Electronic Colloquium on Computational Complexity, Report No. 147 (2012)

New Independent Source Extractors with Exponential Improvement

Xin Li*
Department of Computer Science
University of Washington
Seattle, WA 98905, U.S.A.
lixints@cs.washington.edu

November 6, 2012

Abstract

We study the problem of constructing explicit extractors for independent general weak random sources. For weak sources on n bits with min-entropy k, perviously the best known extractor needs to use at least $\frac{\log n}{\log k}$ independent sources [Rao06, BRSW06]. In this paper we give a new extractor that only uses $O(\log(\frac{\log n}{\log k})) + O(1)$ independent sources. Thus, our result improves the previous best result exponentially. We then use our new extractor to give improved network extractor protocols, as defined in [KLRZ08]. The network extractor protocols also give new results in distributed computing with general weak random sources which dramatically improve previous results. For example, we can tolerate a nearly optimal fraction of faulty players in synchronous Byzantine agreement and leader election, even if the players only have access to independent n-bit weak random sources with min-entropy as small as $k = \text{polylog}(n)$.

Our extractor for independent sources is based on a new condenser for somewhere random sources with a special structure. We believe our techniques are interesting in their own right and are promising for further improvement.

*Supported by a Simons postdoctoral fellowship.

ISSN 1433-8092

1 Introduction

Motivated by the enormous applications in computation that rely on the use of truly uniform random bits (e.g, algorithm design, distributed computing and cryptography), and the fact that random sources in practice are rarely uniform, the broad area of *randomness extraction* studies the problem of converting a weakly random source into a distribution that is close to uniform. Here we measure the randomness in a random source X by the standard min-entropy.

Definition 1.1. The *min-entropy* of a random variable X is

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

For $X \in \{0,1\}^n$, we call X an $(n, H_\infty(X))$-source, and we say X has *entropy rate* $H_\infty(X)/n$.

Given an n-bit weak source X, a *randomness extractor* takes X as the input and outputs a distribution that is close to uniform in statistical distance. Ideally, one would like to construct a deterministic extractor that works for any source with enough min-entropy. However, it is not hard to show that no deterministic extractor can work for all sources with min-entropy as large as $n-1$. Instead, what we can construct is an extractor that uses an additional short uniform random seed. This is called a (strong) *seeded extractor*.

Definition 1.2. A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a *strong* (k, ε)*-(seeded) extractor* if for every source X with min-entropy k and independent R which is uniform on $\{0,1\}^d$,

$$(\mathsf{Ext}(X, R), R) \approx_\varepsilon (U_m, R),$$

where U_m is the uniform distribution on m bits independent of R, and $\approx_\varepsilon$ denotes the two distributions are within ϵ to each other in statistical distance.

The seed R is generally much shorter than the source, say only $O(\log n)$ bits. Although the extractor needs an additional random seed, it already suffices for some applications (e.g., simulating randomized algorithms with weak sources) just by trying all possible seeds, which only blows up the running time by a $\mathrm{poly}(n)$ factor. Besides this direct application, seeded extractors have found many other applications in computer science and nowadays we have explicit constructions with almost optimal parameters (e.g. [GUV09]). However, for applications such as distributed computing and cryptography, it is not clear how to use this trick. Instead, we need extractors that only use weak sources as inputs.

One kind of extractors that fits into this category is independent source extractors. These are extractors that take as input several independent weak sources, and output a distribution that is close to uniform. Indeed, these extractors are used in [KLRZ08, KLR09] to construct *network extractor protocols* that can be used to run distributed computing and cryptographic applications with weak random sources.

The study of independent source extractors dates back to the well known Lindsey's lemma, which gives an extractor for two independent (n, k) sources with $k > n/2$. Besides the applications in distributed computing and cryptography, independent source extractors are themselves interesting objects since they strongly resemble some properties of *random functions*. For example, using the probabilistic method, one can show that with high probability a random function is a deterministic extractor for just two independent sources with logarithmic min-entropy. Thus, constructing explicit independent source extractors is also closely related to the general problem of *derandomization*. However, although researchers have spent considerable efforts on this problem [CG88, BIW04, BKS$^+$05, Raz05, Bou05, Rao06, BRSW06, Li11], the known constructions are far from achieving optimal parameters. Currently the best explicit extractor for two independent (n, k) sources only achieves min-entropy $k = 0.49n$ [Bou05], the best explicit extractor for three independent (n, k) sources only achieves min-entropy $k = n^{1/2+\alpha}$ for an arbitrary constant $\alpha > 0$ [Li11], and the best explicit extractor for independent (n, k) sources requires $O(\log n/\log k)$ sources [Rao06, BRSW06].

1.1 Network extractor protocols

One application of independent source extractors is in distributed computing with imperfect randomness. Historically, Goldwasser, Sudan, and Vaikuntanathan [GSV05] were the first to consider this problem. They showed that it is possible to run distributed computing applications (e.g., Byzantine agreement) with imperfect randomness. However, they only considered fairly restricted weak sources. Kalai, Li, Rao and Zuckerman [KLRZ08] later improved this result to general weak random sources, where they also defined *network extractor protocols*.

The basic setting is, in a network with point to point or broadcast channels (synchronous or asynchronous), a set of players each has a private independent weak random source. They wish to communicate with each other so that at the end of the communication protocol, they end up with random strings that are close to being independent, uniform and private. However, some of the players are corrupted by an adversary, who is passive but otherwise can see every message transmitted in the network and has unlimited computational power. The protocol has to ensure that in the end, a large fraction of the honest players end up with private and uniform random strings. Below we give the formal definition of a network extractor protocol. For simplicity, in this paper we only consider the synchronous model.

We assume that p total players communicate with each other via point-to-point channels in order to perform a task, of which an unknown t are *faulty*. We allow Byzantine faults: faulty players may behave arbitrarily and even collude adversarially. In other words, we assume that the faulty players are controlled by an *adversary*. We assume that the the adversary can see all communication in the channels. This is called the *full information model*.

In a *synchronous* network, all communication takes place in rounds and every message transmitted at the beginning of a round is guaranteed to reach its destination at the end of the round. We allow rushing in this case: the faulty players may wait for all honest players to transmit their messages for a particular round, and then decide what to transmit for their own messages.

We now introduce some notation. Player i begins with a sample from a weak source $x_i \in \{0, 1\}^n$ and ends up with a hopefully uniform string $z_i \in \{0, 1\}^m$. Let b be the concatenation of all the messages that were sent during the protocol. We use capital letters such as Z_i and B to denote these strings viewed as random variables.

Definition 1.3. [KLRZ08] [Network Extractor] A protocol for p players is a (t, g, ϵ) *network extractor* for min-entropy k if for any min-entropy k independent sources $X_1, \ldots, X_p$ over $\{0, 1\}^n$ and any choice of t faulty players, after running the protocol, the number of players i for which

$$|(B, Z_i) - (B, U_m)| < \epsilon$$

is at least g. Here U_m is the uniform distribution on m bits, independent of B, and the absolute value of the difference refers to statistical distance.

The main goal of the network extractor protocol is to tolerate as many faulty players as possible (ideally a linear fraction), and to achieve g as close to $p-t$ as possible. In [KLRZ08], for any constant $0 < \beta < 1$, the authors constructed a $1/\beta + 1$ round $(t, p - (1.1 + 1/\beta)t, 2^{-k^{\Omega(1)}})$ network extractor protocol for min-entropy $k \geq 2^{\log^{\beta} n}$. Using this network extractor, they obtained synchronous Byzantine agreement protocols that tolerate roughly 1/4 fraction of faulty players for weak sources with $k \geq n^{\beta}$ and that tolerate roughly $1/(3 + 1/\beta)$ fraction of faulty players for weak sources with $k \geq 2^{\log^{\beta} n}$. For leader election they obtained similar results. Note that for $k \geq 2^{\log^{\beta} n}$, if β is small, although the protocol can still tolerate a linear fraction of faulty players, the fraction is quite small.

1.2 Our results

In this paper, we obtain new independent source extractors for general (n, k) sources that significantly improve the previous best results [Rao06, BRSW06]. Specifically, we have the following theorem.

Theorem 1.4. *For every $n, k \in \mathbb{N}$ with $k > \log^4 n$ there exists a polynomial time computable function $\mathsf{IExt} : (\{0,1\}^n)^t \to \{0,1\}^m$ with $m = \Omega(k)$ and $t = O(\log(\frac{\log n}{\log k})) + O(1)$ such that if $(X_1, \cdots, X_t)$ are t independent (n, k) sources, then*

$$\mathsf{IExt}(X_1, \cdots, X_t) \approx_{\epsilon} U_m,$$

where $\epsilon = 1/\text{poly}(k)$.

Thus, for (n, k) sources, our extractor only needs roughly $O(\log(\frac{\log n}{\log k}))$ sources to output a distribution that is close to uniform. Compared to the previous best result which uses $\Omega(\log n/\log k)$ sources, this is an *exponential* improvement.

We also show that our extractor works in a weaker setting, namely, when we only have a constant number of independent (n, k) sources and two independent k-block sources[1] with $O(\log(\frac{\log n}{\log k}))+O(1)$ blocks of size n.

Theorem 1.5. *There exists an absolute constant $c > 0$ such that for any $n, k \in \mathbb{N}$ with $k > \log^{10} n$ there exists a polynomial time computable function $\mathsf{BExt} : \{0,1\}^{cn} \times \{0,1\}^{tn} \times \{0,1\}^{tn} \to \{0,1\}^m$ with $m = \Omega(k)$ and $t = O(\log(\frac{\log n}{\log k}))$ $+ O(1)$ such that if $X = (X_1, \cdots, X_c)$ are c independent (n, k) sources and $Y = (Y_1 \circ \cdots \circ Y_t), W = (W_1 \circ \cdots \circ W_t)$ are 2 independent $(k, \cdots, k)$ block sources, then*

$$\mathsf{BExt}(X, Y, W) \approx_{\epsilon} U_m,$$

where $\epsilon = 1/\text{poly}(k)$.

Next, we apply our extractor to the network extractor protocols in [KLRZ08]. By using our improved independent source extractor, we also obtain improved network extractor protocols and protocols for Byzantine agreement/leader election with weak random sources. Specifically, we have

Theorem 1.6. *There exists a constant $c > 1$ such that for every $n, k, p, t \in \mathbb{N}$ with $k > \log^c n$, there is an explicit 2-round $(t, p - 3.1t, 1/\text{poly}(k))$ network extractor protocol for (n, k) sources.*

[1] A k-block source is a source with several blocks such that conditioned on any fixing of previous blocks, every block has min-entropy k.

Theorem 1.7 (Synchronous Byzantine Agreement). *There exists a constant $c_1 > 1$ such that for any constants $\alpha > 0$ and $c_2 > 1$ the following holds. Assume p players each has access to an independent (n, k)-source with $k > \log^{c_1} n$ and $k > p^{1/c_2}$, then there exists an explicit (in n) synchronous $O(\log p)$ expected round protocol for Byzantine Agreement in the full information model that tolerates $(1/5 - \alpha)p$ faulty players.*

Theorem 1.8 (Leader Election). *There exists a constant $c_1 > 1$ such that for any constants $\alpha > 0$ and $c_2 > 1$ the following holds. Assume p players each has access to an independent (n, k)-source with $k > \log^{c_1} n$ and $k > p^{1/c_2}$, then there exists an explicit (in n) synchronous $\log^* p + O(1)$ round protocol for leader election that tolerates $(1/4 - \alpha)p$ faulty players.*

Note that here we can tolerate a nearly optimal fraction of faulty players (for Byzantine agreement, the optimum is $1/3$ fraction and for leader election, the optimum is $1/2$ fraction), even for weak sources with min-entropy as small as $k = \text{polylog}(n)$. These results dramatically improve previous results.

2 Overview of The Constructions and Techniques

Here we give a brief overview of our constructions and the techniques. To give a clear description of the ideas, we shall be informal and imprecise sometimes.

2.1 Independent source extractor

Similar as in [Rao06, BRSW06], our extractor is obtained by repeatedly condensing somewhere random sources (SR-source for short). Take an (n, k) source X and a strong seeded extractor Ext with seed length $O(\log n)$, by applying Ext to X with all possible choices of the seed, we obtain an SR-source with $N = \text{poly}(n)$ rows such that at least one row (in fact, most of the rows) is (close to) uniform. The condenser in [Rao06, BRSW06] reduces the number of rows in the SR-source from N to $N/k^{0.9}$ each time, while consuming a constant number of independent (n, k) sources. Once the number of rows decreases to $k^{0.9}$, extraction becomes easy with an additional two independent (n, k) sources. This results in a total number of $O(\frac{\log n}{\log k})$ sources.

The decreasing of the number of rows from N to $N/k^{0.9}$ is inherently limited by the techniques in [Rao06, BRSW06]. In this paper, however, by using a new condenser, we can reduce the number of rows in the SR-source much faster. Specifically, each time by consuming just one independent (n, k) source, our condenser reduces the number of rows in the SR-source from N to $N^{3/4}$. If $N >> k$ then $N^{1/4} >> k^{0.9}$. Note that initially $N = \text{poly}(n)$, thus especially for small k such as $k = \text{polylog}(n)$, our condenser performs much better than the condenser in [Rao06, BRSW06].

Once we have this condenser, we can use it repeatedly to reduce the number of rows in the SR-source to say k^5. At this time we can use the extractor in [Rao06, BRSW06] to extract random bits with a constant number more of independent sources. Since initially $N = \text{poly}(n)$, the condensing process uses $O(\log(\frac{\log n}{\log k})) + O(1)$ sources. Thus our extractor uses $O(\log(\frac{\log n}{\log k})) + O(1)$ sources.

We now describe our condenser. Unfortunately, to get this super efficiency we have to sacrifice some generality. Unlike the condenser in [Rao06, BRSW06], our condenser does not work for a general SR-source, but it works for SR-sources with some special structure. We now explain in more details. As mentioned before when we take a strong seeded extractor Ext with seed length $O(\log n)$ and applies it to a source X with all possible choices of the seed, we obtain a SR-source with $\text{poly}(n)$ rows such that most of the rows are (close to) uniform. Ignoring the error, now suppose these rows are indeed uniform, and moreover, these rows are *independent*. We note that it is not clear at all that we can achieve this (in fact, it's impossible to achieve with just one weak source), but for now let us assume that we can indeed get such an SR-source. Now we want to reduce the number of rows in the SR-source while still keeping it to be an SR-source with the same structure, what can we do?

We will now borrow some ideas from a distributed computing problem. Imagine that in the SR-source, each row is associated with a player, and each player has a string that is supposed to be uniform and independent, which is the corresponding row in the SR-source. Those rows that are uniform and independent are associated with honest players, since their strings are indeed uniform and independent. The other rows are associated with faulty players, since their strings may not be uniform and may depend arbitrarily on the honest players' strings. Now we want to select a committee from the players, which has size much smaller than the number of players and which has roughly the same fraction of honest players. This problem is very similar to our condenser problem. On the other hand, this is a well-studied problem in leader election [RZ01, Fei99]. In particular, Feige [Fei99] gave a beautiful *lightest bin* protocol to solve this problem.

The lightest bin protocol is as follows. Take r bins and each player uses his random string to randomly select a bin. The players who select the *lightest* bin (the bin selected by the fewest number of players) form the selected committee. The idea is that, since the strings of the honest players are uniform and independent, by a Chernoff bound with high probability the honest players are roughly evenly distributed into each bin. Thus, no matter how the faulty players' strings depend on the honest players' strings, in the lightest bin the fraction of faulty players cannot be much larger than the original fraction of faulty players.

Back to our condenser problem, we can use the same lightest bin protocol to select a subset of the rows, such that with high probability the "good" rows (rows that are originally uniform and independent) in this subset has roughly the same fraction. Now we take another independent (n, k) source X' and apply the strong seeded extractor Ext to X' using each row in the selected subset as the seed. Ignoring the error, and assume that k is larger than the size of the subset times the output size of the extractor, one can show that with high probability conditioned on the fixing of the original SR-source, the outputs of the extractor which correspond to the good rows are also uniform and independent. Moreover, these outputs are now a deterministic function of the new source X' (since the original SR-source is fixed). This corresponds to one round in the original lightest bin protocol in leader election. Note that after we select the subset of rows, we need to use a new source X' to get another SR-source for the next "round". This is because in the lightest bin protocol the faulty players' strings can depend arbitrarily on the honest players' strings in the same round, but cannot depend on the honest players' strings in future rounds. Thus, by consuming one independent source we have obtained a new SR-source with fewer rows and the same structure as the original SR-source. We can now iterate the above process. This gives our condenser.

Note that if the good rows in the SR-source are indeed uniform and independent, then by a Chernoff bound we can take the number of bins to be $r = N/\log N$ and each time we can reduce the number of rows from N to $\log N$. This is how [Fei99] achieves a leader election protocol with $\log^* p + O(1)$ rounds, where p is the number of players. Back to our extractor problem, in one step we can reduce the number of rows in the SR-source from $N = \text{poly}(n)$ to $O(\log n)$. Thus, as long as $k \geq \text{polylog}(n)$ we can use the extractor in [Rao06, BRSW06] to extract random bits from the SR-source and ONE additional (n, k) source. This will give us an extractor for (n, k) sources with

k as small as polylog(n) that uses just a constant number of sources!

This sounds really great, except that we cannot achieve an SR-source such that the good rows are indeed uniform and independent. What we can achieve now is that every pair of the good rows is close to being uniform and independent. In other words, ignoring the error, we can achieve pair-wise independence in the good rows. Thus, we cannot apply the Chernoff bound in the analysis. Luckily, by pair-wise independence we can still apply Chebysev's inequality, which guarantees that in one step we can take the number of bins to be $r = N^{1/4}$ and reduce the number of rows in the SR-source from N to $N^{3/4}$. This is our actual condenser.

So now the question is how to obtain an SR-source such that the good rows are close to being pair-wise independent. For this, we need the definition of a non-malleable extractor.

Definition 2.1. [DLWZ11] A function $\mathsf{nmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)-non-malleable extractor if, for any source X with $H_\infty(X) \geq k$ and any function $\mathcal{A} : \{0,1\}^d \to \{0,1\}^d$ such that $\mathcal{A}(r) \neq r$ for all r, the following holds. When R is chosen uniformly from $\{0,1\}^d$ and independent of X,

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

Note that a non-malleable extractor is a stronger version of a strong extractor, in the sense that the output is required to be close to uniform conditioned on both the seed and the output on another different but otherwise arbitrarily correlated seed. Non-malleable extractors were originally introduced in [DW09] to study the problem of privacy amplification with an active adversary. We now claim that if we take a source X and apply a non-malleable extractor nmExt to X with all possible choices of the seed, then ignoring the error, we obtain an SR-source such that a large fraction of the rows are pair-wise independent. Indeed, for any seed r if there exists a seed $r' \neq r$ such that $\mathsf{nmExt}(X,r)$ is not close to uniform conditioned on $\mathsf{nmExt}(X,r')$, then we can let $\mathcal{A}(r) = r'$. The definition of a non-malleable extractor asserts that the fraction of these r's is small. Thus, for the rest of the seeds, the outputs of the non-malleable extractor are pair-wise independent.

Now this is very nice, except another problem. Currently the best explicit non-malleable extractor only works for $k = 0.49n$ [Li12c], while we need constructions for essentially any min-entropy. Thus, we switch to a relaxation of a non-malleable extractor, a non-malleable condenser.

Definition 2.2. [Li12a] A (k, k', ϵ) non-malleable condenser is a function $\mathsf{nmCond} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that given any (n,k)-source X, an independent uniform seed $R \in \{0,1\}^d$, and any (deterministic) function $\mathcal{A} : \{0,1\}^d \to \{0,1\}^d$ such that $\forall r, \mathcal{A}(r) \neq r$, we have that with probability $1-\epsilon$ over the fixing of $R = r$,

$$\Pr_{z' \leftarrow \mathsf{nmCond}(X,\mathcal{A}(r))}[\mathsf{nmCond}(X,r)|_{\mathsf{nmCond}(X,\mathcal{A}(r))=z'} \text{ is } \epsilon-\text{close to an } (m,k') \text{ source}] \geq 1-\epsilon.$$

Non-malleable condensers were introduced in [Li12a]. Note that it is indeed a relaxation of a non-malleable extractor in the sense that it only requires the output to have a certain amount of min-entropy. Once we have a non-malleable condenser nmCond, we can apply it to a source X with all possible choices of the seed, and (ignoring the error) we obtain a source such that for a large fraction of the rows, each pair of the rows is a k'-block source with 2 blocks of size m. Recently, Li [Li12b] constructed explicit non-malleable condensers for essentially any min-entropy, with error $\epsilon = 1/\text{poly}(n)$ and seed length $d = O(\log^2 n)$ such that $k' > \sqrt{m}$. However, this does not give us an SR-source. To fix this, we take several independent sources and from each one we obtain a source with N rows by applying the non-malleable condenser. For each source, let $S_i \subset [N]$ be the set of "good" rows. Now let $S = \cap S_i$ and S still takes up a large fraction of $[N]$. Moreover, the good rows in S are now aligned across these sources. Now for every $j \in S$, we apply the extractor in [Rao06, BRSW06] to all the row j's in these sources. Since each row is a (m, k')-source with $k' > \sqrt{m}$, we only need a constant number of sources to extract uniform random bits. Moreover, conditioned on the fixing of all row j's, for any $l \in S, l \neq j$, all the row l's are still independent (m, k')-sources. Thus the output of row l is uniform and independent of the output of row j. Thus now we obtain an SR-source such that a large fraction of the good rows are pair-wise independent.

However, there is still another problem. The problem is that the non-malleable condenser in [Li12b] has seed length $d = O(\log^2 n)$ which will make the initial $N = n^{O(\log n)}$. Thus this only gives us a quasi-polynomial time algorithm. To fix this, we note that the non-malleable condenser $\mathsf{nmCond}(X, R)$ in [Li12b] uses a seed $R = (R_1, R_2)$ and outputs $Z = (Y_1, Y_2)$. For a different seed $R' = (R'_1, R'_2)$, Y_1 takes care of the case where $R_1 = R'_1$ and Y_2 takes care of the case where $R_1 \neq R'_1$. The reason the seed has length $d = O(\log^2 n)$ is that Y_2 is an encoding of R_1 using some random variable produced by X and R_2 with an alternating extraction protocol, which requires R_2 to be a uniform string with size at least $\log^2 n$. In our case, however, we have the advantage of a supply of independent sources, whereas in the non-malleable condenser case we only have one weak source. Thus, we will use another weak source to provide the entropy used in the alternating extraction protocol. Specifically, we take 4 independent sources (X_1, X_2, X_3, X_4) and a seed $R = (R_1, R_2)$ such that $|R_1| = d = O(\log n)$ and $|R_2| = 10d$. We use (X_1, R_1, R_2) to produce Y_1. To produce Y_2, we first compute $W_2 = \mathsf{Raz}(R_2, X_2)$, where Raz is a two-source extractor in [Raz05] which works as long as R_2 has entropy rate $> 1/2$. This is because in the analysis fixing R'_1 may cause R_2 to lose entropy, but since $|R_2| = 10|R_1|$ conditioned on this fixing R_2 still has entropy rate roughly $9/10$. We then compute $W_3 = \mathsf{Ext}(X_3, W_2)$ so that W_3 is uniform and has size close to k. Now we can use X_4 and W_3 to perform the alternating extraction protocol to produce Y_2. As long as $k > \log^2 n$ this will satisfy our requirement, while ensuring that the seed length $|R| = O(\log n)$.

Now we are almost done, except for one remaining small problem. The problem is that in the above analysis we ignored all the error. However, it turns out that the error we achieved in the above process $\epsilon = 1/\text{poly}(n)$ is not small enough for the condenser to work. Note that if the good rows are indeed pair-wise independent then we wouldn't have any cross terms when computing the variance in Chebysev's inequality. However since they are only close to being pair-wise independent we will have roughly $N^2 = \text{poly}(n)$ cross terms, and each is bounded by roughly $O(\epsilon)$. It turns out the N here is too big for ϵ. Thus we need a smaller error. To fix this, we take several independent sources and from them obtain c' independent copies of the SR-sources we described above, each with N rows. We now compute the xor of these sources. Note that the aligned good rows in all these sources still take up a large fraction of $[N]$. On the other hand, since these sources are independent, after the xor a pair of aligned good rows will be $\epsilon^{c'}$-close to uniform. We show that we only need a constant c' to achieve a small enough error $\epsilon^{c'}$ for our condenser, and the error suffices for all subsequent condensing steps.

This gives our whole extractor construction. Thus, we use a constant number of independent sources to prepare the initial SR-source for the condenser. Then we use $O(\log(\frac{\log n}{\log k})) + O(1)$ sources to reduce the number of rows to k^5, and finally we use another constant number of independent sources to extract random bits. The dominating error comes from the last step where we apply the condenser (the lightest bin protocol), which is $1/\text{poly}(k)$. By choosing the number of rows where we stop condensing to be k^C, we can make the error k^{-C} for any constant $C > 1$.

By observing that the condenser works for two independent block sources, we can extend our extractor to work for a constant number of independent (n, k) sources (which are used to prepare the initial SR-source) and another 2 independent k-block sources with $O(\log(\frac{\log n}{\log k})) + O(1)$ blocks.

2.2 Network extractor protocol

In [KLRZ08], the authors showed that if we have a C-source extractor for (n, k) sources with output length $\Omega(k)$ and error ϵ, then there is an explicit r-round $(t, p - 1.1(r+1)t, \epsilon + 2^{-k^{\Omega(1)}})$ network extractor protocol for (n, k) sources, where $r = \left\lceil \frac{\log \mathsf{C}}{\log \log k} \right\rceil + 1$. By plugging in our independent source extractor, we obtain the improved network extractor protocol and thus the improved results for distributed computing with weak random sources.

Organization. After some preliminaries, we define alternating extraction in Section 4. We give our independent source extractor in Section 5, and the applications in network extractor protocols in Section 6. Finally we conclude with some open problems in Section 7.

3 Preliminaries

We often use capital letters for random variables and corresponding small letters for their instantiations. Let $|S|$ denote the cardinality of the set S. All logarithms are to the base 2.

3.1 Probability distributions

Definition 3.1 (statistical distance)**.** Let W and Z be two distributions on a set S. Their *statistical distance* (variation distance) is

$$\Delta(W, Z) \stackrel{def}{=} \max_{T \subseteq S}(|W(T) - Z(T)|) = \frac{1}{2} \sum_{s \in S} |W(s) - Z(s)|.$$

We say W is ε-close to Z, denoted $W \approx_\varepsilon Z$, if $\Delta(W, Z) \leq \varepsilon$. For a distribution D on a set S and a function $h : S \to T$, let $h(D)$ denote the distribution on T induced by choosing x according to D and outputting $h(x)$.

3.2 Somewhere Random Sources, Extractors and Condensers

Definition 3.2 (Somewhere Random sources)**.** A source $X = (X_1, \cdots, X_t)$ is $(t \times r)$ *somewhere-random* (SR-source for short) if each X_i takes values in $\{0,1\}^r$ and there is an i such that X_i is uniformly distributed.

Definition 3.3. (Block Sources) A distribution $X = X_1 \circ X_2 \circ \cdots, \circ X_t$ is called a $(k_1, k_2, \cdots, k_t)$ block source if for all $i = 1, \cdots, t$, we have that for all $x_1 \in \mathsf{Supp}(X_1), \cdots, x_{i-1} \in \mathsf{Supp}(X_{i-1})$, $H_\infty(X_i | X_1 = x_1, \cdots, X_{i-1} = x_{i-1}) \geq k_i$, i.e., each block has high min-entropy even conditioned on any fixing of the previous blocks. If $k_1 = k_2 = \cdots = k_t = k$, we say that X is a k block source.

Definition 3.4. A function $\mathsf{TExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ is a *strong two source extractor* for min-entropy k_1, k_2 and error ϵ if for every independent (n_1, k_1) source X and (n_2, k_2) source Y,

$$|(\mathsf{TExt}(X,Y),X) - (U_m,X)| < \epsilon$$

and

$$|(\mathsf{TExt}(X,Y),Y) - (U_m,Y)| < \epsilon,$$

where U_m is the uniform distribution on m bits independent of (X,Y).

3.3 Average conditional min-entropy

Definition 3.5. The *average conditional min-entropy* is defined as

$$\widetilde{H}_\infty(X|W) = -\log\left(\mathrm{E}_{w\leftarrow W}\left[\max_x \Pr[X=x|W=w]\right]\right) = -\log\left(\mathrm{E}_{w\leftarrow W}\left[2^{-H_\infty(X|W=w)}\right]\right).$$

Lemma 3.6 ([DORS08])**.** *For any $s>0$, $\Pr_{w\leftarrow W}[H_\infty(X|W=w) \geq \widetilde{H}_\infty(X|W) - s] \geq 1-2^{-s}$.*

Lemma 3.7 ([DORS08])**.** *If a random variable B has at most 2^ℓ possible values, then $\widetilde{H}_\infty(A|B) \geq H_\infty(A) - \ell$.*

3.4 Prerequisites from previous work

Sometimes it is convenient to talk about average case seeded extractors, where the source X has average conditional min-entropy $\widetilde{H}_\infty(X|Z) \geq k$ and the output of the extractor should be uniform given Z as well. The following lemma is proved in [DORS08].

Lemma 3.8. *[DORS08] For any $\delta > 0$, if* Ext *is a (k,ϵ) extractor then it is also a $(k+\log(1/\delta), \epsilon+\delta)$ average case extractor.*

For a strong seeded extractor with optimal parameters, we use the following extractor constructed in [GUV09].

Theorem 3.9 ([GUV09])**.** *For every constant $\alpha > 0$, and all positive integers n,k and any $\epsilon > 0$, there is an explicit construction of a strong (k,ϵ)-extractor* Ext : $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\log n + \log(1/\epsilon))$ and $m \geq (1-\alpha)k$. It is also a strong (k,ϵ) average case extractor with $m \geq (1-\alpha)k - O(\log n + \log(1/\epsilon))$.*

We need the following construction of strong two-source extractors in [Raz05].

Theorem 3.10 ([Raz05])**.** *For any n_1, n_2, k_1, k_2, m and any $0 < \delta < 1/2$ with*

- $n_1 \geq 6\log n_1 + 2\log n_2$
- $k_1 \geq (0.5+\delta)n_1 + 3\log n_1 + \log n_2$
- $k_2 \geq 5\log(n_1 - k_1)$
- $m \leq \delta \min[n_1/8, k_2/40] - 1$

There is a polynomial time computable strong 2-source extractor Raz : $\{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *for min-entropy k_1, k_2 with error $2^{-1.5m}$.*

Theorem 3.11 ([Rao06, BRSW06]). *There exist constants $c > 0$ and c' such that for every n, k with $k = k(n) = \Omega(\log^4 n)$ there exists a polynomial time computable function* $\mathsf{MExt} : (\{0,1\}^n)^u \to \{0,1\}^m$ *with $m = \Omega(k)$ and $u \leq c' \frac{\log n}{\log k}$ s.t. if $X^1, X^2, \ldots, X^u$ are independent (n, k) sources then*

$$|\mathsf{MExt}(X^1, \ldots, X^u) - U_m| < 2^{-k^c}.$$

Moreover, MExt *is a strong extractor.*

Theorem 3.12 ([BRSW06]). *There exist constants $c > 0$ and c' such that for every n, k, ℓ with $k = k(n) > \log^{10} n$ and $\ell \leq \mathrm{poly}(n)$ there exists a polynomial time computable function* $\mathsf{SRExt} : \{0,1\}^{\ell k} \times \{0,1\}^{un} \to \{0,1\}^m$ *with $m = \Omega(k)$ and $u \leq c' \frac{\log \ell}{\log k}$ s.t. if $X = X^1 \circ X^2 \circ \cdots \circ X^u$ is a $(k, \cdots, k)$ block sources and Y is an independent $\ell \times k$ SR-source then*

$$|\mathsf{SRExt}(Y, X) - U_m| < 2^{-k^c}.$$

Moreover, SRExt *is a strong extractor.*

Theorem 3.13. *[DLWZ11, CRS12, Li12a] For every constant $\delta > 0$, there exists a constant $\beta > 0$ such that for every $n, k \in \mathbb{N}$ with $k \geq (1/2 + \delta)n$ and $\epsilon > 2^{-\beta n}$ there exists an explicit (k, ϵ) non-malleable extractor with seed length $d = O(\log n + \log \epsilon^{-1})$ and output length $m = \Omega(n)$.*

The following standard lemma about conditional min-entropy is implicit in [NZ96] and explicit in [MW97].

Lemma 3.14 ([MW97]). *Let X and Y be random variables and let $\mathcal{Y}$ denote the range of Y. Then for all $\epsilon > 0$, one has*

$$\Pr_Y \left[H_\infty(X|Y = y) \geq H_\infty(X) - \log |\mathcal{Y}| - \log \left(\frac{1}{\epsilon} \right) \right] \geq 1 - \epsilon.$$

We also need the following lemma.

Lemma 3.15. *[Li12b] Let X and Y be random variables and let $\mathcal{Y}$ denote the range of Y. Assume that X is ϵ-close to having min-entropy k. Then for any $\epsilon' > 0$*

$$\Pr_Y \left[(X|Y = y) \text{ is } \epsilon'\text{-close to a source with min-entropy } k - \log |\mathcal{Y}| - \log \left(\frac{1}{\epsilon'} \right) \right] \geq 1 - \epsilon' - \frac{\epsilon}{\epsilon'}.$$

Lemma 3.16. *[BIW04] Assume that $Y_1, Y_2, \cdots, Y_t$ are independent random variables over $\{0,1\}^n$ such that for any $i, 1 \leq i \leq t$, we have $|Y_i - U_n| \leq \epsilon$. Let $Z = \oplus_{i=1}^t Y_i$. Then $|Z - U_n| \leq \epsilon^t$.*

4 Alternating Extraction

An important ingredient in our construction is the following alternating extraction protocol.

Alternating Extraction. Assume that we have two parties, Quentin and Wendy. Quentin has a source Q, Wendy has a source X. Also assume that Quentin has a uniform random seed S_1 (which may be correlated with Q). Suppose that (Q, S_1) is kept secret from Wendy and X is kept secret from Quentin. Let Ext_q, Ext_w be strong seeded extractors with optimal parameters, such as

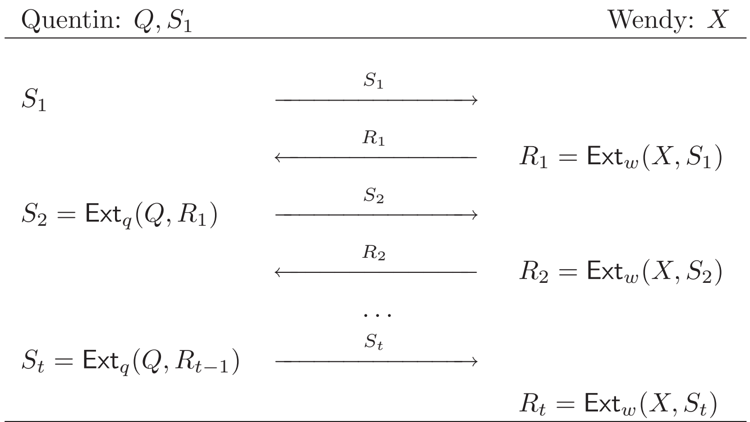

Figure 1: Alternating Extraction.

that in Theorem 3.9. Let ℓ be an integer parameter for the protocol. For some integer parameter $t > 0$, the *alternating extraction protocol* is an interactive process between Quentin and Wendy that runs in t steps.

In the first step, Quentin sends S_1 to Wendy, Wendy computes $R_1 = \mathsf{Ext}_w(X, S_1)$. She sends R_1 to Quentin and Quentin computes $S_2 = \mathsf{Ext}_q(Q, R_1)$. In this step R_1, S_2 each outputs ℓ bits. In each subsequent step i, Quentin sends S_i to Wendy, Wendy computes $R_i = \mathsf{Ext}_w(X, S_i)$. She replies R_i to Quentin and Quentin computes $S_{i+1} = \mathsf{Ext}_q(Q, R_i)$. In step i, R_i, S_{i+1} each outputs ℓ bits. Therefore, this process produces the following sequence:

$$S_1, R_1 = \mathsf{Ext}_w(X, S_1), \cdots, S_t = \mathsf{Ext}_q(Q, R_{t-1}), R_t = \mathsf{Ext}_w(X, S_t).$$

Look-Ahead Extractor. Now we can define our look-ahead extractor. Let $Y = (Q, S_1)$ be a seed, the look-ahead extractor is defined as

$$\mathsf{laExt}(X, Y) = \mathsf{laExt}(X, (Q, S_1)) \stackrel{def}{=} R_1, \cdots, R_t.$$

We first prove the following lemma.

Lemma 4.1. *Let $Y = (Q, S_1)$ where Q is an (n_q, k_q) source and S_1 is the uniform distribution over ℓ bits. Let $Y' = (Q', S'_1)$ be another random variable on the same support of Y that is arbitrarily correlated to Y. Assume X is an (n, k) source independent of (Y, Y'). Assume that Ext_q and Ext_w are strong seeded extractors that use ℓ bits to extract from $(n_q, k_q - 2t\ell)$ sources and $(n, k - 2t\ell)$ sources respectively, with error ϵ and $\ell = O(\log(max\{n_q, n\}) + \log(1/\epsilon))$. Let $(R_1, \cdots, R_t) = \mathsf{laExt}(X, Y)$ and $(R'_1, \cdots, R'_t) = \mathsf{laExt}(X, Y')$. Then for any $0 \le i \le t-1$, we have*

$$(Y, Y', [R'_1, \cdots, R'_i], [R_{i+1}, \cdots, R_t]) \approx_{\epsilon_1} (Y, Y', [R'_1, \cdots, R'_i], U_{\ell(t-i)}),$$

where $\epsilon_1 = O(t^2\epsilon)$.

Proof. Let $\{S'_i\}$ denote the random variables corresponding to $\{S_i\}$ that are produced in $\mathsf{laExt}(X, Y')$. For any $i, 0 \le i \le t-1$, let $\bar{S}_i = (S_0, \cdots, S_i)$, $\bar{S}'_i = (S'_0, \cdots, S'_i)$, $\bar{R}_i = (R_0, \cdots, R_i)$ and $\bar{R}'_i = (R'_0, \cdots, R'_i)$. We first prove the following claim.

Claim 4.2. *For any i, we have that*

$$(R_i, \bar{S}_{i-1}, \bar{S}'_{i-1}, \bar{R}_{i-1}, \bar{R}'_{i-1}, S_i, S'_i, Y, Y') \approx_{(2i-1)\epsilon} (U_\ell, \bar{S}_{i-1}, \bar{S}'_{i-1}, \bar{R}_{i-1}, \bar{R}'_{i-1}, S_i, S'_i, Y, Y')$$

and

$$(S_{i+1}, \bar{S}_i, \bar{S}'_i, \bar{R}_i, \bar{R}'_i) \approx_{(2i)\epsilon} (U_\ell, \bar{S}_i, \bar{S}'_i, \bar{R}_i, \bar{R}'_i).$$

Moreover, conditioned on $(\bar{S}_{i-1}, \bar{S}'_{i-1}, \bar{R}_{i-1}, \bar{R}'_{i-1}, S_i, S'_i)$, (R_i, R'_i) are both deterministic functions of X and the average conditional min-entropy of Q is at least $k_q - 2i\ell$; conditioned on $(\bar{S}_i, \bar{S}'_i, \bar{R}_i, \bar{R}'_i)$, $(Q, Q', S_{i+1}, S'_{i+1})$ is independent of X and the average conditional min-entropy of X is at least $k - 2i\ell$.

We prove the claim by induction on i. When $i = 0$, the statement is trivially true. Now we assume that the statements hold for $i = j$ and we prove them for $i = j + 1$.

We first fix $(\bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j)$. Note that now $(Q, Q', S_{j+1}, S'_{j+1})$ is independent of X. Moreover S_{j+1} is $(2j)\epsilon$-close to uniform. Since the average conditional min-entropy of X is at least $k - 2j\ell \geq k - 2t\ell$, By Theorem 3.9 we have that

$$(R_{j+1}, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}) \approx_{(2j+1)\epsilon} (U_\ell, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}).$$

Since $(Q, Q', S_{j+1}, S'_{j+1})$ is independent of X, we also have

$$(R_{j+1}, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}, Y, Y') \approx_{(2j+1)\epsilon} (U_\ell, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}, Y, Y').$$

Moreover, conditioned on $(\bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1})$, (R_{j+1}, R'_{j+1}) are both deterministic functions of X, and the average conditional min-entropy of Q is at least $k_q - 2j\ell - 2\ell = k_q - 2(j+1)\ell$.

Next, since conditioned on $(\bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1})$, (R_{j+1}, R'_{j+1}) are both deterministic functions of X, they are independent of (Q, Q'). Moreover R_{j+1} is $(2j+1)\epsilon$-close to uniform. Since the average conditional min-entropy of Q is at least $k_q - 2(j+1)\ell \geq k_q - 2t\ell$, By Theorem 3.9 we have that

$$\begin{aligned}&(S_{j+2}, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}, R_{j+1}, R'_{j+1})\\ \approx_{(2j+2)\epsilon}&(U_\ell, \bar{S}_j, \bar{S}'_j, \bar{R}_j, \bar{R}'_j, S_{j+1}, S'_{j+1}, R_{j+1}, R'_{j+1}).\end{aligned}$$

Namely,

$$(S_{j+2}, \overline{S_{j+1}}, \overline{S'_{j+1}}, \overline{R_{j+1}}, \overline{R'_{j+1}}) \approx_{(2(j+1))\epsilon} (U_\ell, \overline{S_{j+1}}, \overline{S'_{j+1}}, \overline{R_{j+1}}, \overline{R'_{j+1}}).$$

Moreover, conditioned on $(\overline{S_{j+1}}, \overline{S'_{j+1}}, \overline{R_{j+1}}, \overline{R'_{j+1}})$, $(Q, Q', S_{j+2}, S'_{j+2})$ is independent of X since S_{j+2} and S'_{j+2} are deterministic functions of Q and Q' respectively. Also note that now the average conditional min-entropy of X is at least $k - 2j\ell - 2\ell = k - 2(j+1)\ell$.

Therefore, we have that for any i,

$$(R_i, \bar{R}_{i-1}, \bar{R}'_{i-1}, Y, Y') \approx_{(2i-1)\epsilon} (U_\ell, \bar{R}_{i-1}, \bar{R}'_{i-1}, Y, Y').$$

Thus for any i,

$$(Y, Y', [R'_1, \cdots, R'_i], [R_{i+1}, \cdots, R_t]) \approx_{\epsilon_1} (Y, Y', [R'_1, \cdots, R'_i], U_{\ell(t-i)}),$$

where $\epsilon_1 = \sum_{j=i+1}^{t}((2j-1)\epsilon) = O(t^2\epsilon)$. □

Next, we need the following definitions and constructions from [DW09].

Definition 4.3. [DW09] Given $S_1, S_2 \subseteq \{1, \cdots, t\}$, we say that the ordered pair (S_1, S_2) is top-heavy if there is some integer j such that $|S_1^{\geq j}| > |S_2^{\geq j}|$, where $S^{\geq j} \stackrel{def}{=} \{s \in S | s \geq j\}$. Note that it is possible that (S_1, S_2) and (S_2, S_1) are both top-heavy. For a collection Ψ of sets $S_i \subseteq \{1, \cdots, t\}$, we say that Ψ is pairwise top-heavy if every ordered pair (S_i, S_j) of sets $S_i, S_j \in \Psi$ with $i \neq j$, is top-heavy.

Now, for any m-bit message $\mu = (b_1, \cdots, b_m)$, consider the following mapping of μ to a subset $S \subseteq \{1, \cdots, 4m\}$:

$$f(\mu) = f(b_1, \cdots, b_m) = \{4i - 3 + b_i, 4i - b_i | i = 1, \cdots, m\}$$

i.e., each bit b_i decides if to include $\{4i-3, 4i\}$ (if $b_i = 0$) or $\{4i-2, 4i-1\}$ (if $b_i = 1$) in S.

We now have the following lemma.

Lemma 4.4. *[DW09] The above construction gives a pairwise top-heavy collection Ψ of 2^m sets $S \subseteq \{1, \cdots, t\}$ where $t = 4m$. Furthermore, the function f is an efficient mapping of $\mu \in \{0,1\}^m$ to S_μ.*

Now we have the following construction.

Let $r \in (\{0,1\}^\ell)^t$ be the output of the look-ahead extractor defined above, i.e., $r = (r_1, \cdots, r_t) = \mathsf{laExt}(X, (Q, S_1))$. Let $\Psi = \{S_1, \cdots, S_{2^m}\}$ be the pairwise top-heavy collection of sets constructed above. For any string $\mu \in \{0,1\}^m$, define the function $\mathsf{laMAC}_r(\mu) \stackrel{def}{=} [r_i | i \in S_\mu]$, indexed by r.

5 Independent Source Extractor

In this section we present our construction of independent source extractor. Let Ext, nmExt, Raz, MExt be the extractors in theorem 3.9, theorem 3.13, theorem 3.10 and theorem 3.11 respectively. Let laExt and laMAC be the look-ahead extractor and the function defined above. We first show how to use a constant number of independent (n, k) sources with $k \geq \log^4 n$ to obtain a somewhere random source such that there exists a large fraction of rows where each pair is close to being independent and uniform.

Let X_1, X_2, X_3, X_4 be 4 independent (n, k) sources. Let $r = (r_1, r_2)$ be a string such that r_1 has length d and r_2 has length $20d$ where $d = O(\log n)$ is the seed length that guarantees error $\epsilon = 1/\mathrm{poly}(n)$ in theorem 3.9. For every $r \in \{0,1\}^{21d}$, do the following and obtain a source Y^r.

1. Let $W_1 = \mathsf{Ext}(X_1, r_1)$ and $Y_1 = \mathsf{nmExt}(W_1, r_2)$ such that Y_1 has $2d\sqrt{k}$ bits.

2. Let $W_2 = \mathsf{Raz}(r_2, X_2)$, $W_3 = \mathsf{Ext}(X_3, W_2)$ and $W_4 = \mathsf{laExt}(X_4, W_3)$ with $t = 4d$ and $\ell = 2\sqrt{k}$, where W_3 is viewed as Q and S_1 is the prefix of W_3 with d bits.

3. Let $Y_2 = \mathsf{laMAC}_{W_4}(r_1)$ and $Y^r = Y_1 \circ Y_2$.

Now assume that $R \in \{0,1\}^{21d}$ is a uniform random seed independent of X_1, X_2, X_3, X_4. Let $\mathcal{A} : \{0,1\}^{21d} \to \{0,1\}^{21d}$ be any deterministic function such that $\forall r \in \{0,1\}^{21d}, \mathcal{A}(r) \neq r$. Let Y^r be the source obtained with r and $Y^{\mathcal{A}(r)}$ be the source obtained with $\mathcal{A}(r)$. We have the following lemma.

Lemma 5.1. *For some $\epsilon = 1/\text{poly}(n)$, with probability $1-\epsilon$ over the fixing of $R = r$, we have*

$$\Pr_{y' \leftarrow Y^{\mathcal{A}(r)}}[Y^r|_{Y^{\mathcal{A}(r)}=y'} \textit{ is } \epsilon - \textit{close to a } (10d\sqrt{k}, \sqrt{k}) \textit{ source}] \geq 1-\epsilon.$$

Proof. Let $r' = \mathcal{A}(r) = (r'_1, r'_2)$. Since $r' \neq r$, we have two cases: $r'_1 = r_1$ or $r'_1 \neq r_1$. We call a string $r \in \{0,1\}^{21d}$ bad if conditioned on $R = r$,

$$\Pr_{y' \leftarrow Y^{\mathcal{A}(r)}}[Y^r|_{Y^{\mathcal{A}(r)}=y'} \text{ is } \epsilon - \text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] < 1-\epsilon.$$

Now we consider two other deterministic functions $\mathcal{A}_1 : \{0,1\}^{21d} \to \{0,1\}^{21d}$ and $\mathcal{A}_2 : \{0,1\}^{21d} \to \{0,1\}^{21d}$. For all the r's such that $r'_1 = r_1$, we let $\mathcal{A}_1(r) = r'$. For all the other r's we let $\mathcal{A}_1(r)_1 = r_1$ and choose $\mathcal{A}_1(r)_2$ arbitrarily but such that $\mathcal{A}_1(r)_2 \neq r_2$. For all the r's such that $r'_1 \neq r_1$, we let $\mathcal{A}_2(r) = r'$. For all the other r's we choose $\mathcal{A}_2(r)$ arbitrarily but such that $\mathcal{A}_2(r)_1 \neq r_1$. Thus for any r, we have $\mathcal{A}_1(r)_1 = r_1$ and $\mathcal{A}_2(r)_1 \neq r_1$. Note that any bad r in $\mathcal{A}$ is either a bad r in $\mathcal{A}_1$ or $\mathcal{A}_2$. Thus the number of bad r's in $\mathcal{A}$ is at most the sum of the numbers of bad r's in $\mathcal{A}_1$ and $\mathcal{A}_2$.

First consider $\mathcal{A}_1$. We slightly abuse notation and let $R' = \mathcal{A}_1(R) = (R'_1, R'_2)$. Note that $R'_1 = R_1$. We now fix $R_1 = r_1$. By theorem 3.9, with probability $1-\epsilon_1$ over this fixing, $W_1 = \mathsf{Ext}(X_1, r_1)$ is ϵ_1-close to uniform, where $\epsilon_1 = 1/\text{poly}(n)$. Note that after we fix $R_1 = r_1$, R_2 is still uniform and R'_2 is now a deterministic function of R_2 with $R'_2 \neq R_2$. Thus when W_1 is ϵ_1-close to uniform, by theorem 3.13,

$$(Y_1, Y'_1, R_2) \approx_{\epsilon_1 + 1/\text{poly}(n)} (U_{2d\sqrt{k}}, Y'_1, R_2),$$

where $Y'_1 = \mathsf{nmExt}(W_1, R'_2)$.

Thus with probability $1-\epsilon_2$ over the further fixing of $R_2 = r_2$, we have

$$\Pr_{y' \leftarrow Y'_1}[Y_1|_{Y'_1=y'} \approx_{\epsilon_2} U_{2d\sqrt{k}}] \geq 1-\epsilon_2,$$

where $\epsilon_2 = 1/\text{poly}(n)$. Note that once r is fixed, (Y_2, Y'_2) is a deterministic function of (X_2, X_3, X_4) and thus is independent of Y_1. Therefore we can further condition on Y'_2 and Y_1 is still close to uniform, and thus Y^r is close to a source with min-entropy $2d\sqrt{k} > \sqrt{k}$ conditioned on $Y^{r'}$. Thus the fraction of bad r's in $\mathcal{A}_1$ is at most $\epsilon_1 + \epsilon_2 = 1/\text{poly}(n)$.

Next, consider $\mathcal{A}_2$. We slightly abuse notation and let $R' = \mathcal{A}_2(R) = (R'_1, R'_2)$. We will use letters with prime to denote the random variables produced with R'. Now we have $R'_1 \neq R_1$. We first fix $R_1 = r_1$ and $R'_1 = r'_1$. After R_1 is fixed, R_2 is still uniform. However, fixing R'_1 may cause R_2 to lose entropy. By lemma 3.14, with probability $1-\epsilon_3$ over this fixing, R_2 has min-entropy $20d - d - d = 18d$, where $\epsilon_3 = 1/2^d = 1/\text{poly}(n)$. Note that after this fixing, R'_2 is a deterministic function of R_2. Thus by theorem 3.10 we can output d bits in W_2 and we have that

$$(W_2, R_2, R'_2) \approx_{1/\text{poly}(n)} (U_d, R_2, R'_2).$$

Thus we can further fix $R_2 = r_2$ (and also $R'_2 = r'_2$) and with probability $1 - \epsilon_4 = 1 - 1/\mathrm{poly}(n)$ over this fixing, W_2 is $1/\mathrm{poly}(n)$-close to uniform. Note that now we have fixed R and R', and W_2 is a deterministic function of X_2. Note also that W'_2 is correlated with W_2. When W_2 is $1/\mathrm{poly}(n)$-close to uniform, by theorem 3.9 we can output $0.9k$ bits in W_3 and we have that W_3 is $1/\mathrm{poly}(n)$-close to uniform. Note that W'_3 is correlated with W_3. However, (W_3, W'_3) is independent of X_4. Let $W_4 = \mathsf{laExt}(X_4, W_3) = (V_1, \cdots, V_t)$ and $W'_4 = \mathsf{laExt}(X_4, W'_3) = (V'_1, \cdots, V'_t)$. By lemma 4.1 (and notice that $2t\sqrt{k} << k$, $d << k$), we have that for any $0 \le i \le t-1$,

$$([V'_1, \cdots, V'_i], [V_{i+1}, \cdots, V_t]) \approx_{\epsilon_5} ([V'_1, \cdots, V'_i], U_{2(t-i)\sqrt{k}}),$$

where $\epsilon_5 = 1/\mathrm{poly}(n) + O(t^2 2^{-\Omega(\sqrt{k})}\mathrm{poly}(n)) = 1/\mathrm{poly}(n)$.

Now note that $Y_2 = \mathsf{lrMAC}_{W_4}(r_1)$ and $Y'_2 = \mathsf{lrMAC}_{W'_4}(r'_1)$. Let the two sets in Lemma 4.4 that correspond to r_1 and r'_1 be H and H' respectively. Since $r_1 \ne r'_1$, by Lemma 4.4 there exists $j \in [4d]$ such that $|H^{\ge j}| > |H'^{\ge j}|$. Let $l = |H^{\ge j}|$. Thus $l \le t = 4d$ and $|H'^{\ge j}| \le l - 1$. Let V_H be the concatenation of $\{V_i, i \in H^{\ge j}\}$ and $V'_{H'}$ be the concatenation of $\{V'_i, i \in H'^{\ge j}\}$. By the above equation we have that

$$([V'_1, \cdots, V'_{j-1}], V_H) \approx_{\epsilon_5} ([V'_1, \cdots, V'_{j-1}], U_{2l\sqrt{k}}).$$

Thus with probability $1 - \sqrt[3]{\epsilon_5}$ over the fixings of $(V'_1, \cdots, V'_{j-1})$, V_H is $\sqrt[3]{\epsilon_5^2}$-close to $U_{2l\sqrt{k}}$.

Since the size of $V'_{H'}$ is at most $2(l-1)\sqrt{k}$, we can further fix $V'_{H'}$ and by lemma 3.15 we have that with probability $1 - 2\sqrt[3]{\epsilon_5}$ over this fixing, V_H is $\sqrt[3]{\epsilon_5}$-close to a source with min-entropy $2\sqrt{k} - O(\log n) > \sqrt{k}$. Note that Y'_2 is fixed when both $(V'_1, \cdots, V'_{j-1})$ and $V'_{H'}$ are fixed. Thus we have shown that

$$\Pr_{y' \leftarrow Y'_2}[Y_2|_{Y'_2 = y'} \text{ is } \sqrt[3]{\epsilon_5} - \text{close to a } \sqrt{k}\text{-source}] \ge 1 - 3\sqrt[3]{\epsilon_5}.$$

Finally, note that we have already fixed (R, R') before. After this fixing, (Y_2, Y'_2) is a deterministic function of (X_2, X_3, X_4), while (Y_1, Y'_1) is a deterministic function of X_1. Thus (Y_2, Y'_2) is independent of (Y_1, Y'_1). Therefore we can further fix Y'_1 and Y_2 is still close to a source with min-entropy $\sqrt{k}$. Thus Y^r is close to a source with min-entropy $\sqrt{k}$ conditioned on $Y^{r'}$. Note the fraction of bad r's in $\mathcal{A}_2$ is at most $\epsilon_3 + \epsilon_4 = 1/\mathrm{poly}(n)$. Now choose $\epsilon = max\{\epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4, 3\sqrt[3]{\epsilon_5}\} = 1/\mathrm{poly}(n)$ and the lemma is proved. □

We now have the following lemma.

Lemma 5.2. *For some $\epsilon = 1/\mathrm{poly}(n)$ there exists a subset $S \subset \{0,1\}^{21d}$ with $|S| \ge (1-\epsilon)2^{21d}$ such that for any $i, j \in S, i \ne j$, we have*

$$\Pr_{y \leftarrow Y^j}[Y^i|_{Y^j = y} \textit{ is } \epsilon - \textit{close to a } (10d\sqrt{k}, \sqrt{k}) \textit{ source}] \ge 1 - \epsilon.$$

Proof. Let ϵ be the same as in lemma 5.1. For any $i, j \in \{0,1\}^{21d}, i \ne j$, we say that j is bad for i if

$$\Pr_{y \leftarrow Y^j}[Y^i|_{Y^j = y} \text{ is } \epsilon - \text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] < 1 - \epsilon.$$

Let $B = \{i \in \{0,1\}^{21d} : \exists j \in \{0,1\}^{21d}, j \neq i \text{ and } j \text{ is bad for } i\}$. We claim that $|B| \leq \epsilon 2^{21d}$. Otherwise, we can construct a deterministic function $\mathcal{A} : \{0,1\}^{21d} \to \{0,1\}^{21d}$ as follows. For all $i \in B$, let $\mathcal{A}(i)$ be a bad j for i; for all the other $i \in \{0,1\}^{21d}$, define $\mathcal{A}(i)$ to be any $j \in \{0,1\}^{21d}$ such that $j \neq i$. Thus when R is uniformly sampled from $\{0,1\}^{21d}$, we have that with probability at least ϵ over the fixing of $R = r$,

$$\Pr_{y' \leftarrow Y^{\mathcal{A}(r)}}[Y^r|_{Y^{\mathcal{A}(r)}=y'} \text{ is } \epsilon-\text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] < 1 - \epsilon,$$

which contradicts lemma 5.1.

Now let $S = \{0,1\}^{21d} \setminus B$. We have that $|S| \geq (1-\epsilon)2^{21d}$ and for any $i, j \in S, i \neq j$,

$$\Pr_{y \leftarrow Y^j}[Y^i|_{Y^j=y} \text{ is } \epsilon-\text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] \geq 1 - \epsilon.$$

□

Now let c be the number of independent $(10d\sqrt{k}, \sqrt{k})$ sources that are needed to extract $m = \Omega(\sqrt{k})$ bits with error $\epsilon' = 2^{-k^{\Omega(1)}} \leq 1/\text{poly}(n)$, as in theorem 3.11. Note that since $k \geq \log^4 n$ we have $\sqrt{k} = \Omega(d^2)$. Thus $c = O(\log(10d\sqrt{k})/\log(\sqrt{k}))$ is an absolute constant. We now take $4c$ independent (n,k) sources $X_1, X_2, \cdots, X_{4c}$ and divide them equally into c sets $\{X_1, X_2, X_3, X_4\}, \cdots, \{X_{4c-3}, X_{4c-2}, X_{4c-1}, X_{4c}\}$. For each set we use the procedure described above to get $2^{21d} = \text{poly}(n)$ number of sources $\{Y_i^r\}$, for $r \in \{0,1\}^{21d}$ and $i \in [c]$. Now for any $r \in \{0,1\}^{21d}$, let $Z^r = \mathsf{MExt}(Y_1^r, \cdots, Y_c^r)$. Thus we obtain $2^{21d} = \text{poly}(n)$ number of sources $\{Z^r\}$, for $r \in \{0,1\}^{21d}$. We now have the following lemma.

Lemma 5.3. *For some $\epsilon = 1/\text{poly}(n)$ there exists a subset $S \subset \{0,1\}^{21d}$ with $|S| \geq (1-\epsilon)2^{21d}$ such that for any $i, j \in S, i \neq j$, we have*

$$(Z^i, Z^j) \approx_\epsilon U_{2m},$$

where $m = \Omega(\sqrt{k})$.

Proof. Let ϵ_1 be the error in lemma 5.2. By lemma 5.2, for each $t \in [c]$, there exists a subset $S_t \subset \{0,1\}^{21d}$ with $|S_t| \geq (1-\epsilon_1)2^{21d}$ such that for any $i, j \in S_t, i \neq j$, we have

$$\Pr_{y \leftarrow Y_t^j}[Y_t^i|_{Y_t^j=y} \text{ is } \epsilon_1-\text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] \geq 1 - \epsilon_1.$$

Let $S = \cap_t S_t$. Then we have $|S| \geq (1 - c\epsilon_1)2^{21d}$ and for any $i, j \in S, i \neq j$, we have that for any $t \in [c]$,

$$\Pr_{y \leftarrow Y_t^j}[Y_t^i|_{Y_t^j=y} \text{ is } \epsilon_1-\text{close to a } (10d\sqrt{k}, \sqrt{k}) \text{ source}] \geq 1 - \epsilon_1.$$

By the above we know that $\forall t \in [c]$, Y_t^j is $2\epsilon_1$-close to a $(10d\sqrt{k}, \sqrt{k})$ source. Thus by theorem 3.11 we have

$$Z^j \approx_{2c\epsilon_1 + \epsilon'} U_m.$$

Next, we fix all $Y_t^j, t \in [c]$, and we have that with probability $1 - c\epsilon_1$ over this fixing, for any t, Y_t^i is ϵ_1-close to a $(10d\sqrt{k}, \sqrt{k})$ source. Note that after this fixing Y_t^i are still independent, thus by theorem 3.11 we have

$$Z^i \approx_{c\epsilon_1+\epsilon'} U_m.$$

Since we already fixed all $Y_t^j, t \in [c]$, this implies that

$$(Z^i, Z^j) \approx_{2c\epsilon_1+\epsilon'} (U_m, Z^j).$$

Thus we have

$$(Z^i, Z^j) \approx_{4c\epsilon_1+2\epsilon'} U_{2m}.$$

Let $\epsilon = 4c\epsilon_1 + 2\epsilon' = 1/\text{poly}(n)$, and the lemma is proved. □

We now describe the lightest bin protocol.

Lightest bin protocol: Assume there are N strings $\{z^i, i \in [N]\}$ where each $z_i \in \{0,1\}^m$ with $m > \log N$. The output of a lightest bin protocol with $r < N$ bins is a subset $T \subset [N]$ that is obtained as follows. Image that each string z^i is associated with a player P_i. Now, for each i, P_i uses the first $\log r$ bits of z_i to select a bin j, i.e., if the first $\log r$ bits of z_i is the binary expression of $j-1$, then P_i selects bin j. Now let bin l be the bin that is selected by the fewest number of players. Then

$$T = \{i \in [N] : P_i \text{ selects bin } l.\}$$

We now have the following lemma.

Lemma 5.4. *Assume that we have N sources $Z_1^i, i \in [N]$ over $m > 10\log(1/\epsilon)$ bits and a subset $S \subset [N]$ with $|S| \geq \alpha N$ for some constant $\alpha > 0$ such that for any $i, j \in S, i \neq j$,*

$$(Z_1^i, Z_1^j) \approx_\epsilon U_{2m}$$

with $\epsilon < 1/N^{12}$.

Let $Z_1 = Z_1^1 \circ \cdots \circ Z_1^N$. Run the lightest bin protocol with $N^{1/4}$ bins and let the output contain N_2 elements $\{i_1, i_2, \cdots, i_{N_2} \in [N]\}$. Assume that X is an (n,k) source independent of Z_1 with $k > 40\log(1/\epsilon)$. For any $j \in [N_2]$, let $Z_2^j = \mathsf{Ext}(X, Z_1^{i_j})$ where Ext is the strong seeded extractor in theorem 3.9 and output $m_2 = k/4$ bits. Then for any $\delta > N^{-1/2}$, with probability at least $1-3N^{1/2}/(\delta^2 s)-4N^{-1/2}$ over the fixing of Z_1, there exists a subset $S_2 \subset [N_2]$ with $|S_2| \geq \alpha(1-\delta)N_2$ such that for any $i, j \in S_2, i \neq j$,

$$(Z_2^i, Z_2^j) \approx_{\epsilon_2} U_{2m_2}$$

with $\epsilon_2 < 1/N_2^{12}$ and $m_2 > 10\log(1/\epsilon_2)$.

Proof. Note that the lightest bin contains at most $N^{3/4}$ elements. We first show that in the lightest bin protocol, with high probability every bin contains at least $(\alpha-\delta)N^{3/4}$ elements in S.

Consider a particular bin and consider the choices of the Z_1^i's with $i \in S$. Let $s = |S|$. Let V_i be the indicator variable of whether Z_1^i chooses this bin and let $V = \sum_{i \in S} V_i$. Let $p_i = \Pr[V_i = 1]$ and $q_i = \Pr[V_i = 0]$. Then we have

$$E[V] = \sum_{i \in S} E[V_i] = \sum_{i \in S} p_i.$$

We know for any $i \in S$, Z_1^i is ϵ-close to uniform. Thus $\Pr[V_i = 1] \geq N^{-1/4} - \epsilon$. Therefore

$$E[V] \geq (N^{-1/4} - \epsilon)s.$$

Note that

$$\begin{aligned}\Pr[V < N^{-1/4}(1-\delta)s] &\leq \Pr[|V - E[V]| > \delta N^{-1/4}s - \epsilon s] \\ &\leq \Pr[|V - E[V]| > 0.9\delta N^{-1/4}s],\end{aligned}$$

since $\epsilon s < 1$ and $\delta > N^{-1/2}$.

Thus by Chebysev's inequality we have

$$\Pr[V < N^{-1/4}(1-\delta)s] \leq \mathsf{Var}[V]/(0.81\delta^2 N^{-1/2}s^2) < 2N^{1/2}\mathsf{Var}[V]/(\delta^2 s^2).$$

We now compute $\mathsf{Var}[V]$. By definition

$$\begin{aligned}\mathsf{Var}[V] = E(V - E[V])^2 &= E\left(\sum_{i \in S}(V_i - E[V_i])\right)^2 \\ &= \sum_{i \in S}\mathsf{Var}[V_i] + \sum_{i,j \in [S], i \neq j} E[(V_i - E[V_i])(V_j - E[V_j])].\end{aligned}$$

For each $i \in S$, we have

$$\mathsf{Var}[V_i] = p_i q_i < p_i \leq N^{-1/4} + \epsilon.$$

Next, note that

$$E[(V_i - E[V_i])(V_j - E[V_j])] = E[V_i V_j] - E[V_i]E[V_j].$$

Since $(Z_1^i, Z_1^j) \approx_\epsilon U_{2m}$, we have $E[V_i V_j] = \Pr[V_i = 1, V_j = 1] \leq N^{-1/2} + \epsilon$, $E[V_i] \geq N^{-1/4} - \epsilon$ and $E[V_j] \geq N^{-1/4} - \epsilon$. Thus

$$E[V_i V_j] - E[V_i]E[V_j] \leq N^{-1/2} + \epsilon - (N^{-1/4} - \epsilon)^2 < (2N^{-1/4} + 1)\epsilon < 2\epsilon.$$

Thus

$$\mathsf{Var}[V] < (N^{-1/4} + \epsilon)s + 2s^2\epsilon < N^{-1/4}s + 3,$$

since $\epsilon < 1/N^{12}$. Therefore we have

$$\Pr[V < N^{-1/4}(1-\delta)s] < 2N^{1/2}(N^{-1/4}s + 3)/(\delta^2 s^2) < 3N^{1/4}/(\delta^2 s).$$

Thus by the union bound, we have that the probability that every bin contains at least $N^{-1/4}(1-\delta)s$ elements in S is at least $1 - 3N^{1/2}/(\delta^2 s)$. When this happens, let S_2 be the set of elements in S in the lightest bin. Then we have $|S_2| \geq N^{-1/4}(1-\delta)s \geq \alpha(1-\delta)N^{3/4} \geq \alpha(1-\delta)N_2$.

Next, we show that with high probability the new sources with index in S_2 are pair-wise close to uniform. For this, consider any $i, j \in [S], i \neq j$. Let $W^i = \mathsf{Ext}(X, Z_1^i)$ and $W^j = \mathsf{Ext}(X, Z_1^j)$. Note that $(Z_1^i, Z_1^j) \approx_\epsilon U_{2m}$. First assume that (Z_1^i, Z_1^j) is indeed uniform, then by theorem 3.9 we have

$$(W^i, Z_1^i) \approx_\epsilon (U_{m_2}, Z_1^i).$$

Now we fix Z_1^i and W^i. Note that after fixing Z_1^i, W^i is a deterministic function of X. Thus by lemma 3.14, with probability $1 - 2^{-k/4} > 1 - \epsilon$ over this fixing, X is an $(n, k - k/4 - k/4 = k/2)$ source. After this fixing, Z_1^j is still uniform and independent of X, thus again by theorem 3.9 we have

$$(W^j, Z_1^j) \approx_\epsilon (U_{m_2}, Z_1^j).$$

Therefore

$$(W^i, W^j, Z_1^i, Z_1^j) \approx_{3\epsilon} (U_{2m_2}, Z_1^i, Z_1^j).$$

Adding back the error where $(Z_1^i, Z_1^j) \approx_\epsilon U_{2m}$, we have

$$(W^i, W^j, Z_1^i, Z_1^j) \approx_{4\epsilon} (U_{2m_2}, Z_1^i, Z_1^j).$$

Therefore, with probability $1 - 4N^{-2.5}$ over the fixing of (Z_1^i, Z_1^j), (W^i, W^j) is $N^{2.5}\epsilon$-close to uniform. Thus by the union bound (and noticing that $s \leq N$), we have that with probability at least $1 - 4N^{-1/2}$ over the fixing of Z_1, for any $i, j \in [S], i \neq j$, (W^i, W^j) is $N^{2.5}\epsilon$-close to uniform. In particular, this implies that the new sources with index in S_2 are pair-wise close to uniform. Note that $N_2 \leq N^{3/4}$ and $\epsilon < 1/N^{12}$, thus $N^{2.5}\epsilon < 1/N_2^{12}$. Also note that $m_2 = k/4 > 10\log(1/\epsilon) > 10\log(1/\epsilon_2)$. By the union bound, the lemma is proved. □

Now we have the following construction.

Construction 5.5. Independent Source Extractor.
Let ϵ be the error in lemma 5.3 and let $N_1 = 2^{21d}$. Let c_1 be an integer constant such that $\epsilon^{c_1} < 1/N_1^{12}$. We first take $C = 4cc_1$ independent (n, k) sources and from them obtain c_1 SR-sources $Z_1', \cdots, Z_{c_1}'$ where each $Z_i' = Z_i'^1 \circ Z_i'^2 \circ \cdots \circ Z_i'^{N_1}$ contains N_1 rows, as in lemma 5.3. Let $Z_1 = \bigoplus_{i=1}^{c_1} Z_i'$. Set $t = 1$. While the number of rows in Z_t is bigger than $\ell = k^5$ we do the following:

1. Run the lightest bin protocol with Z_t and $r_t = N_t^{1/4}$ bins and let the output contain N_{t+1} elements $\{i_1, i_2, \cdots, i_{N_{t+1}} \in [N_t]\}$.

2. Take a fresh independent (n, k) source X_{C+t} and for any $j \in [N_{t+1}]$, let $Z_{t+1}^j = \mathsf{Ext}(X_{C+t}, Z_t^{i_j})$ where Ext is the strong seeded extractor in theorem 3.9 and output $m_2 = k/4$ bits.

3. Let $Z_{t+1} = Z_{t+1}^1 \circ \cdots \circ Z_{t+1}^{N_{t+1}}$. Set $t = t + 1$.

At the end of the iteration we get a source Z_t with at most $\ell = k^5$ rows. Let SRExt be the extractor in theorem 3.12, set up to extract from an $\ell \times \frac{k}{4}$ source and c_2 independent (n, k) sources $X_{C+t+1}, \cdots, X_{C+t+c_2}$ (note that independent sources are a special case of block sources). The final output is $W = \mathsf{SRExt}(Z_t, X_{C+t+1}, \cdots, X_{C+t+c_2})$.

Theorem 5.6. *The above construction is an extractor for* $O(\log(\frac{\log n}{\log k}))+O(1)$ *independent sources with error* $1/\mathrm{poly}(k)$.

Proof. We first show that the number of independent sources we use is $O(\log(\frac{\log n}{\log k}))+O(1)$. To see this, note that to obtain one Z_i' we use a constant $4c$ number of independent sources, and the error is $\epsilon = 1/\mathrm{poly}(n)$ as in lemma 5.3. Note that $N_1 = 2^{21d} = \mathrm{poly}(n)$, thus it suffices to take c_1 to be a constant. Next note that in the lightest bin protocol each time the number of rows decreases from N to at most $N^{3/4}$, thus it takes $t = O(\log(\frac{\log n}{\log k})) + O(1)$ number of independent sources to get the number of rows down to k^5. Finally note that $c_2 = O(\log(k^5)/\log k) = O(1)$. Thus the total number of independent sources used is $O(\log(\frac{\log n}{\log k})) + O(1)$.

Next, by lemma 5.3 we know that for each Z_i' there exists a subset $S_i \subset [N_1]$ with $|S_i| \geq (1-\epsilon)N_1$ such that any pair of rows in S_i is ϵ-close to uniform. Now let $S = \cap_{i=1}^{c_1} S_i$. Then we have that $|S| \geq (1 - c_1\epsilon)N_1$ and by lemma 3.16, for any $i, j \in S, i \neq j$,

$$(Z_1^i, Z_1^j) \approx_{\epsilon^{c_1}} U_{2m},$$

where $m = \Omega(\sqrt{k})$.

Note that $\epsilon^{c_1} < 1/N_1^{12}$ and $k > \log^4 n$. Thus Z_1 satisfies the conditions in lemma 5.4 with $\alpha = 1 - c_1\epsilon = 1 - 1/\mathrm{poly}(n)$. Now let $\delta = 1/(3t)$ in lemma 5.4 and consider the lightest bin protocol where we get $Z_1, \cdots, Z_t$ with each Z_i having N_i rows. By lemma 5.4 if the "good" event in the lemma always happens, then the "good" set S_i in each Z_i has size at least $s_i \geq \alpha(1-\delta)^t N_i > \alpha(1-\delta t)N_i > N_i/2$. Thus the probability of the "bad" event in lemma 5.4 is at most $3N_i^{1/2}/(\delta^2 s_i) + 4N_i^{-1/2} = O(t^2 N_i^{-1/2})$. Note that for any $i \leq t-1$, $N_i > k^5$. Thus the total error is at most

$$tO(t^2k^{-5/2}) = O(t^3k^{-5/2}) < 1/(10k^2).$$

Thus we have that with probability $1 - 1/(10k^2)$ over the fixings of all previous independent sources, Z_t is k^{-40}-close to (note that $N_t \geq s_t \geq (k^5)^{3/4}/2$) an SR-source with at most k^5 rows. Now by theorem 3.12, W is $2^{-k^{\Omega(1)}}$-close to uniform. Therefore, the total error of the output is at most $1/(10k^2) + 2^{-k^{\Omega(1)}} < 1/k^2$. ☐

Remark 5.7. The error in the extractor can be made $1/k^C$ for any constant $C > 1$, just by setting the number of rows in the final source Z_t to be an appropriate $\mathrm{poly}(k)$. The time of the algorithm can be larger (but still polynomial in n), and the number of sources needed is still $O(\log(\frac{\log n}{\log k})) + O(1)$.

Remark 5.8. When the entropy k is smaller, we can get better error dependence on k. Specifically, we can set the number of rows in the final source Z_t to be $k^{\Omega(\log(\frac{\log n}{\log k})+1)}$. In this way the number of sources needed is still $O(\log(\frac{\log n}{\log k})) + O(1)$, but the error is $k^{-\Omega(\log(\frac{\log n}{\log k})+1)}$. As an example, when k is at most $2^{\log^\alpha n}$ for some constant $0 < \alpha < 1$, the error is $k^{-\Omega(\log\log n)}$.

Now we show that we can actually give an extractor for a constant number of independent (n, k) sources, plus two independent (n, k)-block sources, each with $O(\log(\frac{\log n}{\log k})) + O(1)$ blocks. The extractor is very similar to the extractor for independent sources.

Theorem 5.9. *There exists an absolute constant $c > 0$ and a polynomial time computable function $\mathsf{BExt} : \{0,1\}^{cn} \times \{0,1\}^{tn} \times \{0,1\}^{tn} \to \{0,1\}^m$ such that for any $n, k \in \mathbb{N}$ with $k > \log^{10} n$ and $t = O(\log(\frac{\log n}{\log k})) + O(1)$, if $X = (X_1, \cdots, X_c)$ are c independent (n, k) sources and $Y = (Y_1 \circ \cdots \circ Y_t), W = (W_1 \circ \cdots \circ W_t)$ are 2 independent $(k, \cdots, k)$ block sources such that X is independent of (Y, W), then*

$$\mathsf{BExt}(X, Y, W) \approx_\epsilon U_m,$$

where $m = \Omega(k)$ and $\epsilon = 1/\mathrm{poly}(k)$.

Proof sketch. As before, we first use a constant number of independent sources $X = (X_1, \cdots, X_c)$ to obtain a somewhere random source Z with $N = \mathrm{poly}(n)$ rows such that there exists $S \subset [N]$ with $|S| \geq (1-\epsilon)N$ such that any pair of rows in S is ϵ-close to uniform, for some $\epsilon = 1/\mathrm{poly}(n)$. Next, we want to use the lightest bin protocol to reduce the number of rows in the somewhere random source. In the extractor for independent sources, each time we use an independent (n, k) source to reduce the number of rows from N to roughly $N^{3/4}$. Here, however, each time we will use one block from either Y or W. If at one time we use a block from Y, then the next time we will use a block from W. More specifically, first we run the lightest bin protocol on Z, and use the strings in the lightest bin as seeds to apply a strong extractor Ext to Y_1. Thus we obtain a somewhere random source Z_1. Next we run the lightest bin protocol on Z_1, and use the strings in the lightest bin as seeds to apply a strong extractor Ext to W_1. Thus we obtain a somewhere random source Z_2. We then run the lightest bin protocol on Z_2, and use the strings in the lightest bin as seeds to apply a strong extractor Ext to Y_2. Thus we obtain a somewhere random source Z_3. We keep on doing this until the rows in the somewhere random source Z_t reduces to say k^5. Assume Z_t is obtained from Y, finally we can use the extractor BExt from theorem 3.12 to extract from Z_t and another $O(1)$ blocks of W.

For the analysis, notice that by lemma 5.4, when we are computing Z_{i+1} we can fix all previous $Z, Z_1, \cdots, Z_{i-1}$ and with high probability over this fixing, Z_i is a somewhere random source such that there exists a large fraction of rows where any pair of the rows is close to uniform. By induction one can show that after all these fixings, Z_i is a deterministic function of either Y_j or W_j, for some block j. Without loss of generality assume that Z_i is a deterministic function of Y_j. Thus Z_i is independent of W_j. The property of a block source guarantees that after the fixings, W_j is still a k-source. Thus we can use Z_i and W_j to compute Z_{i+1}. Note that if we now further fix Z_i, then indeed Z_{i+1} is a deterministic function of W_j. Moreover, Y_{j+1} is still a k-source. Finally, we can use the extractor BExt from theorem 3.12 to obtain the final output. ■

6 Applications in Network Extractor Protocol

We now apply our independent source extractors to network extractor protocols. The following theorem is proved in [KLRZ08].

Theorem 6.1. *For every $n, k, p, t \in \mathbb{N}$ assume that there is an explicit C-source extractor for (n, k) sources with output length $\Omega(k)$ and error ϵ, then there is an explicit r-round $(t, p - 1.1(r+1)t, \epsilon + 2^{-k^{\Omega(1)}})$ network extractor protocol for (n, k) sources, where $r = \lceil \frac{\log \mathsf{C}}{\log \log k} \rceil + 1$.*

Remark 6.2. The constant 1.1 can be replaced by $1 + \alpha$ for any constant $\alpha > 0$.

Plugging our extractor for independent sources which takes $O(\log(\frac{\log n}{\log k})) + O(1)$ sources with error $1/\mathrm{poly}(k)$, we obtain the following theorem.

Theorem 6.3. *There exists a constant $c > 1$ such that for every $n, k, p, t \in \mathbb{N}$ with $k > \log^c n$, there is an explicit 2-round $(t, p - 3.1t, 1/\mathrm{poly}(k))$ network extractor protocol for (n, k) sources.*

Remark 6.4. The constant 3.1 can be replaced by $3 + \alpha$ for any constant $\alpha > 0$.

In the definition of a network extractor, let $\mathcal{G} = \{i_1, \ldots, i_g\}$ denote the set of players with private, random outputs: $|(B, Z_i) - (B, U_m)| < \epsilon$. Because each Z_i depends only on X_i and B, the above condition implies that

$$|(B, (X_i)_{i \notin \mathcal{G}}, (Z_i)_{i \in \mathcal{G}}) - (B, (X_i)_{i \notin \mathcal{G}}, U_{gm})| < g\epsilon.$$

In other words, after running the network extractor protocol, the joint distribution of the outputs of all the players in $\mathcal{G}$ is close to being independent and uniform, even after seeing all communication and all the sources of the rest of the players. Since $g < p$ and our independent source extractor can be made to have error $1/k^C$ for any constant $C > 1$, as long as p/k^C is small enough, we can run any existing distributed computing protocols using the output of our network extractor protocol. For example, we can obtain the following theorems.

Theorem 6.5 (Synchronous Byzantine Agreement)**.** *There exists a constant $c_1 > 1$ such that for any constants $\alpha > 0$ and $c_2 > 1$ the following holds. Assume p players each has access to an independent (n, k)-source with $k > \log^{c_1} n$ and $k > p^{1/c_2}$, then there exist explicit (in n) synchronous $O(\log p)$ expected round protocols for Byzantine Agreement in the full information model that tolerates $(1/5 - \alpha)p$ faulty players.*

Theorem 6.6 (Leader Election)**.** *There exists a constant $c_1 > 1$ such that for any constants $\alpha > 0$ and $c_2 > 1$ the following holds. Assume p players each has access to an independent (n, k)-source with $k > \log^{c_1} n$ and $k > p^{1/c_2}$, then there exist explicit (in n) synchronous $\log^* p + O(1)$ round protocols for leader election that tolerates $(1/4 - \alpha)p$ faulty players.*

7 Conclusions and Open Problems

In this paper we give new explicit extractors for independent weak random sources that improve previous best results exponentially. We then apply our extractor to network extractor protocols and obtain distributed computing protocols that can tolerate a nearly optimal fraction of faulty players even for weak sources with entropy as small as $\mathrm{polylog}(n)$. This dramatically improves previous results.

Several natural interesting open problems remain. The first is to reduce the error of our extractor. Currently we only achieve error $1/\mathrm{poly}(k)$ (or slightly better). It would be nice to improve the error to $2^{-k^{\Omega(1)}}$, as in [BRSW06]. Second and more importantly, our techniques seem promising for further improvement. For example, instead of just using pair-wise independence in the SR-source, we can try to use r-wise independence for larger r. This may reduce the number of rows in the SR-source faster, and thus resulting in extractors that need fewer sources. However, if r gets larger then correspondingly we need the error ϵ to be smaller, which may need more independent sources to achieve. Thus, there is some trade-off and it would be nice to see what is the limit of our techniques. Finally, it is an open problem to see if our techniques can be applied to constructing extractors or dispersers for other classes of sources.

References

[BIW04] Boaz Barak, R. Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393, 2004.

[BKS+05] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.

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

[BRSW06] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2 source dispersers for $n^{o(1)}$ entropy and Ramsey graphs beating the Frankl-Wilson construction. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

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

[CRS12] Gil Cohen, Ran Raz, and Gil Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity*, 2012.

[DLWZ11] Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy amplification and non-malleable extractors via character sums. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011.

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

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 601–610, 2009.

[Fei99] Uriel Feige. Noncryptographic selection protocols. In IEEE, editor, *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 142–152. IEEE Computer Society Press, 1999.

[GSV05] Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In *DISC 2005*, 2005.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM*, 56(4), 2009.

[KLR09] Yael Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

[Li11] Xin Li. Improved constructions of three source extractors. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011.

[Li12a] Xin Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, 2012.

[Li12b] Xin Li. Non-malleable condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. Technical report, Arxiv, 2012. arXiv:1211.0651.

[Li12c] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the 53nd Annual IEEE Symposium on Foundations of Computer Science*, 2012.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *CRYPTO '97*, 1997.

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

[Rao06] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[Raz05] Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RZ01] Alexander Russell and David Zuckerman. Perfect information leader election in $\log^* n + O(1)$ rounds. *Journal of Computer and System Sciences*, 63(4):612–626, 2001.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de