume that the constant hidden in the $\Omega(1)$ above is α.

We first design a function $\mathsf{BE}'(x_1, x_2)$ where x_1 is a $v \times k^\mu$ matrix, $x_2 \in \{0,1\}^n$ and $BE'(x_1, x_2)$ outputs a $v/k^{\mu\cdot\gamma} \times k^\mu$ matrix. This component reduces the number of rows in the matrix, and the final construction will run it iteratively to obtain a single row.

Construction 8.2 (Building block BE'.)**.**

- *Take matrix x_1 and partition it to $v/k^{\mu\cdot\gamma}$ distinct blocks each consisting of consecutive $k^{\mu\cdot\gamma}$ rows. More precisely, for every $1 \leq i \leq v/k^{\mu\cdot\gamma}$ we define a $k^{\mu\cdot\gamma} \times k^\mu$ matrix w_i which rows are rows $(i-1)\cdot k^{\mu\cdot\gamma}+1, \ldots, i\cdot k^{\mu\cdot\gamma}$ in x_1.*
- *For every $1 \leq i \leq v/k^{\mu\cdot\gamma}$ apply Rao on the matrix w_i to obtain output $s_i \in \{0,1\}^{k^\mu/2}$.*
- *Let $z_i = E(x_2, s_i)$ and output the matrix z that has all z_i's as rows.*

Lemma 8.3. *Let X_1 be a $v \times k^\mu$ affine somewhere random source, let X_2 be an affine source such that $H(X_2|X_1) \geq k$, and let $Z = BE'(X_1, X_2)$. There exists a set G of $v \times k^\mu$ matrices such that:*

- $\Pr[X_1 \in G] \geq 1 - 3\cdot 2^{-k^\alpha}$.
- *For every $x_1 \in G$, $(Z|X_1 = x_1)$ is an affine somewhere random source.*

Proof. By Lemma 8.1 we can write X_2 as a sum $X_2 = X_2^1 + X_2^2$ where $X_2^1 = L(X_1)$ for some linear function L and X_2^2 is an affine source with $H(X_2^2) \geq k$ that is independent of X_1.

By the fact that E is a linear seed strong extractor with error 2^{-k^α} there is a set $G' \subseteq \{0,1\}^{O(\log n)}$ of relative size $\geq 1 - 2\cdot 2^{-k^\alpha}$ such that for every $y \in G'$, $E(X_2^2, y)$ is uniform. We have that X_1 is somewhere random and therefore there exists an i for which W_i is somewhere random. By the guarantee of Rao it follows that S_i is 2^{-k^α}-close to uniform. Thus, $\Pr[S_i \in G'] \geq 1 - 3\cdot 2^{-k^\alpha}$. Let G be the set of all x_1 that lead to $s_i \in G'$. We have that $\Pr[X_1 \in G] \geq 1 - 3\cdot 2^{-k^\alpha}$. For every $x_1 \in G$, we have that $(S_i|X_1 = x_1)$ is some constant $s_i \in G'$

$$(Z_i|X_1 = x_1) = (E(X_2, S_i)|X_1 = x_1) = (E(X_2, s_i)|X_1 = x_1)$$

$$= (E((X_2^1 + X_2^2), s_i)|X_1 = x_1) = ((E(X_2^1, s_i) + E(X_2^2, s_i))|X_1 = x_1)$$

We have that $(X_2^1|X_1 = x_1)$ is some constant $L(x_1)$ and therefore,

$$= ((E(L(x_1), s_i) + E(X_2^2), s_i)|X_1 = x_1) = (E(L(x_1), s_i) + E(X_2^2, s_i))$$

where the last equality follows because X_2^2 is independent of X_1. We have that $E(X_2^2, s_i)$ is uniform and therefore $(Z_i|X_1 = x_1)$ is uniform.

E is a linear seed extractor, and for every j, $Z_j = E(X_2, S_j)$ where S_j is a function of X_1. Thus, $(Z_j|X_1 = x_1)$ is an affine source for every x_1 in the support of X_1. □

We are now ready to prove Theorem 5.1.

Proof. (of Theorem 5.1). We define BE as follows: Given input $x \in \{0,1\}^{b\times n}$ we first go over all $v = n^{O(1)}$ seeds y of E' and compute $w_y = E'(x, y)$. Let w^1 be the matrix which has v rows and each row has w_y for some ordering of the seeds. We now iteratively compute matrices $w^2, w^3, \ldots$ by $w^{i+1} = BE'(w^i, x_{i+1})$. Note that w^i has $v/k^{\mu\gamma i}$ rows and so after $b = \frac{\log v}{\mu\cdot\gamma\cdot\log k} = O(\frac{\log n}{\log k})$ we obtain a matrix with a single row.

Let X be a b-block affine source with entropy threshold k. As E' is a linear seeded strong extractor we have that W^1 is an affine somewhere random source. Therefore, the sources in the first application of

BE' fulfill the guarantees of Lemma 8.3 and we conclude that there exists a set $G \subseteq \{0,1\}^n$ such that $\Pr[X_1 \in G] \geq 1 - 3 \cdot 2^{-k^\alpha}$ and for every $x_1 \in G$, $(W^2|X_1 = x_1)$ is an affine somewhere random source.

We can continue the analysis inductively maintaining the invariant that after the i'th application of BE' there is some set G_i such that $\Pr[(X_1, \ldots, X_i) \in G_i] \geq 1 - 3 \cdot i \cdot 2^{-k^\alpha}$ and for every $(x_1, \ldots, x_i) \in G^i$ the source W^{i+1} obtained after the i'th application of BE' satisfies that $(W^{i+1}|(X_1, \ldots, X_i) = (x_1, \ldots, x_i))$ is somewhere random. It follows that the final output of BE is $(3 \cdot O(\log n / \log k) \cdot 2^{-k^\alpha})$-close to uniform. For a sufficiently large constant a, we have that $k > (\log n)^a$ and therefore the error above is bounded by 2^{-k^ν} for some constant $\nu > 0$. □

9 Conclusion

We give an explicit construction of dispersers for affine sources with entropy $k = 2^{\log^{0.9} n} = n^{o(1)}$. Interesting open problems are:

- Improve the entropy threshold k to say polylog(n).
- Increase the number of extracted bits. We remark that by the technique of Gabizon and Shaltiel [GS08] improving the number of extracted bits to $m = 2 \log n$ will automatically give an improvement to $m = \Omega(k)$.
- Our construction would be significantly simpler if one can explicitly construct a somewhere extractor for affine sources with a small number of output rows. More specifically, to get a final result for entropy $k \geq 2^{\log^{0.9} n}$ we will require a somewhere extractor for entropy $\approx k$ that has a number of output rows which is significantly smaller than k, (say $k^{o(1)}$).
- Finally, our methods yield dispersers rather than extractors. It is open to explicitly construct extractors for affine sources with entropy $k = n^{o(1)}$.

Acknowledgements

I thank Ariel Gabizon and Raghu Meka for inspiring discussions. I am grateful to anonymous referees for many detailed comments and suggestions.

References

[BKS+10] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. *J. ACM*, 57(4), 2010.

[Bou07] Jean Bourgain. On the construction of affine extractors. *Geometric And Functional Analysis*, 17(1):33–57, 2007.

[BRSW06] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for sub-polynomial entropy and ramsey graphs beating the frankl-wilson construction. In *STOC*, pages 671–680, 2006.

[BSK09] Eli Ben-Sasson and Swastik Kopparty. Affine dispersers from subspace polynomials. In *STOC*, pages 65–74, 2009.

[BSZ11] Eli Ben-Sasson and Noga Zewi. From affine to two-source extractors via approximate duality. In *STOC*, 2011.

[DG10] Matt DeVos and Ariel Gabizon. Simple affine extractors using dimension expansion. In *IEEE Conference on Computational Complexity*, pages 50–57, 2010.

[DK11] Evgeny Demenkov and Alexander Kulikov. An elementary proof of $3n - o(n)$ lower bound on the circuit complexity of affine dispersers. Technical report, Electronic Colloquium on Computational Complexity, 2011.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

[GS08] Ariel Gabizon and Ronen Shaltiel. Increasing the output length of zero-error dispersers. In *APPROX-RANDOM*, pages 430–443, 2008.

[Li11a] Xin Li. Improved constructions of three source extractors. In *IEEE Conference on Computational Complexity*, 2011.

[Li11b] Xin Li. A new approach to affine extractors and dispersers. In *IEEE Conference on Computational Complexity*, 2011.

[Nis96] Noam Nisan. Extracting randomness: How and why a survey. In *IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[Rao09a] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM J. Comput.*, 39(1):168–194, 2009.

[Rao09b] Anup Rao. Extractors for low-weight affine sources. In *IEEE Conference on Computational Complexity*, pages 95–101, 2009.

[RSW06] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. *SIAM J. Comput.*, 35(5):1185–1209, 2006.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[Sha08] Ronen Shaltiel. How to get more mileage from randomness extractors. *Random Struct. Algorithms*, 33(2):157–186, 2008.

[Sha11] Ronen Shaltiel. An introduction to randomness extractors. In *ICALP (2)*, pages 21–41, 2011.

[SU05] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001.

[Vad02] S. Vadhan. Randomness extractors and their many guises. In *FOCS*, pages 9–12, 2002.

[Vad07] Salil P. Vadhan. The unified theory of pseudorandomness. *SIGACT News*, 38(3):39–54, 2007.

[Yeh10] Amir Yehudayoff. Affine extractors over prime fields. *Manuscript*, 2010.