The Dissertation Committee for Jesse John Kamp
certifies that this is the approved version of the following dissertation:

Deterministic Extractors

Committee:

David Zuckerman, Supervisor

Anna Gál

Adam Klivans

Vijaya Ramachandran

Salil Vadhan

Deterministic Extractors

by

Jesse John Kamp, S.B

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2007

To Noelle

Acknowledgments

Firstly, I would like to thank my advisor, David Zuckerman. I appreciate his always being able to suggest interesting research problems, while at the same time always being open to listen to and encourage whatever ideas excite me at the time. As well as being a great advisor, he has been a great collaborator on nearly all of the work in this thesis. I also appreciate him taking me with him for his stay at Harvard University.

I wish to thank Anup Rao and Salil Vadhan, who were collaborators on some of the results in this thesis. I really appreciated being able to work with them. I also wish to thank the members of my committee: Anna Gál, Adam Klivans, Vijaya Ramachandran, and Salil Vadhan.

There were many others who contributed to my journey through graduate school. Among these I would first like to thank Anindya Patthak, Vladimir Trifonov, Ned Dimitrov, and all of my other fellow students. Secondly, I would like to thank all of my other friends who supported me. I would also like to thank Vitaly Shmatikov, David Wagner, Hoeteck Wee, and Luca Trevisan for illuminating conversations, as well as anyone else who I've forgotten about.

I would like to thank my parents, without whom I wouldn't exist, for raising me well and encouraging my learning from a young age. I would also like to thank

God, without whom none of this would have been possible.

Finally, I would like to thank Noelle for her love and support, even despite the sometimes difficult circumstances of being apart for so long. I look forward to soon spending the rest of my life with her.

JESSE JOHN KAMP

The University of Texas at Austin

May 2007

Deterministic Extractors

Publication No. _____

Jesse John Kamp, Ph.D.
The University of Texas at Austin, 2007

Supervisor: David Zuckerman

We study constructions of deterministic extractors for various specialized classes of sources. Deterministic extractors for a class of sources are functions such that for any random source in the class, the output of the extractor is close to uniform. Thus, we can transform weak randomness into true randomness that can be used in applications. For example, the true randomness we extract can be used in cryptographic protocols, such as generating cryptographic keys. Other applications include distributed computing and randomized algorithms. We have examined some of the most general and interesting classes of sources for which we can construct such extractors. For each class, our goal is to construct extractors with exponentially small error that extract as much of the min-entropy in the source as possible and that work

even when the min-entropy in the source is small.

In particular, we construct extractors for sources that only have access to a small amount of space. This construction is based on a construction for independent sources, which are sources consisting of a number of independent smaller sources. We also give results for oblivious bit-fixing and symbol-fixing sources. In such sources some of the bits (or symbols) are fixed and then the rest are chosen uniformly at random. These results also have an application in the area of exposure-resilient cryptography, giving adaptive All-Or-Nothing-Transforms. We also have results on non-oblivious bit-fixing sources and affine sources.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

True randomness is needed for many applications. For example, randomness is needed in cryptographic protocols, distributed computing, and randomized algorithms. Yet most physical sources of randomness are not truly random, and some are quite weak in that they can have substantial biases and correlations. Weak random sources can also arise in cryptography when an adversary learns some partial information about a random string. A natural approach to dealing with weak random sources is to apply a *randomness extractor* — a function that transforms a weak random source into an almost-perfect random source.

Perhaps the first and most well known example of a randomness extractor was due to von Neumann [vN51]. His extractor extracts randomness from a sequence of tosses of a coin with unknown bias. It works by pairing the bits given by the coin and outputting 1 if they are 01 and 0 if they are 10. Otherwise, on 00 and 11 it doesn't output anything. In this way it is able to extract completely unbiased bits. This extractor has also seen practical application, as Intel's random number generator [JK99] uses it as one of its components.

With von Neumann's early work as inspiration, there was a significant body

of work in the 80's focused on this problem of randomness extraction, with researchers considering richer and richer models of weak sources, e.g. [Blu86, SV86, CG88, Vaz87, CFG+85, BBR88, BOL90, LLS89]. However, attempts to handle sources that do not have a significant amount of independence ran into strong impossibility results showing that it is impossible to devise a single function that extracts even one bit of randomness from sufficiently general classes of sources [SV86].

These impossibility results led researchers to focus on the weaker task of simulating probabilistic algorithms with weak random sources [VV85, CG88, Vaz86, CW89, Zuc96]. This line of work culminated in the introduction, by Nisan and Zuckerman [NZ96], of the notion of a *seeded* randomness extractor, which uses a small number of additional *truly random* bits, known as the *seed*, as a catalyst for the randomness extraction. Using a random seed seems at first to be cheating, since if we had access to true randomness we wouldn't need an extractor in the first place. However, for the particular application of simulating probabilistic algorithms we can enumerate over all possible seeds, then combine the results (by taking the majority, for example) to get the answer. If the seed length is logarithmic (which many constructions give), then the algorithm is still polynomial time. Seeded randomness extractors have turned out to have a wide variety of other applications and were found to be closely related to many other important pseudorandom objects. Thus, they were the main focus of attention in the area of randomness extraction in the 90's, with a variety of very efficient constructions. (See [NTS99, Sha02] for surveys.)

However, in some applications where we need randomness, such as in many cryptographic protocols, we can't enumerate over all of the seeds of a seeded extractor and so we really do need a way to directly generate randomness from weak random sources. For example, in key generation we need to randomly generate a

single cryptographic key, and so being able to enumerate over all possible keys generated by the seeded extractor doesn't help. Since we can't have "seedless", a.k.a. *deterministic* extractors for general weak sources, the question then becomes what types of more restricted classes of sources we can extract from deterministically. After being mostly abandoned in favor of seeded extractors, interest in this question was renewed in recent years by the works of Trevisan and Vadhan [TV00] and Dodis [Dod00a]. In this thesis, we continue this investigation into constructing deterministic extractors for various classes of sources. We also examine applications of deterministic extractors. For example, seedless extractors for specific classes of sources were found to be useful in mitigating partial key exposure in cryptography [CDH+00, Dod00b].

Before proceeding, we recall a few standard definitions: the *min-entropy* k of a source X is defined as $H_\infty(X) = \min_s(\log(1/\Pr[X = s]))$. (Here and throughout, all logarithms are base 2 unless otherwise specified.) We use min-entropy rather than the usual Shannon entropy because min-entropy better represents the worst case entropy of a single sample from the source, which is what we need for extraction. The *min-entropy rate* δ for a source on $\{0,1\}^n$ is defined as $\delta = H_\infty(X)/n$, so it is the fraction of the total possible min-entropy that is present in the source. The *variation distance* between random variables X_1 and X_2 on Ω is defined as

$$|X_1 - X_2| = \max_{S \subseteq \Omega} |\Pr[X_1 \in S] - \Pr[X_2 \in S]| = \frac{1}{2} \sum_{s \in \Omega} |\Pr[X_1 = s] - \Pr[X_2 = s]|.$$

The following definition, taken from [Dod00a] and generalizing definitions from [TV00], formalizes our notion of a deterministic extractor.

Definition 1.0.1. A function $f : \{0,1\}^n \to \{0,1\}^m$ is an ε-extractor for a set $\mathcal{X}$ of random sources if, for every $X \in \mathcal{X}$, $f(X)$ is within variation distance ε of the

uniform distribution.

1.1 Nonconstructive Results

Before looking for any explicit constructions, a natural question to ask is whether deterministic extractors for a given class of sources $\mathcal{X}$ even exist. In particular, we consider the general case where all we know is that each source in the class $\mathcal{X}$ has min-entropy at least k. Since for all of the classes of sources we consider we have a lower bound on the min-entropy, this general case encompasses all of these classes. This will allow us to compare our explicit constructions with our nonconstructive bounds to see how much better we can hope to make our explicit constructions. To show that an extractor for a class $\mathcal{X}$ exists, we show that any randomly chosen function is an extractor for $\mathcal{X}$. Using the probabilistic method, it's not hard to show that even if the number of sources $|\mathcal{X}|$ is exponentially large a random function is still an extractor with high probability. In fact, we can have $|\mathcal{X}| = 2^{2^{\alpha k}}$ for any $\alpha < 1$. (Note that the set of all random sources with min-entropy k has size about 2^{n2^k}, which is much larger.) The output length of this function can even be almost as large as k.

One of the main problems with this approach is that a random function is not efficiently computable and takes exponentially many random bits to select. However, Dodis [Dod00a], with an argument based on [TV00], showed that similar results can be achieved by selecting a random t-wise independent function (for a high enough t). This construction is efficient and only requires a few random bits. Note that it's not necessarily cheating to use randomness to select the function, since we only require a one time use of randomness in the beginning, and not each and every time we use the function as in the case of seeded extractors. However, the param-

eters for the results proved in [Dod00a] are not quite as good as in the results we prove.

While these nonconstructive results are nice, they still require a one time investment of randomness and the function we obtain is only an extractor with high probability. What these results do show is that as long as the number of sources we wish to extract from is not too large, constructing deterministic extractors is potentially feasible. This motivates our interest in looking for natural classes of sources for which we might be able to construct explicit deterministic extractors. We wish to find the most natural, useful, and general sources for which deterministic extraction is possible. To do this, we examine many particular classes of sources.

1.2 Overview

In the rest of the introduction we outline in more detail the classes of sources and applications we examine in this thesis. For each class, we outline what was known previously and the new results we obtain in this thesis. The first classes we consider are bit-fixing and symbol-fixing sources, which consist of a string of bits or symbols, some of which are fixed and the rest of which are chosen uniformly at random. The results for these classes are based on joint work with David Zuckerman [KZ06]. We next look at sources consisting of a number of independent smaller sources. The most general class we consider are sources generated using a small amount of space. The results for these two classes are based on joint work with Anup Rao, Salil Vadhan, and David Zuckerman [KRVZ06]. Finally, we look at sources uniformly distributed over affine subspaces of $\{0,1\}^n$.

For all of the classes of sources, we will look at tradeoffs between the various parameters in the extractor. We will usually think of the length of the source as be-

ing fixed, and attempt to optimize the error, the output length, and the min-entropy of the sources for which our extractors work. Since when extracting randomness we want the output to be very close to uniform, it is most useful to have error exponentially small in the output length. Since it's impossible to have a higher output length than the min-entropy, our primary goal is to construct extractors with exponentially small error which have output length as close as possible to the min-entropy of the source. The primary issue that comes up is that frequently our constructions only work when the min-entropy is at least a certain threshold. There is often a trade-off among various constructions between the min-entropy threshold and the output length. Some constructions work for smaller min-entropies but extract fewer bits. Ideally we want extractors which extract a near optimal number of bits which work even for min-entropies that are logarithmic in the length of the source.

1.3 Oblivious Bit-Fixing and Symbol-Fixing Sources

The first sources we consider are bit-fixing sources, in which some subset of the bits are fixed and the rest are chosen at random. We start with these sources because they are the simplest class of sources that we study, yet they also have interesting applications. There are two classes of bit-fixing sources, depending on whether the fixed bits are chosen before or after the random bits are determined, known respectively as oblivious and non-oblivious bit-fixing sources. We first look at the oblivious case.

1.3.1 Previous Work

In oblivious bit-fixing sources, the fixed bits are chosen first and then the rest are chosen randomly [CFG+85, CW89]. Such sources arise in the context of producing random bits from random coins in a distributed environment, where some of the processors produce faulty coins (not dependent on the good coins). Clearly we can extract one uniform bit by taking the parity of the input bits. However, to extract even two perfectly uniform bits Chor et al. [CFG+85] showed that at least $1/3$ of the bits need to be random. Friedman generalized this result to obtain bounds on the number of random bits needed for longer outputs [Fri92]. However, since we allow some error in our extractor, these bounds don't apply. Despite this, the best previous constructions for extracting many bits still required that at least half of the bits be random [KJS01, BS00].

1.3.2 Our Results

In this thesis, we improve upon the previous results by showing how to extract $\Omega(k^2/n)$ bits from any n bit long oblivious bit-fixing source with min-entropy $k > \sqrt{n}$ and $\Omega(\log k)$ bits for any k. In both cases we have exponentially small error.

We also introduce a variant on oblivious bit-fixing sources known as oblivious symbol-fixing sources. In d-ary oblivious symbol-fixing sources, instead of bits we have a string of symbols from a d-symbol alphabet. Some of the symbols are uniformly random while the rest are fixed. This model is somewhat more restricted than the bit-fixing model. For example, for $d = 2$, this model is the same as the oblivious bit-fixing model, and for $d = 4$, it corresponds to oblivious bit-fixing sources where the fixed and random bits have to come in pairs. However, it is still an extremely natural and interesting model.

For symbol-fixing sources with $d > 2$, we get much better results than for oblivious bit-fixing sources. We construct an extractor that extracts a constant fraction of the randomness for sources with any number of random symbols, with the constant depending on d. In particular, as d grows large it extracts almost all of the randomness.

Subsequent to our work, Gabizon, Raz, and Shaltiel [GRS04] have improved upon our constructions of extractors for oblivious bit-fixing sources. They give two main extractor constructions. The first construction is able to extract almost all of the random bits from oblivious bit-fixing sources that have min-entropy $k > \sqrt{n}$, with exponentially small error. The second construction is able to extract $\Omega(k)$ bits as long as $k > \log^c n$ for some constant c, with much higher error.

1.3.3 Exposure-Resilient Cryptography

Our extractors for oblivious bit-fixing sources also have applications in the area of exposure-resilient cryptography. In traditional cryptography, secret keys are required to remain secret. Most cryptographic schemes have no security guarantees even when an adversary learns only a small part of the secret key. Is it possible to achieve security even when the adversary learns most of the secret key? The class of mappings known as All-Or-Nothing Transforms (AONT) address this issue. AONT's were originally introduced by Rivest [Riv97] to address security concerns arising in the context of block ciphers. An AONT is an efficient randomized mapping that is easy to invert given the entire output, but where an adversary would gain "no information" about the input even if it could see almost the entire output of the AONT. Various important applications of the AONT have been discovered, such as the previously mentioned application of protecting against almost complete

exposure of secret keys [CDH+00], and increasing the efficiency of block ciphers [MPR, JSY99, Bla96].

Boyko used the Random-Oracle model to give the first formalizations and constructions of the AONT [Boy99]. Canetti et al. gave the first constructions in the standard computational model [CDH+00]. For their construction, they introduced a new, related primitive known as an Exposure-Resilient Function (ERF). An ERF is an efficiently computable deterministic function where the output looks random even if the adversary obtains almost all of the bits of a randomly chosen input. They then reduced the task of constructing an AONT to constructing an equivalent ERF. This work was extended by Dodis et al. [DSS01] to the adaptive setting, where the adversary can decide which bits to look at based on the bits he has already seen. This setting is applicable to the problem of partial key exposure, where it is likely that the adversary would be adaptive.

An important idea used in both [CDH+00] and [DSS01] is that it is possible to construct ERF's in the computational setting by first constructing ERF's in the statistical setting and then applying a pseudorandom generator to the output. This allows us to get longer output lengths, which is useful for applications. Because of this observation, we can restrict our attention to constructing ERF's in the statistical setting, where the output must be statistically close to the uniform distribution. However, though Dodis et al. [DSS01] give a probabilistic construction of adaptive statistical ERF's, the problem of giving an explicit construction was left open (see also [Dod00a]).

We address this problem by giving an explicit construction of efficient adaptive ERF's in the statistical setting, which in turn gives an explicit construction of adaptive AONT's. Our construction actually gives a stronger function, known as an almost-perfect resilient function (APRF), introduced by Kurosawa et al. [KJS01].

An APRF is like an ERF, except it works for even the case where the adversary can fix some bits of the input instead of merely looking at them. The connection between APRF's and exposure resilient cryptography was shown in [DSS01], where it was proved that APRF's are also adaptive ERF's. In fact, it is easy to see that APRF's are essentially the same as deterministic extractors for oblivious bit-fixing sources. So by constructing extractors for oblivious bit-fixing sources, we will also get APRF's and thus adaptive statistical ERF's and AONT's.

1.4 Non-Oblivious Bit-Fixing Sources

1.4.1 Previous Work

Non-oblivious bit-fixing sources [BOL90, KKL88], unlike the oblivious case, can have the fixed bits depend on the random bits chosen. The problem of extracting from such sources was originally studied in the context of collective coin-flipping [BOL90]. Collective coin-flipping is the problem of producing a random bit from random coins in a distributed environment, where some of the bits may be adversarially chosen. Here, we consider the case where each player produces a single bit, and the adversarial bits can depend upon the bits produced by the other players. This scenario is exactly the same as extraction of a single bit from non-oblivious bit-fixing sources. For the case of extracting a single bit, nearly optimal lower [KKL88] and upper [AL93] bounds are known, though the upper bound is not completely constructive. Because these sources are much more general than in the oblivious case, we need almost all of the bits in the source to be random in order to extract.

1.4.2 Our Results

Previously, little attention has been given to generalizing the single bit results to the case of multiple output bits. We give bounds for this case. We show how to construct an ε-extractor for the set of length n non-oblivious bit-fixing sources with $n-\ell$ random bits which extracts $(\varepsilon/\ell)^{1/\log_3 2}n$ bits. In the other direction, we show that at most $O(n\varepsilon/\ell)$ bits can be extracted from such non-oblivious bit-fixing sources.

1.5 Total-Entropy Independent Sources

1.5.1 Previous Work

Much work has been done on extracting from a small number of *independent sources*. In these sources, we know nothing about the structure of the individual sources. The only guarantee is that each source has a certain amount of min-entropy. This model was essentially introduced by Chor and Goldreich [CG88]. The idea of generating randomness using independent sources had previously been introduced using a somewhat different model by Santha and Vazirani [SV86]. The initial results of Chor and Goldreich extracted from two independent sources with min-entropy rate greater than one half. Recently, extractors have been obtained for multiple independent sources with any constant and even subconstant, polynomially small, min-entropy rate, but each of these require at least 3 independent sources [BIW04, BKS+05, Raz05, Rao06]. The best extractor for 2 independent sources requires that the sources have min-entropy at least some constant slightly less than $1/2$ [Bou05]. This model is appealing because the individual sources can have arbitrary correlations and biases, and it seems plausible that we can ensure

independence between a few such sources.

We note that the Chor and Goldreich model for extracting from two sources is actually stronger than the model stated above. Instead of requiring each source to have a certain min-entropy, they only require a bound on the total min-entropy over both sources. This requirement is intuitively appealing since it is easier to guarantee the total min-entropy than the min-entropy for each source, so this model is much more general than the standard independent source model. The big question is whether the results for more than two sources can be similarly generalized to a model where only the total min-entropy is known. This is the case we address.

1.5.2 Our Results

To generalize the independent source model, we introduce the model of *total-entropy independent sources*. These are sources consisting of r independent sources over $\{0,1\}^\ell$. The case of independent sources where you only know the total min-entropy then corresponds to r being small while ℓ is large. So in this case we're viewing each symbol as a long source. For example, when $r = 2$ we get the sources studied by Chor and Goldreich [CG88].

This model also encompasses the case where r is large, but ℓ is small. This case can be thought of as a generalization of symbol-fixing sources to the case where instead of each symbol being totally uniformly random or fixed, each symbol is allowed to be chosen from any probability distribution. For example, when $\ell = 1$, we get a sequence of independent bits, each of which has a (possibly) different and unknown bias. In this way, we see that our model not only generalizes the bit-fixing model, but also generalizes von Neumann's biased coin model [vN51]. Even for this relatively simple and natural case, nothing had been known prior to our work.

All of our results are obtained by generalizing previous techniques for extracting from independent and symbol-fixing sources. All of these results have various tradeoffs between the number of symbols r, the length of each source ℓ, and the total min-entropy k. Generally speaking, which construction is better depends on whether the total-entropy independent source you wish to extract from looks more like the independent source model or the symbol-fixing source model.

For total-entropy independent sources that are more like symbol-fixing sources, we generalize our techniques for extracting from oblivious bit-fixing sources together with the techniques of Gabizon, Raz, and Shaltiel [GRS04] to get an extractor which extracts $(1-o(1))k$ bits with exponentially small error as long as $\ell < \frac{1}{2}\log r$ and the min-entropy rate is at least $\tilde{O}(1/\sqrt{r\ell})$. We can also generalize our extractors for oblivious bit-fixing sources with small min-entropy, together with techniques from [GRS04], to get an extractor that extracts $(1-o(1))k$ bits from any total-entropy independent source as long as $k \geq (2^{\ell}\log r)^{C}$ and $\ell \leq b\log k$ for some constants C and b. This extractor, however, has only polynomially small error.

If we think of the total-entropy independent sources as being more like independent sources, we can get other results which generalize independent source extractors. The first result we get is that we can easily generalize the exponential sum based independent source extractor of Bourgain et al. [BGK06, Bou05] to the total-entropy independent source case. Combining this extractor with the techniques from [GRS04], we get an extractor that extracts any constant fraction of the min-entropy, say $.99k$, from any constant min-entropy rate total-entropy independent source with exponentially small error. Note that in this case, unlike the previous case, we have no restriction on the source length ℓ and can have arbitrarily large sources.

In a more involved construction, we construct an extractor that uses some

of the ideas from the work of Rao [Rao06] and recent constructions of randomness efficient condensers [BKS+05, Raz05], again combined with [GRS04]. It extracts $(1-o(1))k$ bits with exponentially small error from total-entropy independent sources with min-entropy rates at least $(r\ell)^{-\eta}$ for some fixed constant η. This improves on the previous construction by allowing us to extract even when the min-entropy rate is sub-constant. This construction requires the ability to find large primes efficiently, which can be done assuming Cramer's conjecture on the density of primes [Cra37].

1.6 Small-Space Sources

1.6.1 Previous Work

Perhaps the most natural small class of sources to consider are those computable with a small amount of computational resources. Such sources were first studied by Trevisan and Vadhan [TV00]. This seems to be a plausible model for physical random sources and generalizes a number of the previously studied models. They focused on the case that the source is sampled by either a small circuit or an algorithm with a limited running time. Their main result is a construction of polynomial-time extractors for such sources based on some strong but plausible complexity assumptions. It would be nice to have unconditional constructions (as well as ones that are more efficient and have better error). However, they showed that complexity assumptions are needed for the original model of sources generated by a time-bounded algorithms. Thus, they suggested, as a research direction, that we might be able to construct unconditional extractors for sources generated by *space-bounded* algorithms. This is one of one of the primary models we study

in this thesis.

The first example of an explicit construction was due to Blum [Blu86], who showed how to extract from sources generated by a finite Markov chain. His results generalized the earlier results of von Neumann [vN51] for extracting from an independent coin with unknown bias. However, the finite Markov chain model is very restricted; it has a constant-size description and does not capture most sources computable with small non-uniform space.

The exact model for small-space sources we consider essentially generalizes the finite Markov chain model of Blum [Blu86] to time dependent Markov chains. Doing so yields a much richer class of sources. This model is similar to the one previously considered by Koenig and Maurer [KM04, KM05].

Koenig and Maurer [KM04, KM05] gave the first explicit constructions of extractors for space-bounded sources. Their extractors require the min-entropy rate to be least $1/2$. We do not know of any other constructions for space-bounded sources prior to our work, even space 0 sources, which are simply sources of independent bits each of which has a different, unknown, bias.

1.6.2 Our Results

Before describing our results, we first need to give a more specific description of the model we use for space s sources. Our model of a space s source is basically a source generated by a width 2^s branching program. The exact model we consider is that at each step the process generating the source is in one of 2^s states. This can be modelled by a layered graph with each layer corresponding to a single time-step and consisting of vertices corresponding to each of the states. From each node v in layer i, the edges leaving v (going to layer $i+1$) are assigned a probability distribution as

well as an output bit for each edge. Unlike in Blum's model [Blu86], the transitions can be different at each time-step. See Figure 1.6.2 for an example of a space $s = 2$ source.

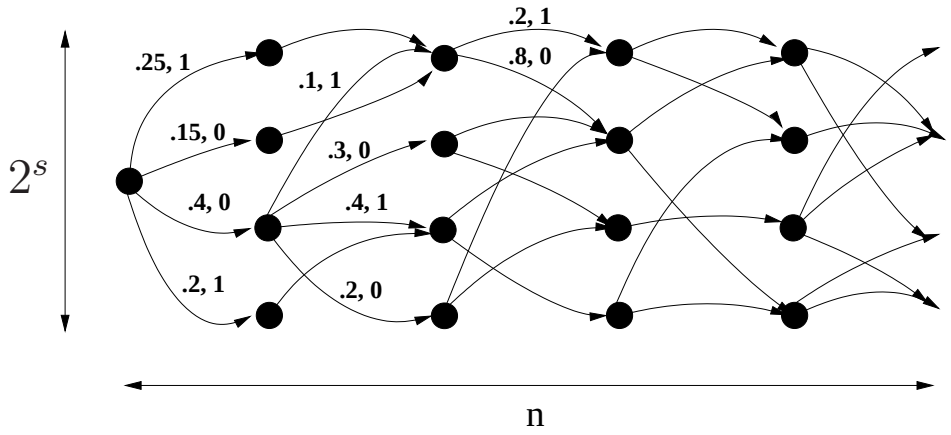

Figure 1.1: Part of a space $s = 2$ source

Small-space sources are very general in that most other classes of sources that have been considered previously can be computed with a small amount of space. This includes von Neumann's model of a coin with unknown bias [vN51], Blum's finite Markov chain model [Blu86], symbol-fixing sources, and independent sources. Strong results in this last model will not follow directly from strong results in the small-space model, but our results do generalize, for example, the results of [BIW04]. In fact, the only model for which deterministic extractors have been given that appears unrelated to our model is that of affine sources, which we discuss later.

In this thesis, we give a number of constructions of extractors for small-space sources. All of these extractors are obtained by reducing small-space sources to total-entropy independent sources and applying the extractors from the previous section.

We obtain the best results by using our extractor that uses some of the ideas of the work of Rao [Rao06], together with recent constructions of randomness efficient condensers [BKS+05, Raz05]. This extractor extracts $(\delta - \beta)n$ bits with exponentially small error from sources with min-entropy $k = \delta n$. It works whenever $\beta > n^{-\eta}$ for some fixed constant η and $s = O(\beta^3 n)$. In particular, for constant min-entropy rate δ sources, it extracts almost all of the entropy even when the space is a constant fraction of n. However, this result requires the assumption that it is possible to find large primes efficiently.

We obtain unconditional constructions using our other total-entropy independent source extractors. Using our extractor based on the exponential sum estimates of [BGK06], it is possible to extract almost all of the entropy with exponentially small error from constant min-entropy rate sources even when the space is a small constant fraction of n. In particular, we note that this result basically matches our conditional result for constant min-entropy rate sources. Using our extractor that's based on our random walk techniques for oblivious symbol-fixing sources, it is possible to extract almost all of the entropy for min-entropy rate as low as $\delta = \Omega(1/\log n)$. However, this extractor only works for space $O(\delta \log n)$.

Note that even our best construction requires min-entropy rate at least $n^{-\eta}$ for some small constant η. For space 1 small-space sources, we're able to overcome this limitation for a slightly more restricted model where we restrict the output value at each step to be the same as the next state. In this model, we construct an extractor that works as long as $k \gg n^{4/5}$, and that outputs $\tilde{\Omega}(k^2/n)$ bits.

1.7 Affine Sources

Another generalization of oblivious bit-fixing sources is that of affine sources. These are sources of n bits which are uniformly distributed over an affine subspace of dimension k. Such subspaces are also a natural subclass of all subsets, from which we know extraction is impossible. In this thesis, we construct $2^{n/2-k+m/2}$-extractors for affine sources with output length m. These constructions work well for $k > n/2$, but don't work for smaller k. Prior to our results, it was only known how to extract a single bit from affine sources with $k > n/2$. Recently, Bourgain [Bou07] has improved upon our results by giving a construction of extractors which work for k any constant fraction of n.

Chapter 2

Mathematical Preliminaries

For ease of notation, we sometimes assign non-integer values to integer variables when we mean to round off the values. It is easy to observe that any errors introduced in this manner do not affect our results. We frequently write our definitions in terms of a single function f, though we really mean f to represent a family of functions over all input lengths, so asymptotic notions make sense. Given a string $x \in (\{0,1\}^{\ell})^r$ and a set $S \subseteq [r]$ we use x_S to denote the string obtained by restricting x to the indices in S. We use $\circ$ to denote concatenation.

2.1 Probability Definitions

We need some standard definitions for probability distributions. First, we express our probability distributions as probability vectors $p = (p_1, ..., p_n)$ with $\sum_i p_i = 1$. Unless otherwise stated, π represents the uniform probability vector (of the appropriate length). The *variation (statistical) distance* $|p-q|$ between two distributions with probability vectors p and q is half the ℓ_1 distance, so $|p-q| = \frac{1}{2}\sum_i |p_i - q_i|$. Also, we use $\|.\|$ to represent the standard ℓ_2 norm for vectors. It is well known that

$|p-q| \leq \frac{1}{2}\sqrt{n}\|p-q\|$.

A *source* is a family of probability distributions (a probability ensemble). For ease of notation, we usually refer to a source as a single probability distribution.

2.2 Classes Of Sources

We know formally define the classes of sources we study in this thesis, starting with oblivious bit-fixing and symbol-fixing sources, which are the simplest sources we consider.

Definition 2.2.1. [CFG+85] An oblivious bit-fixing source X on $\{0,1\}^n$ with k random bits is a source in which all but k of the bits are fixed and the rest are then chosen uniformly at random.

Definition 2.2.2. An oblivious symbol-fixing source X on $[d]^n$ with k random symbols is a source with n independent symbols each taken from $[d]$, of which all but k are fixed and the rest are then chosen uniformly at random.

Note that for $d = 2^t$, symbol-fixing sources can be viewed as a special case of bit-fixing sources where the bits are divided up into blocks of size t and each block is either fixed or random.

Non-oblivious bit-fixing sources are more difficult to handle, since the fixed bits can depend arbitrarily on the random bits.

Definition 2.2.3. [BOL90] An non-oblivious bit-fixing source X on $\{0,1\}^n$ with k random bits is a source in which k of the bits are chosen uniformly at random and then the remaining $n-k$ bits are chosen, possibly depending on the random bits.

We will also need a slightly weaker notion of symbol-fixing sources when converting bit-fixing sources to symbol-fixing sources.

Definition 2.2.4. An (k, ε)-approximate oblivious symbol-fixing source X on $[d]^n$ is a source with n symbols independently chosen from $[d]$, of which k have distributions within an ℓ_2 distance of ε of uniform.

The next class of sources we study are small-space sources, which we model using branching programs.

Definition 2.2.5. A *space s source* X on $\{0,1\}^n$ is a source generated by a width 2^s branching program. That is, the branching program is viewed as a layered graph with $n+1$ layers with a single start vertex in the first layer and 2^s vertices in each subsequent layer. Each edge is labeled with a probability and a bit value. From a single vertex there can be multiple edges corresponding to the same output bit. The source is generated by taking a random walk starting from the start vertex and outputting the bit values on every edge.

This definition is very similar to the general Markov sources studied by Koenig and Maurer [KM04, KM05]. This is not quite the most general model of such sources imaginable, because we could consider sources that output a variable number of bits depending on which edge is chosen at each step, including possibly not outputting any bits. However, this restriction makes sense in light of the fact that we are primarily interested in sources of fixed length. In this case, the sources in the more general model can be transformed into our model by modifying the states appropriately.

Another important class of sources we study are independent sources, and in particular total-entropy independent sources.[1]

[1]Though for ease of presentation we define total-entropy independent sources only over sources with alphabet size 2^ℓ, more generally the sources could be over alphabets of any size d, as with symbol-fixing sources. All of our results naturally generalize to this more general case.

Definition 2.2.6. A source consisting of r smaller sources on $\{0,1\}^\ell$ is an *independent source* if each of the r smaller sources are independent. A set of r independent smaller sources on $\{0,1\}^\ell$ has total-rate δ if the total min-entropy over all of the sources is $\delta r\ell$ and total-entropy k if the total min-entropy is k.

Definition 2.2.7. A source on $\{0,1\}^\ell$ is *flat* if it is uniformly distributed over a non-empty subset of $\{0,1\}^\ell$.

Note that when $\ell = 1$, a flat independent source is the same as an oblivious bit-fixing source.

Definition 2.2.8. Let X be a random variable taking values in $\{0,1\}^{t\times a}$, viewed as $t \times a$ matrices with entries in $\{0,1\}$. We say that X on $(\{0,1\}^a)^t$ is $(t \times a)$ *somewhere-random*[2] (*SR-source* for short) if it is a random variable on t rows of r bits each such that one of the rows of X is uniformly random. Every other row may depend on the random row in arbitrary ways. We will say that a collection $X_1,\ldots,X_m$ of $(t \times a)$ SR-sources is *aligned* if there is some i for which the i'th row of each X_j is uniformly distributed.

We will also need a relaxed notion of the previous definition to where the "random" row is not completely random, but only has some min-entropy.

Definition 2.2.9. We say that a $(t \times a)$ source X on $(\{0,1\}^a)^t$ has *somewhere-min-entropy* k, if X has min-entropy k in one of its t rows. We will say that a collection $X_1,\ldots,X_m$ of $(t \times a)$ somewhere-min-entropy k sources is *aligned* if there is some i for which the i'th row of each X_j has min-entropy k.

[2]This definition is slightly different from the original one used by Ta-Shma [TS96]. The original definition considered the closure under convex combinations of the class defined here (i.e. convex combinations of sources that have one random row). We use this definition because we can do so without loss of generality and it considerably simplifies the presentation.

Finally, the last class of sources we consider is that of affine sources.

Definition 2.2.10. An *dimension k affine source* on $\{0,1\}^n$ is a source of length n uniformly distributed over an affine subspace of dimension k.

Another way of thinking of an affine source is as a source with k uniformly random bits with the rest of the bits affine combinations of these k random bits.

2.3 Graph Definitions

We define some standard notions used when studying random walks on graphs. Transition matrices indicate the probability of following any edge in a random walk. A (general) *transition matrix* P for a graph $G = (V,E)$ with n vertices is an $n \times n$ matrix with entries $p_{ij} \geq 0$ if $(i,j) \in E$ and $p_{ij} = 0$ otherwise, and $\sum_{j=1}^{n} p_{ij} = 1$ for all rows i. The *uniform transition matrix* P of a d-regular graph $G = (V,E)$ has all non-zero entries equal to $1/d$. The way to view these definitions is that the probability of choosing edge (i,j) if we are currently at vertex i corresponds to p_{ij}. The *stationary probability vector* π for a random walk with transition matrix P is the vector such that $\pi P = \pi$, and is well defined for connected graphs. In the cases we consider, π corresponds to the uniform distribution on the vertices.

For each random walk, the input is a string of values, each of which can take on any value in $[d]$, where d is the degree of the graph. A directed edge (u,v) is *labeled* i if (u,v) is the edge taken when the random walk is at u and receives input value i.

One property that we need in our graphs is that the error shouldn't accumulate in any of the vertices. In order for our graphs to have this property, we require that no vertex has two incoming edges with the same label. Such a graph is said to

be *consistently labeled*. All of our results apply only to consistently labeled graphs.

An expander graph is a graph that has low degree, but is well connected, so that random walks on expanders converge quickly to the uniform distribution. For a given matrix P, let $\lambda(P)$ denote the second largest eigenvalue in absolute value. Here we define expanders in terms of $\lambda(P)$.

Definition 2.3.1. A family of expander graphs is an infinite set of regular graphs G with uniform transition matrix P that have $\lambda(P) = 1 - \varepsilon$ for some constant $\varepsilon > 0$.

We will need all of our expander graphs that we use to be efficiently constructible, that is, we should find the neighbors of any vertex in polynomial time in the length of the vertex label. There are various constructions that give infinite families of constant-degree consistently labeled expander graphs that are efficiently computable, see e.g. [GG81], [LPS88], [Lub94], and [RVW02]. Though these constructions don't work for every degree, we can always construct an expander for a given degree by adding an appropriate number of self loops to an existing expander. It is easy to see that doing so maintains the eigenvalue separation. We also should note that there are expander constructions that work for degrees as small as 3.

2.4 Convex Combinations

Definition 2.4.1. Let $\mathcal{P}$ be a property of sources. Let X be some random variable over some universe. We will say that X is a convex combination of sources with property $\mathcal{P}$ if there exists some random variable I over an arbitrary universe such that for all $i \in \mathsf{supp}(I)$, $X|I = i$ has property $\mathcal{P}$.

A key observation that is essential to our results is that random variables that

are convex combinations of sources with certain good properties are good themselves. This is captured in the following easy propositions:

Proposition 2.4.2. *Let X, Y be random variables such that X is a convex combination of sources that are ε-close to Y. Then X is ε-close to Y.*

Proposition 2.4.3. *Let X, I be random variables such that X is a convex combination of random variables $\{X_i\}_{i \in I}$. Let f be some function such that for all $i \in I$, $f(X_i)$ is a convex combination of sources that have some property $\mathcal{P}$. Then $f(X)$ is a convex combination of sources that have property $\mathcal{P}$.*

We'll also need the following simple lemma.

Lemma 2.4.4. *Let X, Y, and V be distributions over Ω such that X is ε-close to uniform and $Y = \gamma \cdot V + (1-\gamma) \cdot X$. Then Y is $(\gamma+\varepsilon)$-close to uniform.*

Note that X and V could also be combinations of distributions, so this lemma also says that if Y is a convex combination of distributions that with high probability are close to uniform, then Y itself is also close to uniform.

Proof. Let U denote the uniform distribution on Ω and $S \subseteq \Omega$. Then

$$\begin{aligned} |\Pr[Y \in S] - \Pr[U \in S]| &= |\gamma \cdot \Pr[V \in S] + (1-\gamma) \cdot \Pr[X \in S] - \Pr[U \in S]| \\ &\leq \gamma |\Pr[V \in S] - \Pr[X \in S]| + |\Pr[X \in S] - \Pr[U \in S]| \\ &\leq \gamma + \varepsilon. \end{aligned}$$

□

2.5 Seeded Extractors

We will also need to define what it means to have a seeded extractor for a given class of sources.

Definition 2.5.1. A polynomial-time computable function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^m$ is a *seeded* ε-*extractor* for a set of random sources $\mathcal{X}$, if for every $X \in \mathcal{X}$, $\mathsf{Ext}(X, U_s)$ is ε-close to uniform. The extractor is called *strong* if for Y chosen according to U_s, $Y \circ \mathsf{Ext}(X,Y)$ is also ε-close to uniform.

Chapter 3

Nonconstructive Results

In this chapter, we describe nonconstructive results for all of the classes of sources that we examine in this dissertation. These results will help us interpret the constructive results found in later chapters by comparing the constructive results to what is possible nonconstructively.

These nonconstructive results make use of the probabilistic method. We show that a randomly chosen function is an extractor for each of these classes of sources with high probability, and is able to extract almost all of the entropy even when the min-entropy is logarithmically small. In particular, this argument shows that a function achieving these parameters exists. To do so we use a standard argument that shows that a randomly chosen function is an extractor for any class of sources that is not too large, as long as the sources in the class are close to having high min-entropy.[1]

Theorem 3.0.2. *Suppose we have a set $\mathcal{X}$ of random sources on $\{0,1\}^n$ and $\varepsilon > 0$*

[1] In fact, if we wish to save randomness in selecting the function, then Dodis [Dod00a] showed that we can get a similar result by using a random d-wise independent function instead of a completely random function. However, the parameters he proved are not quite as good as we prove here.

such that $\forall X \in \mathcal{X}$, *there is a source* X' *with* $|X' - X| \leq \frac{\varepsilon}{2}$ *and* $H_\infty(X') \geq k$. *Then, with probability* $1 - 1/2^{2^m}|\mathcal{X}|$ *a function chosen uniformly at random is an extractor for* $\mathcal{X}$ *as long as* $k \geq \log(2^m + \log|\mathcal{X}|) + 2\log(1/\varepsilon) + O(1)$. *In particular, as long as* $k \geq \log\log|\mathcal{X}| + 2\log(1/\varepsilon) + O(1)$, *a random function extracts* $m = k - 2\log(1/\varepsilon) - O(1)$ *bits.*

We need the following Chernoff bound to prove Theorem 3.0.2.

Lemma 3.0.3. *Let* $Z_1, \ldots, Z_r$ *be independent indicator random variables such that* $\Pr[Z_1 = 1] = p_i$. *Let* $Z = \sum_{i=1}^n a_i Z_i$ *where* $0 \leq a_i \leq 1$ *for all i, and let* $\mu = \mathbb{E}[Z]$. *Then for any* $0 < \varepsilon \leq 1$

$$\Pr[|Z - \mu| \geq \varepsilon\mu] < 2\exp(-\mu\varepsilon^2/3).$$

Proof. (of Theorem 3.0.2) We'll first use Lemma 3.0.3 to show that a random function is a good extractor for a single source, and then apply the union bound.

Let $f : \{0,1\}^n \to \{0,1\}^m$ be chosen uniformly at random from all functions from n bits to m bits. Fix $X \in \mathcal{X}$ and $S \subset \{0,1\}^m$. Let X' be such that $|X' - X| \leq \varepsilon/2$ and $H_\infty(X') \geq k$. Let Z_x be the indicator random variable for whether $f(x) \in S$. Let

$$Z = 2^k \Pr_{x \leftarrow_R X'}[f(x) \in S] = 2^k \sum_{x \in \mathsf{supp}(X')} \Pr[X' = x] Z_x$$

Since the function f is chosen uniformly at random, $E[Z] = 2^k|S|/2^m$. Thus we can apply Lemma 3.0.3 to get

$$\begin{aligned}\Pr_f\left[\left|\Pr_{x \in X'}[f(x) \in S] - \frac{|S|}{2^m}\right| \geq \varepsilon'\frac{|S|}{2^m}\right] &= \Pr_f\left[\left|Z - \frac{2^k|S|}{2^m}\right| \geq \varepsilon'\frac{2^k|S|}{2^m}\right] \\ &\leq 2\exp\left(-\varepsilon'^2\frac{2^k|S|}{3\cdot 2^m}\right)\end{aligned}$$

Making the change of variables $\varepsilon' = \varepsilon 2^m/|S|$, we get that for any fixed set S,

we proved that

$$\Pr_f[|\Pr[f(X') \in S] - \Pr[U_m \in S]| \geq \varepsilon/2] \leq 2\exp\left(-\left(\frac{\varepsilon 2^m}{2|S|}\right)^2 \frac{2^k|S|}{3\cdot 2^m}\right)$$
$$= 2\exp\left(-\frac{\varepsilon^2 2^k 2^m}{12|S|}\right)$$

Recall that $|f(X') - U_m| = \max_S\{|\Pr[f(X') \in S] - |S|/2^m|\}$. By the union bound over all sets $S \subset \{0,1\}^m$ and all $X \in \mathcal{X}$, and since $2^m/|S| \geq 1$,

$$\Pr_f[\max_S\{|f(X') - U_m| \geq \varepsilon/2\}] \leq 2\exp\left(-\varepsilon^2 2^k/12\right) 2^{2^m}|\mathcal{X}|$$

Now whenever f does satisfy $|f(X') - U_m| < \varepsilon/2$, we have that $|f(X) - U_m| < \varepsilon/2 + \varepsilon/2 = \varepsilon$. Setting the above error to $1/2^{2^m}|\mathcal{X}|$ and solving for k, we get that a function chosen uniformly at random is an extractor for $|\mathcal{X}|$ with probability $1 - 1/2^{2^m}|\mathcal{X}|$ as long as $k \geq \log(2^m + \log|\mathcal{X}|) + 2\log(1/\varepsilon) + O(1)$. In particular, as long as $k \geq \log\log|\mathcal{X}| + 2\log(1/\varepsilon) + O(1)$, we can extract $m = k - 2\log(1/\varepsilon) - O(1)$ bits.

□

3.1 Bit-Fixing, Symbol-Fixing, and Affine Sources

The case of oblivious bit-fixing and symbol-fixing sources is straightforward since for any given input length there are only a finite number of sources, so we can directly apply Theorem 3.0.2. We start off with the result for oblivious symbol-fixing sources.

Theorem 3.1.1. *For oblivious symbol-fixing sources with k random symbols, a*

function $f : [d]^n \to \{0,1\}^m$ *chosen uniformly at random is an* ε*-extractor with output length* $m = k \log d - 2\log(1/\varepsilon) - O(1)$ *with probability at least* $1 - 1/(\binom{n}{k} d^{n-k} 2^{2^m})$*, as long as* $k \log d \geq \log n + \log\log d + 2\log(1/\varepsilon) + O(1)$.

In particular, we note that we can construct extractors for oblivious symbol-fixing sources that even when the min-entropy $k \log d$ is nearly as small as $\log n$, that is, logarithmic in the total possible min-entropy for the source. These extractors extract almost all of the input min-entropy. The result for oblivious bit-fixing sources is a simple corollary of this result obtained by setting $d = 2$.

Corollary 3.1.2. *For oblivious bit-fixing sources with* k *random bits, a function* $f : \{0,1\}^n \to \{0,1\}^m$ *chosen uniformly at random is an* ε*-extractor with output length* $m = k - 2\log(1/\varepsilon) - O(1)$ *with probability at least* $1 - 1/(\binom{n}{k} 2^{n-k} 2^{2^m})$*, as long as* $k \geq \log n + 2\log(1/\varepsilon) + O(1)$.

As before, it is possible to extract when the number of random bits is as nearly as small as $\log n$.

Proof. (Of Theorem 3.1.1.) The number of oblivious symbol-fixing sources is $|\mathcal{X}| = \binom{n}{k} d^{n-k} < 2^n d^{n-k}$. Since each source $X \in \mathcal{X}$ has $H_\infty(X) = k \log d$, the theorem is obtained directly from Theorem 3.0.2, setting $X' = X$ for each $X \in \mathcal{X}$ and using the fact that $\log\log|\mathcal{X}| < \log(n + (n-k)\log d) \leq \log(2n \log d)$. □

We would like to obtain similar results for non-oblivious bit-fixing sources. However, the number of non-oblivious bit-fixing sources over $\{0,1\}^n$ with k random bits is $\binom{n}{k} 2^{(n-k)2^k}$, which is too large to be able to apply Theorem 3.0.2. Even though a random function is not necessarily a good extractor for non-oblivious bit-fixing sources, there are other techniques to bound the limits of extracting from these sources. We will see these techniques in Chapter 5.

Even though a random function doesn't work for non-oblivious bit-fixing sources, it does work for affine sources, which are in fact a special case of non-oblivious bit-fixing sources as well as being a generalization of oblivious bit-fixing sources. For affine sources, we get the following theorem.

Theorem 3.1.3. *For dimension k affine sources on $\{0,1\}^n$, a function $f : \{0,1\}^n \to \{0,1\}^m$ chosen uniformly at random is an ε-extractor with output length $m = k - 2\log(1/\varepsilon) - O(1)$ with probability at least $1 - 1/2^{2^m} 2^{(k+1)n}$, as long as $k \geq \log n + \log k + 2\log(1/\varepsilon) + O(1)$.*

As with oblivious bit-fixing sources, it is possible to extract even when the number of random bits is as nearly as small as $\log n$.

Proof. Let $\mathcal{X}$ be the set of dimension k affine sources on $\{0,1\}^n$. If we think of each bit in an affine source as an affine combination of k random bits, then there are at most 2^{k+1} possibilities for each bit. Thus there are at most $|\mathcal{X}| = 2^{(k+1)n}$ dimension k affine sources on $\{0,1\}^n$. Since each source has min-entropy k, the theorem is obtained directly from Theorem 3.0.2, setting $X' = X$ for each $X \in \mathcal{X}$. □

3.2 Small-Space Sources

Since the probabilities on the edges in small-space sources can be any real number in $[0,1]$, there are an infinite number of such sources, and so we cannot directly apply Theorem 3.0.2. We instead introduce a more restricted model to which we can apply Theorem 3.0.2, and show that general small-space sources are close to convex combinations of this more restricted model. The more restricted model we consider restricts all probabilities to be a multiple of some α.

Definition 3.2.1. An α-approximate space s source is a space s source where the probabilities on all edges are multiples of α.

We'll show that any rate δ small-space source is a convex combination of α-approximate small-space sources, each of which is close to the original source. Thus any extractor that works on α-approximate sources that are close to having rate δ will also be an extractor for rate δ small-space sources.

Lemma 3.2.2. *Let X be a space s source on $\{0,1\}^n$ with min-entropy rate δ. The source X is a convex combination of α-approximate space s sources, each of which has distance at most $\alpha n 2^s$ to X.*

Proof. We can write X as a convex combination of sources X_a such that each X_a is obtained from X by replacing each edge probability p with either $\lfloor \frac{p}{\alpha} \rfloor \alpha$ or $(\lfloor \frac{p}{\alpha} \rfloor + 1)\alpha$.

We will show that X_a is close to X via a hybrid argument. Let X_a^i be the hybrid obtained by the first i bits having probabilities from X_a and the rest of the bits having probabilities from X. So $X = X_a^0$ and $X_a = X_a^n$. Then $|X - X_a| = |\sum_{i=1}^n (X_a^{i-1} - X_a^i)| \leq \sum_{i=1}^n |X_a^{i-1} - X_a^i|$.

For each term $|X_a^{i-1} - X_a^i|$ the only difference is in the probabilities on the edges in the ith layer, which each differ by at most α. We fix i and calculate this distance. Let $v_{i,j}$ denote the jth vertex in the ith layer. Let $q_{i-1,j}$ denote the probability of reaching $v_{i-1,j}$ in X_a and $p^0_{j,j'}$ ($p^1_{j,j'}$) denote the probability on the 0 (1) edge from $v_{i-1,j}$ to $v_{i,j'}$ in X. Then

$$\begin{aligned} |X_a^{i-1} - X_a^i| &\leq \frac{1}{2} \sum_{j,j'} q_{i-1,j}((p^0_{j,j'} + \alpha - p^1_{j,j'}) + (p^1_{j,j'} + \alpha - p^1_{j,j'})) \\ &\leq \alpha \sum_{j'} \sum_{j} q_{i-1,j} = \alpha \sum_{j'} 1 = \alpha 2^s. \end{aligned}$$

So the overall error is bounded by $|X - X_a| \leq \sum_{i=1}^{n} \alpha 2^s = \alpha n 2^s$. □

Lemma 3.2.3. *The number of α-approximate space s sources on $\{0,1\}^n$ is less than $2^{(s+1)2^s n/\alpha}$.*

Proof. First count the number of possible edge configurations from any given vertex. There are 2^{s+1} possible edges, since there is a 0 edge and a 1 edge for each of the 2^s vertices in the next layer. Since all probabilities are multiples of α, there are less than $2^{(s+1)/\alpha}$ ways to allocate probabilities to these edges. Since there are n layers and 2^s vertices at each layer, the total number of possible sources is $2^{(s+1)2^s n/\alpha}$. □

Now we invoke Theorem 3.0.2 to show that a random function is a good extractor for small-space sources.

Theorem 3.2.4. *For space s sources with min-entropy k, a function $f : \{0,1\}^n \to \{0,1\}^m$ chosen uniformly at random is an ε-extractor with output length $m = k - 2\log(1/\varepsilon) - O(1)$ with probability at least $1 - 1/2^{2^m} 2^{(s+1)n^2 2^{2s+1}/\varepsilon}$, as long as $k \geq 2s + 1 + \log(s+1) + 2\log n + 3\log(1/\varepsilon) + O(1)$.*

This theorem says that extractors exist for sources with space almost as large as $k/2$ and with min-entropy as low as $\Theta(\log n)$.

Proof. First apply Lemma 3.2.2 with $\alpha = \varepsilon/n2^{s+1}$ to show that the each small-space source X is a convex combination of α-approximate sources that are $\varepsilon/2$ close to X. Then apply Theorem 3.0.2 to the set of α-approximate sources that are $\varepsilon/2$ close to having min-entropy k, using Lemma 3.2.3 as the bound on the number of such sources (since this set is a subset of all α-approximate space s sources). Since each min-entropy k space s source is a convex combination of these α-approximate sources, the extractors given by Theorem 3.0.2 also work with these sources. □

3.3 Total-Entropy Independent Sources

We can also apply Theorem 3.0.2 to total-entropy independent sources. Similarly to the small-space case, we define an intermediate model to reduce the number of sources.

Definition 3.3.1. An α-approximate independent source $X_1, \ldots, X_r$ on $(\{0,1\}^\ell)^r$ is an independent source such that $\forall y \in \{0,1\}^\ell$ and $\forall i$, $\Pr[X_i = y]$ is a multiple of α.

We use this model rather than flat independent sources because as we saw in Lemma 6.6.4, we can lose a constant fraction of the min-entropy when viewing an independent source as a convex combination of flat independent sources.

This lemma allows us to restrict our attention to α-approximate independent sources. We'll show that any total-rate δ independent-symbol source is a convex combination of α-approximate independent sources, each of which is close to the original source.

Lemma 3.3.2. *Let $X = X_1, \ldots, X_r$ be an total-rate δ independent source on $(\{0,1\}^\ell)^r$. The source X is a convex combination of α-approximate independent sources, each of which has distance at most $\frac{1}{2}\alpha r 2^\ell$ to X.*

Proof. We can write X as a convex combination of sources $X' = X'_1, \ldots, X'_r$ such that each X'_i is obtained from X_i by replacing each output probability $\Pr[X_i = y]$ with either $\lfloor \frac{p}{\alpha} \rfloor \alpha$ or $(\lfloor \frac{p}{\alpha} \rfloor + 1)\alpha$.

Now the distance

$$\begin{aligned} |X' - X| &= \sum_{i=1}^{r} |X'_i - X_i| = \frac{1}{2} \sum_{i=1}^{r} \sum_{x \in (\{0,1\}^\ell)} |\Pr[X'_i = x] - \Pr[X_i = x]| \\ &\leq \frac{1}{2} \sum_{i=1}^{r} \alpha 2^\ell = \frac{1}{2} \alpha r 2^\ell, \end{aligned}$$

where the first inequality is because each string $x \in \{0,1\}^\ell$ contributes at most α error for each X_i. □

Lemma 3.3.3. *The number of α-approximate independent sources on $(\{0,1\}^\ell)^r$ is less than $2^{\frac{1}{\alpha}r\ell}$.*

Proof. Let $X = X_1, \ldots, X_r$ be an α-approximate total-rate δ independent source on $(\{0,1\}^\ell)^r$. Since there are 2^ℓ possible values for each X_i, each of which has a probability that is a multiple of α, there are less then $2^{\frac{\ell}{\alpha}}$ possible distributions for X_i. Thus there are less than $(2^{\frac{\ell}{\alpha}})^r = 2^{\frac{1}{\alpha}r\ell}$ possible distributions for X. □

Now we can apply Theorem 3.0.2 to show that a random function is a good extractor for total-rate δ independent sources.

Theorem 3.3.4. *For total-entropy k independent sources, a function $f : (\{0,1\}^\ell)^r \to \{0,1\}^m$ chosen uniformly at random is an ε-extractor with output length $m = k - 2\log(1/\varepsilon) - O(1)$ with probability $1 - 1/2^{2^m} 2^{r^2\ell 2^\ell/\varepsilon}$ as long as $k \geq \ell + \log\ell + 2\log r + 3\log(1/\varepsilon) + O(1)$.*

Note that the $k > \ell$ is necessary because otherwise all of the entropy could be contained within a single source, which we know is impossible to extract from. Thus, the bound in this theorem is close to the best we could hope for.

Proof. First apply Lemma 3.3.2 with $\alpha = \varepsilon/r2^\ell$ to show that the each total-entropy k independent source X is a convex combination of α-approximate total-entropy k independent sources that are $\varepsilon/2$ close to X. Then apply Theorem 3.0.2 to the set of α-approximate total-entropy k independent sources that are $\varepsilon/2$ close to having min-entropy k, using Lemma 3.3.3 as the bound on the number of such sources (since this set is a subset of all α-approximate independent sources). Since each

total-entropy k independent source is a convex combination of these α-approximate sources, the extractors given by Theorem 3.0.2 also work with these sources. $\square$

Chapter 4

Oblivious Bit-Fixing and Symbol-Fixing Sources

4.1 Our Results

We start off by formally stating our results for extracting from oblivious bit-fixing and symbol-fixing sources. As noted in the introduction, the best previous constructions for extracting from oblivious bit-fixing sources required that at least half of the bits be random [KJS01, BS00]. We are able to improve on these constructions by outputting $\Omega(n^{2\gamma})$ bits when the input has at least $n^{\frac{1}{2}+\gamma}$ random bits.

Theorem 4.1.1. *For any $\gamma > 0$ and any constant $c > 0$, there exists an ε-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of oblivious bit-fixing sources with $n^{\frac{1}{2}+\gamma}$ random bits, where $m = \Omega(n^{2\gamma})$ and $\varepsilon = 2^{-cm}$. This extractor is computable in a linear number of arithmetic operations on m-bit strings.*

We can even extract some bits when there are fewer random bits, although we get a much shorter output.

Theorem 4.1.2. *There exists an* ε*-extractor* $f : \{0,1\}^n \to \{0,1\}^{\frac{1}{4}\log k}$*, for the set of oblivious bit-fixing sources with* k *random bits, where* $\varepsilon = \frac{1}{2}k^{\frac{1}{4}}\exp(-\frac{\pi^2\sqrt{k}}{2})$*. This extractor is computable in a linear number of arithmetic operations on* $\frac{1}{4}\log k$ *bits.*

For d-ary oblivious symbol-fixing sources with $d > 2$, we get much better results than for oblivious bit-fixing sources. We construct an extractor that extracts a constant fraction of the randomness for sources with any number of random symbols, with the constant depending on d. In particular, as d grows large it extracts almost all of the randomness.

Theorem 4.1.3. *For every* $d > 2$ *there exists a* $c_d > 0$ *such that for every* n *and* k*, there exists an* ε*-extractor* $f : [d]^n \to [d]^m$ *for the set of* d*-ary oblivious symbol-fixing sources with* k *random symbols that outputs* $m = c_d\ k - O(\log_d(1/\varepsilon))$ *symbols, where* $c_d \to 1$ *as* $d \to \infty$*. This extractor is computable in a linear number of arithmetic operations on* m*-symbol strings.*

4.2 Overview Of Our Constructions

We now give an overview of our various extractor constructions along with an outline of the rest of the chapter.

Our extractor for d-ary symbol-fixing sources involves using the input symbols to take a random walk on a d-regular expander graph, starting from an arbitrary fixed start vertex. The extractor then outputs the label of the final vertex on the walk. We show that even though we allow some of the steps to be fixed in advance, corresponding to the fixed bits of the source, these steps will not hurt us. Therefore the random walk behaves essentially like a random walk on the random steps only. We use well known relationships between the eigenvalues of the transition matrix of the

random walk and the distance to uniform to measure the number of steps it takes to "mix" close to uniform. The eigenvalues of expander graphs are such that this mixing is quite rapid, and so we can extract a linear fraction of the entropy, thus proving Theorem 4.1.3. For $d = 2$, we cannot use an expander graph since expanders only exist for degree $d > 2$, but we show that if we take a random walk on a cycle we can still extract some bits out, proving Theorem 4.1.2. We give these constructions in Section 4.3.1. We also note that similar types of random walks have been used in previous pseudorandomness constructions [AKS87, CW89, IZ89].

For oblivious bit-fixing sources, we show that we can extract even more bits by first converting the sources into sources that are close to oblivious symbol-fixing sources, which we call approximate oblivious symbol-fixing sources, and then applying the expander walk extractor. This gives the extractor from Theorem 4.1.1. We show in Section 4.3.2 that our extractor for oblivious symbol-fixing sources also works for approximate oblivious symbol-fixing sources. To convert the oblivious bit-fixing source into a d-ary approximate oblivious symbol-fixing source, we partition the input into blocks. For each block, we take a random walk on the d-cycle and output the label of the final vertex. Enough of the blocks will have enough random bits so that enough of the symbols are almost random. We note that the symbols in the output source have constant error, so we can't just add the errors from the almost random steps since they are too large. Because of this conversion step, we "lose" some of the randomness, which is why we require that the number of random bits be greater than $\sqrt{n}$ in Theorem 4.1.1. In Section 4.4, we show how to do the conversion and prove that the extractor works.

In Section 4.5, we show the relation between our extractors for oblivious bit-fixing sources and exposure-resilient cryptography. Specifically, we show that our extractors for oblivious bit-fixing sources are also adaptive statistical ERF's.

4.3 Constructing Extractors for Oblivious Symbol-Fixing and Approximate Oblivious Symbol-Fixing Sources

In this section, we first show how to construct deterministic extractors for oblivious symbol-fixing sources. We will then show how this construction can be extended to extract from approximate oblivious symbol-fixing sources. We will use the construction for approximate oblivious symbol-fixing sources in the next section to show how we can extract from oblivious bit-fixing sources.

4.3.1 Extracting From Oblivious Symbol-Fixing Sources

In this section, we prove the following generalization of Theorem 4.1.3 to show that we can extract a constant fraction of the randomness from oblivious symbol-fixing sources.

Theorem 4.3.1. *For any $k = k(n)$, ε and $d > 2$, if there exists an efficiently computable d-regular expander with $\lambda(P) \leq d^{-\alpha}$ on d^m vertices, for $m \leq 2\alpha k - \frac{2}{\log d}\log\frac{1}{2\varepsilon}$, then there exists an efficiently computable ε-extractor which outputs m symbols for the set of oblivious symbol-fixing sources on $[d]^n$ with k random symbols.*

The extractor works by taking a walk on an expander with d^m vertices starting at a fixed vertex and using the input symbols as steps. The output is the label of the final vertex.

We get extractors with the longest output length when we use Ramanujan expanders, for which $\lambda(P) = 2\sqrt{(d-1)}/d$. For certain parameters, there exist efficiently computable Ramanujan graphs [LPS88, Lub94]. Note that for Ramanujan

graphs, as d grows large, α approaches $1/2$, so the output length approaches k.

For $d = 2$, we can't use an expander, but we can use the symbols to take a walk on the cycle to get an extractor for oblivious bit-fixing sources that extracts a small number of bits from any source regardless of k. Note that we're restricted to using odd size cycles here, since random walks on even cycles don't converge to uniform, as they alternate between the even and odd vertices.

Theorem 4.3.2. *For odd d, there exists an ε-extractor $f : \{0,1\}^n \to [d]$, for the set of oblivious bit-fixing sources on $\{0,1\}^n$ with k random bits, where $\varepsilon = \frac{1}{2}\sqrt{d}\exp(-\frac{\pi^2 k}{2d^2})$. This extractor is computable in a linear number of arithmetic operations on $\log d$ bits.*

Note that for this extractor to be useful, we must have $\log d < \frac{1}{2}\log k$, which shows that we can output only a small amount of the original randomness with this technique. In particular, if we take $d = k^{\frac{1}{4}}$, we get Theorem 4.1.2.

Both Theorem 4.3.1 and 4.3.2 arise from the following key lemma.

Lemma 4.3.3. *Let P be a uniform transition matrix with stationary distribution π for an undirected non-bipartite d-regular graph G on M vertices. Consider an n step walk on G, with the steps taken according to the symbols from an oblivious symbol-fixing source X on $[d]^n$ with k random symbols. For any initial probability distribution $p = v + \pi$, the distance from uniform at the end of the walk is bounded by*

$$|p\prod_{i=1}^{n} P_i - \pi| \leq \frac{1}{2}\|p\prod_{i=1}^{n} P_i - \pi\|\sqrt{M} \leq \frac{1}{2}\lambda(P)^k\sqrt{M}.$$

To prove this lemma, we show that the random symbols from the source bring us closer to uniform and also that the fixed symbols don't bring us any further away.

For the random steps, it is well known that the distance can be bounded in terms of $\lambda(P)$. This gives the following lemma, a proof of which can be found in, e.g., [Lov96].

Lemma 4.3.4. *Let P be a uniform transition matrix for an undirected, d-regular graph G. Then for any probability vector $p = v + \pi$,*

$$\|pP - \pi\| \leq \lambda(P)\|v\|.$$

In our case, most of the steps in our random walks will be fixed. The consistent labeling property ensures that the transition matrix for these fixed steps will be a permutation matrix. Thus these steps leave the distance from uniform unchanged, and so we get the following lemma.

Lemma 4.3.5. *Let P be a transition matrix for a fixed step on an undirected, d-regular graph G. Then for any probability vector $p = v + \pi$,*

$$\|pP - \pi\| = \|v\|.$$

Now, using the previous two lemmas, we can prove Lemma 4.3.3.

Proof. (Of Lemma 4.3.3.) For the random symbols we can apply Lemma 4.3.4. Since there are k random symbols, this gives us the $\lambda(P)^k$ factor. We also use that by Lemma 4.3.5 the steps corresponding to the fixed symbols don't increase the distance from uniform. Combining both the random and the fixed steps together with the relation between the variation and ℓ_2 distance and the fact that the $\|v\| \leq 1$, we get the stated bound. □

Now we can use Lemma 4.3.3 to prove Theorem 4.3.1.

Proof. (Of Theorem 4.3.1.) We can apply Lemma 4.3.3, where in this case $\lambda(P) \leq d^{-\alpha}$ and $M = d^m$. Thus the error $\varepsilon \leq \frac{1}{2} d^{-\alpha k + (m/2)}$. Taking logarithms and solving for m, we get the stated bound on m. □

Now, using Lemma 4.3.3, we can prove Theorem 4.3.2. We first separate out the following lemma which will be useful later.

Lemma 4.3.6. *Let P be a uniform transition matrix for the random walk on the d-cycle for d odd. Suppose the length of the walk is n, with the steps taken according to the symbols from an oblivious bit fixing source X on $\{0,1\}^n$ with k random bits. For any initial probability distribution $p = v + \pi$, the distance from uniform at the end of the walk is bounded by*

$$|p\prod_{i=1}^{n} P_i - \pi| \leq \frac{1}{2}\|p\prod_{i=1}^{n} P_i - \pi\|\sqrt{d} \leq \frac{1}{2}(\cos(\pi/d))^k\sqrt{d}.$$

Proof. By Lemma 4.3.3 and the fact that the d-cycle has $\lambda(P) = \cos(\pi/d)$ (see [Dia88]). □

Proof. (Of Theorem 4.3.2.) The extractor outputs the result of a random walk on the d-cycle. By Lemma 4.3.6, this will be within $\frac{1}{2}\sqrt{d}(\cos(\pi/d))^k$ of uniform. Since $\cos(\pi/d) \leq \exp(-\frac{\pi^2}{2d^2})$ (see [Dia88], p. 26), we get the desired error. □

There is one slight difficulty, since we may want to use a family of expander graphs (or cycles) that includes graphs that don't have exactly 2^m vertices. (In fact, in the cycle case, we can't use any even sized cycle.) This difficulty can be overcome by outputting the result of the random walk on a much larger graph modulo 2^m. The following lemma shows that doing so has little impact on the error.

Lemma 4.3.7. *If a random variable X is within ε of uniform over $[N]$, then the random variable $Y = X \mod M$ is within $\varepsilon + 1/r$ of uniform over $[M]$, where $r = \lfloor N/M \rfloor$.*

Proof. Divide the $y \in [M]$ up into two classes, those corresponding to r different $x \in [N]$ with $y = x \mod M$ and those corresponding to $r+1$ different $x \in [N]$. The probability that Y assigns to each y is then either r/N or $(r+1)/N$, plus the corresponding part of the original error ε. Since $r/N \leq 1/M \leq (r+1)/N$, the additional error introduced for each y when going from X to Y is at most $1/N$. So the total additional error introduced is at most $M/N \leq 1/r$. □

4.3.2 Extracting From Approximate Oblivious Symbol-Fixing Sources

We now show how the previous construction can be extended to handle the case of approximate oblivious symbol-fixing sources. Our main result in this section is the following variant of Lemma 4.3.3 for approximate oblivious symbol-fixing sources.

Lemma 4.3.8. *Let P be a uniform transition matrix with stationary distribution π for an undirected non-bipartite d-regular graph G on M vertices. Suppose we talk a walk on G for n steps, with the steps taken according to the symbols from an (k, ε)-approximate oblivious symbol-fixing source X on $[d]^n$. For any initial probability distribution $p = v + \pi$, the distance from uniform at the end of the walk is bounded by*

$$|p\prod_{i=1}^{n} P_i - \pi| \leq \frac{1}{2}\|p\prod_{i=1}^{n} P_i - \pi\|\sqrt{M} \leq \frac{1}{2}(\lambda(P) + \varepsilon\sqrt{d})^k\sqrt{M}.$$

In the case of approximate oblivious symbol-fixing sources, the random steps in our random walk will be only almost uniformly random. This introduces

some small amount of error into our transition matrix. We can separate out the error terms by dividing up our new transition matrix P' into the uniform transition matrix P and an error matrix E, which is defined as follows.

Definition 4.3.9. An ε-error matrix E for a d-regular graph G is a matrix with the following properties. If $|E_{ij}| > 0$, then (i, j) is an edge in G; all of the columns of E sum to 0; and the ℓ_2 norm of each column of E is at most ε.

For slightly non-uniform random steps, we can modify the bound from Lemma 4.3.4 slightly to get the following lemma.

Lemma 4.3.10. *Let P be a uniform transition matrix for an undirected, d-regular graph G. Let E be an ε-error matrix for G. Now let $P' = P + E$ be our modified transition matrix. Then P' has the same stationary distribution π as P and for any probability vector $p = v + \pi$,*

$$\|pP' - \pi\| \leq (\lambda(P) + \varepsilon\sqrt{d})\|v\|.$$

Proof. Because π is uniform and because each of the columns of E sum to 0 by definition, $\pi E = 0$. Thus $\pi P' = \pi P + \pi E = \pi$ by the above observation combined with the stationarity of π with respect to P. Thus P' has stationary distribution π.

Now we bound $\|pP' - \pi\|$. We first observe that

$$\|pP' - \pi\| = \|vP' + \pi P' - \pi\| = \|vP'\|$$

since we know from above that π is stationary. Now we can focus on bounding $\|vP'\|$. By the triangle inequality $\|vP'\| \leq \|vP\| + \|vE\|$. We know that $\|vP\| \leq$

$\lambda(P)\|v\|$. Letting e_{ij} denote the entries of E, we get

$$\begin{aligned}\|vE\| &= \left(\sum_j \left(\sum_{i;e_{ij}\neq 0} e_{ij}v_i\right)^2\right)^{\frac{1}{2}} \\ &\leq \left(\sum_j \left(\sum_{i;e_{ij}\neq 0} e_{ij}^2\right)\left(\sum_{i;e_{ij}\neq 0} v_i^2\right)\right)^{\frac{1}{2}} \\ &\leq \varepsilon\left(\sum_j \sum_{i;e_{ij}\neq 0} v_i^2\right)^{\frac{1}{2}} \leq \varepsilon\sqrt{d}\|v\|\end{aligned}$$

where the first line is simply from the definition, and noting that we only need to sum over all non-zero e_{ij}. The second line follows from the Cauchy-Schwarz inequality. The third line follows from the fact that the sum of the square of the errors e_{ij}^2 over any column is at most ε^2. The final inequality comes from the fact that e_{ij} can only be non-zero when ij corresponds to an edge in G. Since there are d edges adjacent to i, we will have at most d v_i^2 terms in the sum for each i.

Putting everything together we get the desired bound on $\|pP' - \pi\|$. □

Unlike in the case of oblivious symbol-fixing sources, the non-random steps may not be fixed, but may simply not have enough randomness in them. However, we would still like to show that these steps do not take us further from the uniform distribution. The following lemma shows that since any step chosen according to a symbol from a d-ary source is a convex combination of permutations, the non-random steps in our random walk don't increase the distance from uniform. Note that this result depends on our assumption that the graph G is consistently labeled.

Lemma 4.3.11. *Let P be a transition matrix for a step chosen according to a symbol X_j from a d-ary source X. Then P is a convex combination of permutation matrices*

and for any probability vector $p = v + \pi$, $\pi P = P$ *and* $\|pP - \pi\| \leq \|v\|$.

Proof. First we show that P is a convex combination of permutation matrices. Every possible value from $i \in [d]$ for x gives a permutation matrix P_i. If X_j is distributed with probabilities α_i for each $i \in [d]$, then $P = \sum_{i=0}^{d-1} \alpha_i P_i$, which is a convex combination of permutation matrices.

Then note that since any permutation of π is still uniform, we have $\pi P_i = \pi$ and thus $\pi P = P$. This gives us $\|pP - \pi\| = \|vP\|$. We bound $\|vP\|$ by the triangle inequality as $\|vP\| \leq \sum_i \alpha_i \|vP_i\| = \sum_i \alpha_i \|v\| = \|v\|$, where the second inequality follows from the fact that since P_i is a permutation, $\|vP_i\| = \|v\|$. □

Using the previous two lemmas, we can prove Lemma 4.3.8.

Proof. Let P_i be the transition matrix of the random walk at the i'th step. By Lemma 4.3.11 P_i is a convex combination of permutation matrices and $\pi P_i = \pi$. This gives us $\pi \prod_{i=1}^{n} P_i = \pi$, so $p \prod_{i=1}^{n} P_i - \pi = v \prod_{i=1}^{n} P_i$.

Let $v_j = \prod_{i=1}^{j} P_i$. Then $v_j = v_{j-1} P_j$, and $v_0 = v$. For k of the steps, the symbols are within an ℓ_2 distance of ε from uniform, which implies $P_j = P + E_j$, where every column of E_j has ℓ_2 norm at most ε. Since G is consistently labeled, the sum of each column of E_j is equal to 0, so E_j is indeed an error matrix. So for these steps, by Lemma 4.3.10, $\|v_{j-1} P_j\| \leq (\lambda(P) + \varepsilon\sqrt{d})\|v_{j-1}\|$. For the other steps, we still have by Lemma 4.3.11 that $\|v_{j-1} P_j\| \leq \|v_{j-1}\|$. So for k steps the ℓ_2 norm is reduced while for the rest of the steps it, at worst, remains the same. Thus

$$\|p \prod_{i=1}^{n} P_i - \pi\| = \|v \prod_{i=1}^{n} P_i\| \leq (\lambda(P) + \varepsilon\sqrt{d})^k \|v\|.$$

Now apply the bound relating the ℓ_2 norm and variation distance and $\|v\| \leq 1$. □

4.4 From Oblivious Symbol-Fixing Sources to Oblivious Bit-Fixing Sources

In this section, we show how to extend our results for oblivious symbol-fixing sources to oblivious bit-fixing sources to get the following theorem, which is basically a restatement of Theorem 4.1.1. Though we state the theorem for general values of δ, we have in mind the case $\delta n = n^{\frac{1}{2}+\gamma}$.

Theorem 4.4.1. *For any positive $\delta = \delta(n) \leq 1$ and any constant $c > 0$, there exists an ε-extractor $f : \{0,1\}^n \to \{0,1\}^m$, for the set of oblivious bit-fixing sources on $\{0,1\}^n$ with δn random bits, where $m = \Omega(\delta^2 n)$ and $\varepsilon = 2^{-cm}$. This extractor is computable in a linear number of arithmetic operations on m-bit strings.*

There are two main steps in the extractor construction. First, we transform the source into an approximate oblivious symbol-fixing source by dividing it into blocks. For each block we take a random walk on the cycle and output the label of the final vertex on the walk. The approximate oblivious symbol-fixing source is then the concatenation of these outputs. Then we use the expander walk extractor from the previous section to extract from the approximate oblivious symbol-fixing source.

We start by applying Lemma 4.3.6 to our degree 2 walks on the d-cycle for each of the blocks. We will show that enough of the blocks mix to within ε' of the uniform distribution, for some ε'. This process gives us an approximate oblivious symbol-fixing source.

Lemma 4.4.2. *For any odd d, any oblivious bit-fixing source on $\{0,1\}^n$ with δn random bits can be deterministically converted into a $(\delta^2 n/4t, \varepsilon)$-approximate oblivious symbol-fixing source on $[d]^{\delta n/2t}$, where $t = \lceil \frac{\log \varepsilon}{\log(\cos(\pi/d))} \rceil$.*

The almost random symbols in the approximate oblivious symbol-fixing source correspond to blocks where we have "enough" random bits. Using a Markov-like argument, we can quantify how many such blocks we will have, as shown in the following lemma.

Lemma 4.4.3. *Suppose we have n bits from an oblivious bit-fixing source, where $k = \delta n$ of the bits are random. For any partition of the n bits into $\delta n/2t$ blocks of size $2t/\delta$, the number r of blocks with at least t random bits satisfies $r > \frac{\delta^2 n}{4t}$.*

Proof. We know that in the r blocks with at least t random bits there are at most $2t/\delta$ random bits. In the remaining blocks there are less than t random bits. Combining these two facts we get that the total number of random bits $k < 2rt/\delta + t((n\delta/2t) - r)$, which after a simple calculation gives the desired result. □

Using this lemma, we can now prove Lemma 4.4.2.

Proof. (Of Lemma 4.4.2.) Divide the input r up into $\delta n/2t$ blocks of size $2t/\delta$. Then take a random walk on a d-cycle using the bits from each block and output the vertex label of the end vertex for each walk. These vertex labels are the symbols for our approximate oblivious symbol-fixing source. We call a block good if this random walk reaches within an ℓ_2 distance of ε from uniform, which means the corresponding symbol is good for our source. By Lemma 4.3.6, if there are at least t random bits in the block the ℓ_2 distance from uniform is at most $(\cos(\pi/d))^t \leq \varepsilon$, which means all such blocks are good. Then by Lemma 4.4.3, the number of good blocks r satisfies $r > \frac{\delta^2 n}{4t}$. Thus the output source is an approximate oblivious symbol-fixing source with the appropriate parameters. □

The symbols from the approximate oblivious symbol-fixing source then correspond to our almost random steps in the expander graph, so we can apply Lemma 4.3.8

to the expander walk to get that the final distribution is close to uniform.

Proof. (Of Theorem 4.4.1.) If $\delta = O(1/\sqrt{n})$, we can take f to be the parity function, since in this case outputting a single bit is enough. Otherwise, let G be a d-regular expander graph on 2^m vertices with uniform transition matrix P. Choose ε' so that $\lambda_{\varepsilon'} = \lambda(P) + \varepsilon'\sqrt{d} < 1$. Then use the procedure in Lemma 4.4.2 to convert the oblivious bit-fixing source on $\{0,1\}^n$ with δn random bits to a $(\frac{\delta^2 n}{4t}, \varepsilon')$-approximate oblivious symbol-fixing source on $[d]^{\frac{\delta n}{2t}}$, where
$t = \lceil (\log \varepsilon')/(\log(\cos(\pi/d))) \rceil$.

Now we use the approximate oblivious symbol-fixing source to take a random walk on G. We take the label of the final vertex of the walk on G as the output $f(r)$. Then we can apply Lemma 4.3.8, which gives that the variation distance from uniform of $f(r)$ is at most

$$\frac{1}{2}\lambda_{\varepsilon'}^r 2^{m/2} < \lambda_{\varepsilon'}^{\frac{\delta^2 n}{4t}} 2^{m/2}.$$

We want this to be at most $\varepsilon = 2^{-cm}$, so setting $m = b\delta^2 n$ for some constant $b > 0$ and taking the logarithm, we get $\frac{1}{4t}\log\frac{1}{\lambda_{\varepsilon'}} \geq b\left(c + \frac{1}{2}\right)$. The left hand side of this inequality is just some positive constant, so for any given value of c we can select b so that the inequality is satisfied. These constants give the desired output length and the desired error ε.

Since there are a linear number of expander steps and there exist expanders that take a constant number of arithmetic operations per step, f is computable in a linear number of arithmetic operations on m-bit strings. □

Note that in the last proof we only needed a bound on the ℓ_2 distance, which from the proof of Lemma 4.3.8 is tighter than the bound on the variation distance,

but this difference only affects the constants in the theorem.

4.5 Exposure-Resilient Cryptography

We now discuss the needed background from exposure-resilient cryptography and how our extractor for oblivious bit-fixing sources can be used to get better statistical adaptive ERF's and AONT's.

There are a few different types of resilient functions that we define, taken from [DSS01], each of which involve making the output look random given an adversary with certain abilities. For all of these definitions, f is a polynomial time computable function $f : \{0,1\}^n \rightarrow \{0,1\}^m$. Also, there is a computationally unbounded adversary $\mathcal{A}$ that has to distinguish the output of f from a uniformly random string $R \in \{0,1\}^m$. A function $\varepsilon(n)$ is said to be *negligible* if $\varepsilon(n) = O(\frac{1}{n^c})$ for all constants c.

Adaptive k-ERFs are defined as functions that remain indistinguishable from uniform even by adversaries that can adaptively read most of the input.

Definition 4.5.1. [DSS01] An adaptive k-ERF is a function f where, for a random input r, when $\mathcal{A}$ can adaptively read all of r except for k bits, $|\Pr[\mathcal{A}^r(f(r)) = 1] - \Pr[\mathcal{A}^r(R) = 1]| \leq \varepsilon(n)$ for some negligible function $\varepsilon(n)$.

Our goal is to construct adaptive ERF's. We might first think that any $\varepsilon(n)$-extractor for oblivious bit-fixing sources would work as long as $\varepsilon(n)$ is negligible. However, [DSS01] show that there are functions that are oblivious bit-fixing extractors but not adaptive ERF's. To solve this problem, they use a stronger condition which they show is sufficient. This condition is that every single output value has to occur with almost uniform probability. Functions that satisfy this stronger condition

are known as almost-perfect resilient functions (APRFs), introduced by Kurosawa et al. [KJS01].

Definition 4.5.2. [KJS01] A $k = k(n)$ almost-perfect resilient function (APRF) is a function f where, for any setting of $n-k$ bits of the input r to any fixed values, the probability vector p of the output $f(r)$ over the random choices for the k remaining bits satisfies $|p_i - 2^{-m}| < 2^{-m}\varepsilon(n)$ for all i and for some negligible function $\varepsilon(n)$.

Theorem 4.5.3. *[DSS01] If f is a k-APRF, then f is an adaptive k-ERF.*

The following lemma shows that any extractor for oblivious bit-fixing sources with small enough error is also an APRF. We use this lemma to show that the extractor we constructed earlier is also an APRF, and hence an adaptive k-ERF.

Lemma 4.5.4. *Any $2^{-m}\varepsilon(n)$-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of oblivious bit-fixing sources on $\{0,1\}^n$ with k random bits, where $\varepsilon(n)$ is negligible, is also a k-APRF.*

Proof. Since f is an extractor, the total variation distance from uniform of the output of f when $n-k$ bits of the input are fixed is within $2^{-m}\varepsilon(n)$. Thus the distance of any possible output from uniform must also be within $2^{-m}\varepsilon(n)$, and the APRF property is satisfied. □

Now using this lemma we get the following theorem.

Theorem 4.5.5. *For any positive constant $\gamma \leq 1/2$, there exists an explicit k-APRF $f : \{0,1\}^n \to \{0,1\}^m$, computable in a linear number of arithmetic operations on m-bit strings, with $m = \Omega(n^{2\gamma})$ and $k = n^{\frac{1}{2}+\gamma}$.*

Proof. Apply Lemma 4.5.4 to the extractor from Theorem 4.4.1, choosing $c > 1$. □

We can use adaptive ERFs to construct all-or-nothing transforms (AONTs), which were introduced by Rivest [Riv97] and extended to adaptive adversaries by Dodis et al. [DSS01]. We first give a formal definition of AONTs. There are two parts to the definition. First, the AONT is an efficient randomized mapping that is easily invertible given the entire output. Second, an adversary gains negligible information about the input to the AONT even when it can read almost the entire output. This is formalized by the adversary not being able to distinguish between any two distinct inputs. Note that the output of the AONT has two parts. We call the first part of the output the secret part and the second part of the output the public part.

Definition 4.5.6. [DSS01] A polynomial time randomized transformation $T : \{0,1\}^m \to \{0,1\}^s \times \{0,1\}^p$ is a statistical adaptive k-AONT if

1. T is invertible in polynomial time.

2. For any adversary $\mathcal{A}$ who has oracle access to string $y = (y_s, y_p)$ and is required not to read at least k bits of y_s, and for any $x_0, x_1 \in \{0,1\}^m$ and some negligible function $\varepsilon(s+p)$:

$$|\Pr[\mathcal{A}^{T(x_o)}(x_0,x_1) = 1] - \Pr[\mathcal{A}^{T(x_1)}(x_0,x_1) = 1]| \leq \varepsilon(s+p).$$

The following lemma from [DSS01] relates adaptive k-ERF's to adaptive k-AONT's, and shows that our construction gives adaptive k-AONT's.

Theorem 4.5.7. *[DSS01] If $f : \{0,1\}^n \to \{0,1\}^m$ is an adaptive k-ERF, then $T(x) = \langle r, x \oplus f(r) \rangle$ is a statistical adaptive k-AONT with secret part r and public part $x \oplus f(r)$.*

Combining Theorem 4.5.7 with Theorem 4.5.5, we get the following theorem.

Theorem 4.5.8. *For any positive constant $\gamma \leq 1/2$, there exists an explicit function $f : \{0,1\}^n \to \{0,1\}^m$ computable in a linear number of arithmetic operations on m-bit strings, with $m = \Omega(n^{2\gamma})$, such that $T(x) = \langle r, x \oplus f(r) \rangle$ is a statistical adaptive k-AONT with secret part r and public part $x \oplus f(r)$.*

4.6 Subsequent Work and Open Questions

Subsequent to our work, Gabizon, Raz, and Shaltiel [GRS04] have improved upon our constructions of extractors for oblivious bit-fixing sources. The extractors they construct are able to extract almost all of the random bits from oblivious bit-fixing sources that have min-entropy $k > \sqrt{n}$, with exponentially small error. Because their extractor for large min-entropy is nearly optimal, the main area for improvement is then constructing a better extractor for small min-entropy. For small min-entropy, they give a different construction which extracts $\Omega(k)$ bits as long as $k > \log^c n$ for some constant c [GRS04]. However, in this case the error is much higher at $k^{-\Omega(1)}$. In contrast, our extractor for small min-entropy only extracts $\Omega(\log k)$ bits but has exponentially small error. So between these two constructions we have a tradeoff between the output length and the error. An interesting open question is to find an explicit construction for small min-entropy ($k \leq \sqrt{n}$) that eliminates this tradeoff and has both large output length and small error.

Also, all of the extractors from [GRS04] have error too large to be used to get adaptive AONTs. Because of this, it would be interesting to find ways to improve upon our extractors in terms of output length, while still having small enough error to get adaptive AONTs. Although it would be nice to have such improved extractors for small k, we wish to emphasize that even the extractor for $k > \sqrt{n}$ from [GRS04] is inadequate for constructing adaptive AONTs, so even an improvement in the high

min-entropy case would be quite interesting.

Chapter 5

Non-Oblivious Bit-Fixing Sources

5.1 Overview Of Our Results

In this chapter, we switch our focus to non-oblivious bit-fixing sources, where the fixed bits can depend on the random bits. We give upper and lower bounds for extracting from such sources.

Previous bounds on non-oblivious bit-fixing sources have been defined in terms of the "influence" of variables on a function [BOL90]. The influence of a set of variables S on a function f, denoted $I_f(S)$, is the probability that if the variables not in S are chosen randomly, the function remains undetermined. In this chapter, we show that the influence of a function is related to the variation distance of the function from uniform when the input comes from a non-oblivious bit-fixing source. We use this connection to give both explicit construction of extractors as well as impossibility results showing that we can't do significantly better than these explicit constructions.

In particular, in Section 5.2, we show that if for a balanced function f

$I_f(S) \leq \varepsilon$ for all sets S of ℓ variables, then f is an ε-extractor for the set of length n non-oblivious bit-fixing sources with $n-\ell$ random bits. We also show how any boolean function with low influence can be used to get a function of longer output length which also has low influence, and hence is an extractor. The idea is simply to divide the input up into blocks and apply the low influence boolean function separately to each block, then concatenate the outputs. The best completely explicit boolean low influence function known is the iterated majority function of Ben-Or and Linial [BOL90]. Using this function, we get an ε-extractor for the set of length n non-oblivious bit-fixing sources with $n-\ell$ random bits which extracts $(\varepsilon/\ell)^{1/\log_3 2} n$ bits.

In Section 5.3, we show that the converse to our previous result on influence holds (up to constant factors), so that if a function has high influence for some set, then it cannot be an extractor. We also generalize an edge-isoperimetric bound on the hypercube that [BOL90] used for single bit functions to the case of longer output lengths. This bound shows that any function which is an extractor must have high influence for some set. So combining these two results, we show that at most $O(n\varepsilon/\ell)$ bits can be extracted from non-oblivious bit-fixing sources.

5.2 Explicit Constructions

To get explicit extractor constructions for non-oblivious bit-fixing sources, we use the following lemma that shows that having low influence for all sets of a given size implies that a function is an extractor.

Lemma 5.2.1. *Suppose $f : \{0,1\}^n \to \{0,1\}^m$ maps the uniform distribution U_n to U_m and $I_f(S) \leq \varepsilon$ for all sets S of ℓ variables. Then f is an ε-extractor for the set of non-oblivious bit-fixing sources on $\{0,1\}^n$ with $n-\ell$ random bits.*

Proof. Let X be a non-oblivious bit-fixing source on $\{0,1\}^n$ with $n-\ell$ random bits and let S denote the set of of fixed variables of X. Since $I_f(S) \leq \varepsilon$, for all but an ε fraction of the choices for the random bits in X, f has the same distribution regardless of whether the rest of the bits are chosen according to X or according to U_n. Thus the variation distance is at most ε. □

Using Lemma 5.2.1, we immediately get that known constructions of boolean functions with low influence [BOL90, AL93] are extractors. To get longer output length, we show that we can construct an extractor that extracts several bits from any boolean function with small influence. The extractor simply works by applying the low influence function to blocks of the input and concatenating the resulting output bits.

Lemma 5.2.2. *Suppose there exists a function $g : \{0,1\}^s \to \{0,1\}$, with expectation $1/2$, and a value $r(s)$ such that any set S of $\ell(s,\varepsilon) = \varepsilon r(s)$ variables has $I_g(S) \leq \varepsilon$ for all $\varepsilon > 0$. Then there exists an ε-extractor $f : \{0,1\}^n \to \{0,1\}^m$ that extracts $m = n/s$ bits for the set of non-oblivious bit-fixing sources on $\{0,1\}^n$ with $n-\ell(s,\varepsilon)$ random bits.*

Proof. Divide the input into $m = n/s$ blocks of size s. The jth output bit of f will be g applied to the jth block. Fix a set S. By Lemma 5.2.1 we need to show that f has $I_f(S) \leq \varepsilon$ for all sets S of $\ell = \ell(s,\varepsilon)$ variables. Let ℓ_i be the number of bits in S in block i and set $\varepsilon_i = \ell_i / r(s)$. The influence for each output bit is then at most ε_i. Now we note that since the random bits for each of these functions are chosen independently, the total influence is at most the sum of the influences for each of these boolean functions. Thus, since $\sum_{i=1}^m \varepsilon_i = (\sum_{i=1}^m \ell_i)/r(s) = \ell/r(s) = \varepsilon$, $I_f(S) \leq \varepsilon$. □

We can apply this lemma to the iterated majority function of Ben-Or and Linial to get an explicit extractor for non-oblivious bit-fixing sources.

Theorem 5.2.3. *[BOL90] For every s, there is an explicit construction of functions $g : \{0,1\}^s \to \{0,1\}$, with expectation $1/2$, where any set S of $\ell(s,\varepsilon) = \varepsilon\left(\frac{s}{3}\right)^{\alpha}$ variables has $I_g(S) \leq \varepsilon$ for every $\varepsilon > 0$, where $\alpha = \log_3 2$.*

Theorem 5.2.4. *For every n, we can construct an ε-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of non-oblivious bit-fixing sources on $\{0,1\}^n$ with $n - \ell$ random bits. The extractor outputs $m = \frac{1}{3}(\varepsilon/\ell)^{1/\alpha} n$ bits, where $\alpha = \log_3 2$.*

Proof. Apply Lemma 5.2.2 using the function from Theorem 5.2.3. □

Ajtai and Linial [AL93] give hope for improvement since their functions allow $\Omega(s/\log^2 s)$ fixed bits. However, their construction is non-explicit, and a bound like that in Lemma 5.2.2 is only known to hold for $\varepsilon \geq 1/\operatorname{polylog}(s)$ [RZ01].

5.3 Impossibility Results

In this section, we show that it is impossible to do significantly better than the explicit constructions in the previous section. Specifically, we show that at most n/ℓ bits can be extracted from non-oblivious bit-fixing sources. As before, we will use the connection between influence and extraction. In particular, we need the following lemma that shows that a function that has a set with high influence cannot be an extractor.

Lemma 5.3.1. *Let S be a set of ℓ variables. If, for some $\varepsilon > 0$, $I_f(S) = \varepsilon$, then there exists a non-oblivious bit-fixing source X on $\{0,1\}^n$ with $n - \ell$ random bits and with set of fixed variables S so that $|f(X) - U_m| \geq \varepsilon/4$.*

Proof. View the possible outputs as vertices of a hypergraph on 2^m vertices. Consider all possible values of the $n-\ell$ bits not in S. Since $I_f(S) = \varepsilon$, we know that an ε fraction of these values leave f undetermined. For each such value, place a hyperedge between all possible output values of f (when going over all possible values for the bits in S).

Eliminate all of the vertices with no edges. Now divide all of the remaining vertices at random into two sets of equal size, A and B. The expected number of hyperedges in the cut between A and B is at least half the total number of hyperedges, so there exists a pair of sets with at least this many hyperedges. Consider such A and B, and look at only the hyperedges in the cut. Now each of these hyperedges corresponds to a setting of the $n-\ell$ bits not in S. So we define two non-oblivious bit-fixing sources X_A and X_B based on how the values of the bits in S are set for each cut hyperedge. Define X_A (X_B) by setting the bits in S for each cut hyperedge so that the output of f lies in A (B). Since these hyperedges have total probability at least $\varepsilon/2$, these sources will differ by at least $\varepsilon/2$. Thus at least one of them will differ by at least $\varepsilon/4$ from the uniform distribution. □

We also need to generalize the edge-isoperimetric bound from [BOL90].

Lemma 5.3.2. *For every function $f : \{0,1\}^n \to \{0,1\}^m$ with output within ε of uniform on uniform input, the expected influence over all sets of variables S of size ℓ is at least $1 - 2\frac{\binom{n-m+1}{l}}{\binom{n}{l}} - 2\varepsilon$.*

Proof. View all 2^n possible inputs as vertices of the n dimensional cube. Color the vertices of the cube with 2^m colors, where the color of x corresponds to $f(x)$. Now for each possible set S of size ℓ and setting of the remaining $n-\ell$ random variables, there is a corresponding subcube of dimension ℓ in the cube. Note that f is undetermined over such a subcube if and only if the subcube is not monochromatic. So

the average influence over all possible S is the probability that a randomly chosen ℓ dimensional subcube is not monochromatic. We divide the set of colors into two classes, those with at most 2^{n-m+1} vertices and those with more, which we call "small" and "large".

First, we handle the large colors. Let t be the number of large colors. Each of these t colors contributes at least 2^{-m} to the error ε of f with uniform input, so $t \leq \varepsilon 2^m$. Since the distance from uniform is at most ε, the total number of vertices with large colors is at most $\varepsilon 2^n + t2^{n-m} \leq 2\varepsilon 2^n$. The probability that a subcube is monochromatic for a large color is at most the probability that the subcube lies completely within this set of vertices, which is at most the probability that any given vertex in the subcube is in this set. Thus, the probability that a subcube is monochromatic for a large color is at most 2ε.

Second, we handle the small colors. Each small color has at most 2^{n-m+1} vertices. By a generalization of the edge-isoperimetric inequality, the set of vertices of size 2^{n-m+1} with the most monochromatic subcubes of dimension ℓ corresponds to a subcube of dimension $n-m+1$ [BR90, BL90]. This larger subcube contains $\binom{n-m+1}{\ell} 2^{n-m+1-\ell}$ subcubes of dimension ℓ. Since there are at most 2^m small colors, the total number of monochromatic subcubes with small colors is at most $2^{n+1-\ell}\binom{n-m+1}{\ell}$. Since there are $2^{n-\ell}\binom{n}{\ell}$ subcubes total, the probability of a randomly chosen subcube being monochromatic for a small color is at most $2\frac{\binom{n-m+1}{l}}{\binom{n}{l}}$.

Thus, the probability of a randomly chosen subcube being not monochromatic is at least $1 - 2\frac{\binom{n-m+1}{l}}{\binom{n}{l}} - 2\varepsilon$, which means that the average influence is at least this much. □

Note that due to the tightness of the isoperimetric bounds, this bound is essentially the best that can be achieved using an averaging argument. Using Lemma 5.3.2

together with Lemma 5.3.1, we're able to prove the following theorem. Note that the theorem says that if $m > n/\ell$, then we can't even extract with error a small constant.

Theorem 5.3.3. *No function $f : \{0,1\}^n \to \{0,1\}^m$ is an ε-extractor for non-oblivious bit-fixing sources on $\{0,1\}^n$ with $n-\ell$ random bits for any $\varepsilon \leq \frac{1}{10}\min\{\frac{\ell\cdot(m-1)}{n}, 1\}$.*

Proof. Suppose f is an ε-extractor. First note that f must be within ε of uniform on uniform input. So by Lemma 5.3.2, there is a set of variables S of size ℓ with

$$\begin{aligned} I_f(S) &\geq 1 - 2\frac{\binom{n-m+1}{l}}{\binom{n}{l}} - 2\varepsilon \\ &\geq 1 - 2\left(1 - \frac{m-1}{n}\right)^{\ell} - 2\varepsilon \\ &\geq 1 - e^{-\ell\cdot(m-1)/n} - 2\varepsilon. \end{aligned}$$

By Lemma 5.3.1, there is a non-oblivious bit-fixing source X on $\{0,1\}^n$ with $n-\ell$ random bits so that $f(X)$ is distance at least $I_f(S)/4$ from uniform, so $\varepsilon > I_f(S)/4$. Thus

$$\varepsilon > (1 - e^{-\ell\cdot(m-1)/n})/6.$$

If $\ell\cdot(m-1)/n \geq 1$, then

$$\varepsilon > (1 - e^{-1})/6 > 1/10$$

If $\ell\cdot(m-1)/n < 1$, then

$$e^{-\ell\cdot(m-1)/n} < 1 - (1 - e^{-1})\frac{\ell\cdot(m-1)}{n},$$

so

$$\varepsilon > \frac{(1-e^{-1})}{6}\frac{\ell\cdot(m-1)}{n} > \frac{1}{10}\frac{\ell\cdot(m-1)}{n}.$$

□

5.4 Open Questions

It is an open question to close the gap between our lower and upper bounds on extracting from non-oblivious bit-fixing sources. In the boolean case, Kahn, Kalai, and Linial [KKL88] are able to improve upon the edge-isoperimetric bound by a factor of $\log n$ using a harmonic analysis argument. However, their technique does not seem to easily generalize to the non-boolean case. In the other direction, if we could make the construction of Ajtai and Linial [AL93] constructive, we could extract $\Omega(\varepsilon n/(\ell \log^2 n))$ bits, which is within a polylogarithmic factor of optimal.

Chapter 6

Small-Space Sources and Total-Entropy Independent Sources

6.1 Overview of Our Constructions

6.1.1 Small-Space Sources

Recall our model of space s sources as being generated by width 2^s branching programs. We previously saw in Theorem 3.2.4 that nonconstructively there exist extractors even when the space s is a constant fraction of the min-entropy k, even when the min-entropy is logarithmically small. In this chapter, we describe efficient deterministic constructions of extractors for small-space sources.

For space s sources with min-entropy $k = \delta n$, we have several constructions, all of which are able to extract almost all of the entropy in the source. These extractors are summarized in Table 6.1. The first extracts whenever $\delta > n^{-\eta}$ for some fixed constant η and extracts almost all of the entropy.

Table 6.1: Small space extractors for sources on $\{0,1\}^n$ that extract 99% of the min-entropy. In this table c and C represent sufficiently small and large constants, respectively.

Reference	Min-entropy Rate	Space	Error
Thm 6.1.1	$\delta \geq n^{-c}$	$c\delta^3 n$	$\exp(-n^c)$
Thm 6.1.3	Any constant δ	cn	$\exp(-\tilde{\Omega}(n))$
Thm 6.1.4	$\delta \geq C/\log n$	$c\delta \log n$	$\exp(-n^{.99})$

Theorem 6.1.1. *Assume we can find primes with length in $[\tau, 2\tau]$ deterministically in time $\mathrm{poly}(\tau)$. Then there is a constant $\eta > 0$ such that for every $n \in \mathbb{N}$, and $\delta > \zeta > n^{-\eta}$, there is a polynomial-time computable ε-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources with min-entropy rate δ, where $s = \Omega(\zeta^3 n)$, $m = (\delta - \zeta)n$, and $\varepsilon = 2^{-n^{\Omega(1)}}$.*

Remark 6.1.2. The assumption about finding primes follows from Cramer's conjecture on the density of primes [Cra37], together with the deterministic primality test of [AKS04].

We also have constructions that do not depend on the ability to find large primes. Though the parameters of these constructions are mostly subsumed by the previous construction, they are considerably simpler and achieve somewhat better error. For constant min-entropy rate sources, we have a construction that extracts any constant fraction of the entropy.

Theorem 6.1.3. *For any constants $\delta > \zeta > 0$ and every $n \in \mathbb{N}$, there is a polynomial-time computable ε-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources with min-entropy rate δ, where $s = \Omega(n)$, $m = (\delta - \zeta)n$, and $\varepsilon = 2^{-\Omega(n/\log^3 n)}$.*

The last extractor works with min-entropy rate as low as $\delta = \Omega(1/\log n)$ and

space $O(\delta \log n)$.

Theorem 6.1.4. *For every $n \in \mathbb{N}$ and $\delta > \zeta > 28/\log n$ and $s \leq (\zeta \log n)/28$, there is a polynomial-time computable ε-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources with min-entropy rate δ, where $m = (\delta - \zeta)n$ and $\varepsilon = \exp(-n/(2^{O(s/\zeta)} \cdot \log^5 n))$.*

In comparison to the previous results (e.g. [KM04, KM05]) we have reduced the min-entropy required from $n/2$ to $n^{1-\Omega(1)}$ (in Theorem 6.1.1). However, we are still far from logarithmic min-entropy, which can be achieved nonconstructively. We also have a gap in terms of the space tolerated. Nonconstructively we can get s to be almost $\delta n/2$ (see Theorem 3.2.4) while our results require s to be smaller than $\delta^3 n$.

Our extractors for small-space sources are all obtained via a reduction from total-entropy independent sources. The reduction we use is based on that of Koenig and Maurer [KM04, KM05], who used it to generalize extractors for two independent sources. Recall that total-entropy independent sources consist of a string of r independent smaller sources of length ℓ such that the total min-entropy of all r sources is at least k. Our reduction is accomplished by dividing the source into n/t blocks of length t. Then, conditioned on the state at the beginning of each block, the blocks form a set of independent smaller sources. Thus, we get that the original source is a convex combination of total-entropy independent sources on $(\{0,1\}^t)^{n/t}$. The sources in the convex combination have total-entropy $\Omega(k)$ with high probability. Thus any extractor for total-entropy independent sources is also an extractor for small-space sources. In particular, an optimal extractor for total-entropy independent sources is also an essentially optimal extractor for small-space sources.

In a partial attempt to close the entropy gap for the case of space 1 sources, we also have an extractor that extracts about $\Omega(k^2/n)$ bits from a more restricted model when $k > n^{0.81}$. The extra restriction is that the output bit is required to be the same as the state.

6.1.2 Total-Entropy Independent Sources

Our extractors for total-entropy independent sources are all based on generalizing various techniques from extractors for independent and symbol-fixing sources.

Koenig and Maurer [KM04, KM05] showed how any extractor for two independent sources with certain algebraic properties can be translated into an extractor for many sources where only two of the sources have sufficient entropy. Their result generalizes to extractors for more than two sources. We show that this also yields extractors for total-entropy independent sources. In particular, we apply this to extractors for independent sources that follow from the exponential sum estimates of Bourgain, Glibichuk, and Konyagin [BGK06] (see Bourgain [Bou05]), and thereby obtain extractors for total-entropy independent sources of any constant min-entropy rate. These extractors are quite simple. Each source is viewed as being an element of a finite field, and the output of the extractor is simply obtained by taking the product of these finite field elements, then dividing by the size of the field and taking the most significant bits.

We also show how to use ideas from the work of Rao [Rao06] for extracting from several independent sources, together with recent constructions of randomness efficient condensers [BKS+05, Raz05], to get extractors for total-entropy independent sources that extract from sources of min-entropy $(r\ell)^{1-\Omega(1)}$.

When the smaller sources each have short length ℓ, we use ideas from our

extractors for oblivious bit-fixing sources from Chapter 4 to construct extractors for total-entropy independent sources with min-entropy k. We can extract many bits when $k > 2^{\ell}\sqrt{r\ell}$, and for $k = \Omega(2^{2\ell}\ell)$ we can still extract $\Omega(\log k)$ bits. The base extractor simply takes the sum of the sources modulo p for some $p > 2^{\ell}$, similar to our cycle walk extractor for oblivious bit-fixing sources. Using this extractor we can extract $\Omega(\log k)$ bits. To extract more bits when k is sufficiently large, we divide the source into blocks, apply the base extractor to each block, and then use the result to take a random walk on an expander as we did for oblivious bit-fixing sources previously.

Unlike the first two extractors, the extractors obtained using this technique use the full generality of the total-entropy independent sources. In the first two constructions, using a Markov argument we can essentially first reduce the total-entropy independent sources into sources where some of the input sources have sufficiently high min-entropy while the rest may or may not have any min-entropy. These reductions also cause some entropy to be lost. In this last construction, however, we benefit even from those sources that have very little min-entropy. Thus we are able to take advantage of all of the entropy, which helps us extract from smaller values of k.

We also show how to generalize the construction of Gabizon et al. [GRS04] to total-entropy independent sources to enable us to extract more of the entropy. Note that we use it to improve not only the extractors based on our extractors for oblivious bit-fixing sources (analogous to what was done in [GRS04] for bit-fixing sources), but also our extractors based on techniques developed for independent sources. The important point is that their construction only depends on the fact that we have a string of symbols which are independent, and we make no use of the facts that the symbols are bits and are either fixed or random. Thus their proof generalize

Table 6.2: Total-entropy independent source extractors for sources on $(\{0,1\}^\ell)^r$ that extract 99% of the min-entropy. In this table c and C represent sufficiently small and large constants, respectively.

Reference	Min-entropy Rate	Error
Thm 6.1.5	$\delta = (r\ell)^{-c}$	$\exp(-(r\ell)^c)$
Thm 6.1.6	Any constant δ	$\exp(-\tilde{\Omega}(r\ell))$
Thm 6.1.7	$\delta = C\frac{d\log^{3/2} r}{(r\ell)^{\frac{1}{2}-\gamma}}$	$\exp(-(r\ell)^{2\gamma})$
Thm 6.1.8	$\delta = (2^\ell \log r)^C / r$	$(\delta r\ell)^{-c}$

easily to our case.

Independently of our work, Shaltiel [Sha06] has recently generalized the ideas in [GRS04] to give a framework for constructing deterministic extractors which extract almost all of the entropy from extractors which extract fewer bits. Our extractor can be seen to fit inside this framework, although we cannot directly use his results as a black box to obtain our results.

Applying the technique based on [GRS04] to our extractors that use the independent sources techniques of Rao [Rao06], the results of [BGK06], and the bit-fixing source extractor from Chapter 4, respectively, we get the following three theorems. These theorems are directly used to obtain the small-space extractors from Theorem 6.1.1, Theorem 6.1.3, and Theorem 6.1.4. Table 6.2 presents a summary of these extractors.

Theorem 6.1.5. *Assuming we can find primes with length in* $[\tau, 2\tau]$ *deterministically in time* $\mathrm{poly}(\tau)$*, there is a constant* η *such that for every* $r, \ell \in \mathbb{N}$ *and* $\delta > \zeta > (r\ell)^{-\eta}$*, there is a polynomial-time computable* ε*-extractor* $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ *for independent sources on* $(\{0,1\}^\ell)^r$ *with total-rate* $\delta > \zeta$ *where* $m = (\delta - \zeta)r\ell$ *and*

$\varepsilon = \exp(-(r\ell)^{\Omega(1)})$.

We note that in the independent sources model this extractor gives comparable results to the extractor from [BIW04] as a corollary.

The following extractor extracts a constant fraction of the entropy from any constant rate source.

Theorem 6.1.6. *For any constants $\delta > \zeta > 0$ and every $r \in \mathbb{N}$, there is a polynomial-time computable ε-extractor $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for total-rate δ independent sources on $(\{0,1\}^\ell)^r$, where $m = (\delta - \zeta)r\ell$ and $\varepsilon = 2^{-\Omega((r\ell)/\log^3(r\ell))}$.*

For the following extractor we can take $\zeta = \tilde{O}(1/\sqrt{r})$ and can then extract randomness from sources with min-entropy rate as small as $\delta = \tilde{O}(1/\sqrt{r})$.

Theorem 6.1.7. *For every $r \in \mathbb{N}$, $1 \leq \ell \leq \frac{1}{2}\log r$ and $\zeta > \sqrt{2^{2\ell}(\log^3 r)/r\ell}$ there is a polynomial-time computable ε-extractor $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for total-rate $\delta > \zeta$ independent sources on $(\{0,1\}^\ell)^r$ where $m = (\delta - \zeta)r\ell$ and $\varepsilon = \exp(-\Omega((\zeta^2 r\ell)/(2^{2\ell}\log^3 r)))$.*

Gabizon et al. also give a technique which improves the output length of extractors that extract only $\Omega(\log k)$ bits. We show that this technique also generalizes to total-entropy independent sources, so we use it together with our extractor that extracts $\Omega(\log k)$ bits that is based on ideas from our bit-fixing source extractors to get the following theorem. This theorem shows that even for polylogarithmic k, for small enough ℓ we can extract almost all of the min-entropy.

Theorem 6.1.8. *There exists a constant $C > 0$ such that for every $r \in \mathbb{N}$, $\ell \geq 1$, $k \geq (2^\ell \log r)^C$, there exists a polynomial-time computable ε-extractor $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k, where $m = k - k^{1-\Omega(1)}$ and $\varepsilon = k^{-\Omega(1)}$.*

We showed in Theorem 3.3.4 that there exist (nonconstructive) extractors that extract even when the min-entropy k is as small as $O(\ell + \log r)$. Note that we always need $k > \ell$, since otherwise all of the entropy could be in a single source, and thus extraction would be impossible. The extractor from Theorem 6.1.8 comes closest to meeting this bound on k, but only works for small ℓ and has suboptimal error, so there is still much room for improvement.

6.2 Organization

In Section 6.3 we describe our reduction from small-space sources to total-entropy independent sources. We then restrict our focus to extracting from total-entropy independent sources, starting with the basic extractors. In Section 6.4 we describe the extractor that provides the basis for the extractor from Theorem 6.1.6. In Section 6.5 we describe the extractor that provides the basis for the extractor from Theorem 6.1.5. In Section 6.6 we describe the extractors that provide the basis for the extractors from Theorem 6.1.7 and Theorem 6.1.8. Then in Section 6.7, we describe how to generalize the techniques of Gabizon et al. [GRS04] so that we can extract almost all of the entropy, and so achieve the theorems described in the introduction. Finally, in Section 6.8, we give the improved extractor for our more restrictive model of space 1 sources.

6.3 Small-Space Sources As Convex Combinations Of Independent Sources

Here we show how small-space sources can be converted into convex combinations of independent sources. Thus we will be able to use our extractor constructions from subsequent sections to extract from small-space sources. The idea is simple: to extract from a space s source X, we divide the n bits in X into n/t blocks of size t. We view each block as a source on t bits. If we condition on the states of the source at the start of each block, all of the blocks become independent, so we end up with a set of n/t independent smaller sources on $\{0,1\}^t$. We show, using techniques similar to Koenig and Maurer [KM04, KM05], that with high probability these sources will have sufficient min-entropy.

Lemma 6.3.1. *Let X be a space s source on $\{0,1\}^n$ with min-entropy rate δ. Then for any $0 < \alpha < 1$, X is $2^{-\alpha\delta n/2}$-close to a convex combination of independent sources on $(\{0,1\}^\ell)^r$ with total-rate δ', where $\ell = 2s/(\alpha\delta)$, $r = \alpha\delta n/2s$ and $\delta' = (1-\alpha)\delta$.*

All of our extractors for small-space sources are obtained by combining Lemma 6.3.1 with the corresponding extractor for total-entropy independent sources. We note that the reduction in this lemma is only interesting when the min-entropy rate $\delta > 1/\sqrt{n}$, since otherwise the total entropy of the independent sources would be less than the length of an individual source. In this case all of the entropy could be in a single source and thus extraction would be impossible.

To prove Lemma 6.3.1 we use the following standard lemma (for a direct proof see Lemma 5 in Maurer and Wolf [MW97], although it has been used implicitly earlier in, e.g., [WZ99]).

Lemma 6.3.2. *Let X and Y be random variables and let $\mathcal{Y}$ denote the range of Y. Then for all $\varepsilon > 0$*

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

Proof. (Of Lemma 6.3.1.) Divide X into $\alpha\delta n/2s$ blocks of size $2s/\alpha\delta$. Let Y represent the values of the initial states for each block. Then each $(X|Y=y)$ is a set of independent smaller sources with each block viewed as a smaller source of length $2s/(\alpha\delta)$. By Lemma 6.3.2, since $|\mathcal{Y}| = (2^s)^{(\alpha\delta n)/(2s)} = 2^{\alpha\delta n/2}$, with probability $1-2^{-\alpha\delta n/2}$ the sources $(X|Y=y)$ have min-entropy $(1-\alpha)\delta n$ and thus min-entropy rate $(1-\alpha)\delta$. □

6.4 Extracting From Total-Entropy Independent Sources By Reducing To Standard Independent Sources

In this section, we show how to construct extractors for total-entropy independent sources using techniques from standard independent sources.

The following Markov-like lemma will be used to show that if we divide a source into blocks, many of the blocks will have a large entropy rate.

Lemma 6.4.1. *For any partition of a total-rate δ independent source on $(\{0,1\}^\ell)^r$ into t blocks of r/t smaller sources each, the number b of blocks with min-entropy rate greater than $\delta/2$ satisfies $b > \delta t/2$.*

Therefore we can view this source as a set of t independent smaller sources on $\{0,1\}^{\ell r/t}$ where at least $\delta t/2$ of the smaller sources have min-entropy rate greater than $\delta/2$.

Proof. We know that b blocks have min-entropy rate greater than $\delta/2$ and at most 1. In each of the remaining blocks the min-entropy rate is at most $\delta/2$. Since the total entropy rate is δ and min-entropies add for independent sources, $\delta \leq (b + (t - b)(\delta/2))/t$, which after a simple calculation gives the desired result. □

Once we are in this model, we can generalize the result from Koenig and Maurer [KM04, KM05] that states that any two source extractor of the form $f(x_1 \cdot x_2)$, where the x_i are elements of some group, can be extended to any number of sources where only two of the sources have sufficient min-entropy.

Lemma 6.4.2. *Let $(\mathcal{G}, *)$ be a group and let $Ext(x_1, x_2, \ldots, x_b) := f(x_1 * x_2 \cdots * x_b)$ be an extractor for b independent sources over $\mathcal{G}$, each of which has min-entropy rate at least δ. Then $F(x_1, \ldots, x_r) := f(x_1 * \cdots * x_r)$ is an extractor for r independent sources over $\mathcal{G}$, b of which have min-entropy rate at least δ.*

The proof is simple and is the same as in [KM04, KM05]. The key idea is that the r sources can be divided into b blocks, each of which contains exactly one of the high entropy sources, since the group operation cannot lower the entropy.

Bourgain, Glibichuk, and Konyagin [BGK06] gave bounds on the exponential sums of the function $f(x_1, \ldots, x_b) = \prod_{i=1}^{b} x_i$ over large subsets of fields without large subfields, in particular $GF(p)$ and $GF(2^p)$. As observed by Bourgain in [Bou05], this estimate gives an extractor for b independent sources where each source has high entropy. Bourgain only explicitly gives an extractor that outputs a single bit, but his result can be easily generalized using his techniques together with Vazirani's XOR lemma [Vaz86] to get the following theorem.

Theorem 6.4.3. *[BGK06] Let the finite field K be either $GF(p)$ or $GF(2^p)$ for some prime p. Let $f(x_1, \ldots, x_b) = \prod_{i=1}^{b} x_i$ and view the output of the function as an integer*

from 0 *to* $|K|-1$. *Then there exist functions* $B(\delta)$ *and* $c(\delta)$ *such that the function* $\mathsf{BGK}(x_1,\ldots,x_b) = \lfloor (2^m f(x_1,\ldots,x_b))/|K| \rfloor$ *(i.e. taking the* m *most significant bits of* $f(x_1,\ldots,x_b)/|K|$*) is an* ε*-extractor for* b *independent min-entropy rate* δ *sources over* K *for* $b \geq B(\delta)$, $m = \Theta(c(\delta)\log|K|)$, *and* $\varepsilon = 2^{-\Omega(m)}$.

Note that for constant δ, we can extract $\Theta(\log|K|)$ bits with only a constant number of sources. For $GF(p)$, [BGK06] make explicit the relationship between δ and the number of sources and entropy. It turns out in this case that we can handle slightly subconstant δ, down to $\delta = \Omega(1/(\log\log|K|)^{(1/C)})$ for some constant C. For $GF(2^p)$, it's not clear whether or not a similar result can be achieved.

Combining this theorem with Lemma 6.4.2, restricting the sources to be over the multiplicative group K^*, we get the following corollary.

Corollary 6.4.4. *Let the finite field* K *be either* $GF(p)$ *or* $GF(2^p)$ *for some prime* p. *Let* $f(x_1,\ldots,x_r) = \prod_{i=1}^r x_i$ *and view the output of the function as a number from* 0 *to* $|K|-1$. *Then there exist functions* $B(\delta)$ *and* $c(\delta)$ *such that the function* $\mathsf{BGK}(x_1,\ldots,x_r) = \lfloor (2^m f(x_1,\ldots,x_r))/|K| \rfloor$ *is an* ε*-extractor for* r *independent sources over* K^*, *at least* $B(\delta)$ *of which have min-entropy rate at least* δ, *and with* $m = \Theta(c(\delta)\log|K|)$ *and* $\varepsilon = 2^{-\Omega(m)}$.

It will also be useful to formulate the following corollary.

Corollary 6.4.5. *For every constant* $\delta > 0$, *there exists a constant* $v(\delta)$ *and a polynomial time computable function* $\mathsf{BGK} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ *that is an* ε*-extractor for* r *independent sources on* $\{0,1\}^\ell$, *such that at least* $v(\delta)$ *of the sources have min-entropy rate* δ *where* $m = \Omega(\ell)$ *and* $\varepsilon = 2^{-\Omega(\ell)}$.

Proof. Find the next smallest prime $p > \ell$ (we know $p \leq 2\ell$), and apply the extractor from Corollary 6.4.4 over $GF(2^p)$, viewing each source as being embedded in $GF(2^p)^*$. □

Now we can combine this extractor with Lemma 6.4.1 to get an extractor for independent sources with constant total min-entropy rate.

Theorem 6.4.6. *For any constant δ, we can construct a polynomial-time computable ε-extractor $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for total-rate δ independent sources on $(\{0,1\}^\ell)^r$, with $m = \Theta(r\ell)$ and $\varepsilon = 2^{-\Omega(m)}$. This extractor can be computed in time $\mathrm{poly}(r,\ell)$.*

Proof. Given an independent source $X = X_1,\ldots,X_n$ on $(\{0,1\}^\ell)^r$, divide it into $t = 2B(\delta/2)/\delta$ blocks of r/t smaller sources each, where $B(\delta)$ is the constant from Corollary 6.4.4. Then by Lemma 6.4.1, we can view X as an independent sources on $(\{0,1\}^{\ell r/t})^t$, where at least $\delta t/2 = B(\delta/2)$ of the smaller sources have min-entropy rate at least $\delta/2$. Find the smallest prime $p > (r\ell)/t$. By Bertrand's postulate, $p \leq 2(r\ell)/t$, we can find such a prime in time $\mathrm{poly}(r,\ell)$ by brute force search. Then we can embed each of our smaller sources into $GF(2^p)^*$ and apply the extractor from Corollary 6.4.4 to get the stated result. □

6.5 Extracting From Polynomial Entropy Rate

In this section we will show how to extract from total-entropy independent sources when the min-entropy of the sources is polynomially small. As in the previous section, we will reduce the problem to another model: we will try to extract from a few independent sources when just some of them have a polynomial amount of entropy, but we don't know exactly which ones. The probabilistic method shows that extractors exist for this model even when just two sources contain logarithmic min-entropy and the total number of sources is polynomially large. Our main theorem is as follows.

Theorem 6.5.1. *Assuming we can find primes with length in* $[\tau, 2\tau]$ *in time* $\mathrm{poly}(\tau)$*, there is a constant* β *such that there exists a polynomial-time computable* ε*-extractor* $\mathsf{Ext} : (\{0,1\}^{\ell})^r \to \{0,1\}^m$ *for total-rate* $\delta \geq \ell^{-\beta}$ *independent sources on* $(\{0,1\}^{\ell})^r$*, where* $n = \Theta(1/\delta^2)$*,* $m = \ell^{\Omega(1)}$ *and* $\varepsilon = 2^{-\ell^{\Omega(1)}}$*.*

We can also get the following corollary for when we have a larger number of smaller sources.

Corollary 6.5.2. *Assuming we can find primes with length in* $[\tau, 2\tau]$ *in time* $\mathrm{poly}(\tau)$*, there exists a constant* η *such that for any* $\delta \geq (r\ell)^{-\eta}$*, there exists a polynomial-time computable* ε*-extractor* $\mathsf{Ext} : (\{0,1\}^{\ell})^r \to \{0,1\}^m$ *for total-rate* δ *independent sources on* $(\{0,1\}^{\ell})^r$*, where* $m = (\delta^2 r\ell)^{\Omega(1)}$ *and* $\varepsilon = 2^{-(\delta^2 r\ell)^{\Omega(1)}}$*.*

Proof. Divide the source into $\Theta(1/\delta^2)$ blocks of $\Theta(\delta^2 n)$ smaller sources each and apply Theorem 6.5.1. □

In this section we will describe a generic technique to turn any extractor for the model where a few smaller sources have min-entropy rate less than half into an extractor that can extract when the min-entropy is as small as $\ell^{1-\alpha_0}$ for some universal constant α_0. There are two major ingredients that will go into our construction:

- The first ingredient is based on recent constructions of randomness efficient condensers [BKS+05, Raz05]. We use these condensers to transform a set of sources with polynomial min-entropy rate into a set of aligned sources with somewhere-min-entropy rate 0.9. An important property that we will need is that the length of each of the rows is much higher than the number of rows. We prove the following theorem in Section 6.5.2.

Theorem 6.5.3. *Assume we can find primes with length in* $[\tau, 2\tau]$ *in time* $\mathrm{poly}(\tau)$. *Let* $X_1, \ldots, X_B$ *all be sources on* $\{0,1\}^\ell$, *for* B *a constant. Then for any small enough constant* $\alpha > 0$ *there exist constants* $\gamma = \gamma(\alpha)$ *and* $\mu(\alpha) > 2\gamma$ *and a polynomial time computable function* $\mathsf{ACond} : \{0,1\}^\ell \to (\{0,1\}^{\ell^\mu})^{\ell^\gamma}$ *such that if each* X_i *has min-entropy rate* $\delta = \ell^{-\alpha}$, *then*

$$\mathsf{ACond}(X_1), \mathsf{ACond}(X_2), \ldots, \mathsf{ACond}(X_B)$$

is $2^{-\Omega(\ell^{1-2\alpha})}$ *close to a convex combination of sets of aligned somewhere-min-entropy rate* 0.9 *sources.*

- The second ingredient is the technique of condensing independent SR-sources from the work of Rao [Rao06]. We will generalize a theorem from that work. We show how to extract from independent sources with only a few of them being aligned SR-sources that have rows that are much longer than the number of rows. Formally, we get the following, proved in Section 6.5.3:

  Theorem 6.5.4. *For every constant* $\gamma < 1$ *there exists a polynomial time* $2^{-\ell^{\Omega(1)}}$*-extractor* $\mathsf{SRExt} : (\{0,1\}^{\ell^{\gamma+1}})^u \to \{0,1\}^m$ *for* u *independent sources, of which* v *are independent aligned* $(\ell^\gamma \times \ell)$ *SR-sources, where* $m = \ell - \ell^{\Omega(1)}$.

We will first describe how to use these two ingredients to extract from an intermediate model. Then we will see that total-entropy independent sources can be easily reduced to this intermediate model to prove Theorem 6.5.1.

6.5.1 Extracting From The Intermediate Model

The intermediate model we work with is defined as follows.

Definition 6.5.5. A (u, v, α) intermediate source X consists of u^2 smaller sources $X^1, \ldots, X^{u^2}$, each on $\{0,1\}^\ell$. These smaller sources are partitioned into u sets $S_1, \ldots, S_u$ such that v of the sets have the property that v of the smaller sources in the set have min-entropy at least $\ell^{1-\alpha}$.

Now we show that for certain constant v and $\alpha > 0$ we can extract from the class of sources in this model.

Theorem 6.5.6. *Assuming we can find primes with length in* $[\tau, 2\tau]$ *in time* $\mathrm{poly}(\tau)$*, for some constants* v *and* $\alpha > 0$ *there exists a polynomial time computable* $2^{-\ell^{\Omega(1)}}$*-extractor* IExt *for* (u, v, α) *intermediate sources, where* $m = \ell^{\Omega(1)}$.

Using this theorem together with Lemma 6.4.1, we can prove Theorem 6.5.1.

Proof. (Of Theorem 6.5.1.) Let $X = X_1, \ldots, X_r$ be an independent source on $(\{0,1\}^\ell)^r$ with total min-entropy rate $\delta \geq 4\ell^{-\alpha}$, where α is the constant from Theorem 6.5.6 and $n = u^2$ where u will be chosen later. Divide the source into u blocks with u smaller sources each. By Lemma 6.4.1, $\delta u/2$ of the blocks have min-entropy rate at least $\delta/2$. Now further divide each of the blocks into u sub-blocks of one smaller source each. By Lemma 6.4.1, for the blocks with min-entropy rate at least $\delta/2$, at least $\delta u/4$ of the sub-blocks have min-entropy rate $\delta/4 \geq \ell^{-\alpha}$. Let $u = 4v/\delta$, where v is the constant from Theorem 6.5.6. Then X is a (u, v, α) intermediate source satisfying the conditions of Theorem 6.5.6, which immediately gives us the theorem. □

Here is the algorithm promised by Theorem 6.5.6:

Construction: $\mathsf{IExt}(x^1, \ldots, x^{u^2})$

Input: $x^1, \ldots, x^{u^2}$ partitioned into sets $S_1, \ldots, S_u$

Output: z.

Let v be a constant that we will pick later.

Let BGK be as in Corollary 6.4.5 - an extractor for independent sources when $v-1$ of the smaller sources have min-entropy.

Let ACond be as in Theorem 6.5.3, letting $B = v^2$ - a condenser that converts sources with sublinear min-entropy into a convex combination of aligned sources with somewhere-min-entropy rate 0.9.

Let SRExt be as in Theorem 6.5.4 - an extractor for independent sources that works when just v of the inputs come from aligned SR-sources.

Set $\varepsilon = 1/v^3$. Let α be a small enough constant to apply Theorem 6.5.3 with α in the hypothesis. Let γ be as in the conclusion of the theorem.

1. Compute $y^i = \mathsf{ACond}(x^i)$ for every source in the input. Let y^i_j denote the jth row of y^i.

2. For every $l \in [u]$, and every $j \in [2^{\ell^\gamma}]$, let b^l_j be the string obtained by applying BGK using the y^i_j from all $i \in S_l$ as input.

   We think of b^l as a sample from a SR-source with ℓ^γ rows, one for each seed s_i.

3. Output $\mathsf{SRExt}(b^1, \ldots, b^u)$.

Proof. (Of Theorem 6.5.6)

If we restrict our attention to the v^2 high min-entropy smaller sources, from Theorem 6.5.3 we know that from the first step from these smaller sources is $2^{-\Omega(\ell^{1-2\alpha})}$ close to a convex combination of sets of aligned somewhere-min-entropy rate 0.9 smaller sources.

Then in the second step, for each distribution in the convex combination BGK succeeds in extracting from the aligned min-entropy rate 0.9 row in each set.

Remark 6.5.7. Actually, we don't really need the Bourgain-Glibichuk-Konyagin extractor for this step. If the min-entropy is so high, it is easy to see that the generalized inner product function is an extractor. Still we use BGK since we will need it later on in the construction.

Thus the result of the first step in the algorithm is a distribution that is $2^{-\ell^{\Omega(1)}}$-close to a convex combination of collections of u independent smaller sources, v of which are independent aligned SR-sources.

Our extractor SRExt then extracts from each distribution in the convex combination, and thus extracts from the entire convex combination. So our algorithm succeeds in extracting from the input. □

6.5.2 Condensing To Aligned Sources With High Somewhere-Min-Entropy

In this section we give the condenser from Theorem 6.5.3. The first ingredient we'll need is the following condenser from [Zuc06], which improves on the condenser from [BKS+05].

Lemma 6.5.8. *[Zuc06] Assuming we can find primes with length in* $[\tau, 2\tau]$ *in time* $\mathrm{poly}(\tau)$*, there exists a constant* $\alpha > 0$ *such that for any* $t, \ell > 0$ *there exists a polynomial-time computable condenser* $\mathsf{Zuck} : \{0,1\}^{\ell} \to (\{0,1\}^{(2/3)^t \ell})^{2^t}$ *such that if* X *has rate* δ*,* $\mathsf{Zuck}(X)$ *is* $t2^{-\Omega(\alpha\delta\ell)}$ *close to somewhere-min-entropy rate* $\min((1+\alpha)^t\delta, 0.9)$.

We'll also need to use the condenser from Raz's work [Raz05] with the improved analysis of Dvir and Raz (Lemma 3.2 in [DR05]), which shows that most of the output rows are statistically close to having high min-entropy.

Lemma 6.5.9. *[DR05] For any constant $c > 0$, there is a polynomial-time computable function $\mathsf{Raz} : (\{0,1\}^{\ell})^{r} \to (\{0,1\}^{\Theta(\ell)})^{2^{\Theta(r)}}$ such that the following holds. If the input source X has somewhere-min-entropy rate δ, the output $\mathsf{Raz}(X)$ is $2^{-\Omega(\delta\ell)}$ close to a convex combination of distributions, each of which has the property that at least a $(1-c)$ fraction of its rows have min-entropy rate at least 0.9δ.*

Now we can apply the functions from the previous two lemmas in succession to transform any source with min-entropy rate δ into a convex combination of sources with high somewhere-min-entropy sources where almost all of the rows in the sources have high min-entropy.

Lemma 6.5.10. *Assuming we can find primes with length in $[\tau, 2\tau]$ in time $\mathrm{poly}(\tau)$, there exists a constant $\alpha > 0$ such that for any constants $t > 0$ and $c > 0$ there exists a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^{\ell} \to (\{0,1\}^{\Theta((2/3)^t \ell)})^{2^{\Theta(2^t)}}$ such that the following holds. If the input source X has min-entropy rate δ, the output $\mathsf{Cond}(X)$ is $2^{-\Omega(\delta\ell)}$ close to a convex combination of distributions, each of which has the property that at least a $(1-c)$ fraction of its rows have min-entropy rate at least $\min(0.9\delta(1+\alpha)^t, 0.9)$.*

Proof. Let $\mathsf{Cond}(x) = \mathsf{Raz}(\mathsf{Zuck}(x))$. □

Corollary 6.5.11. *Assuming we can find primes with length in $[\tau, 2\tau]$ in time $\mathrm{poly}(\tau)$, there is a constant C such that for any constant $c > 0$ there exists a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^{\ell} \to (\{0,1\}^{\Theta(\ell)})^{C}$ such that the following holds. If the input source X has min-entropy rate δ, then the output $\mathsf{Cond}(X)$ is $2^{-\Omega(\delta\ell)}$ close*

to a convex combination of distributions where each source in the convex combination has the property that at least a $(1-c)$ *fraction of its rows have min-entropy rate at least* $\min(2\delta, 0.9)$.

Proof. Pick t large enough (but still constant) in Lemma 6.5.10 so that $0.9(1+\alpha)^t \geq 2$. Then $C = 2^{\Theta(2^t)}$. □

Now we can use this basic condenser to help prove Theorem 6.5.3. To do this, we apply this condenser to our input smaller sources and then recursively apply it to the outputs. We might think we could just apply the union bound to show that most of the output rows are aligned, but that is not true. However, we only need that one single row in the output is aligned, which we can accomplish by ensuring that at each step we have an aligned row, and then concentrating recursively on that aligned row.

Proof. (Of Theorem 6.5.3.) First, apply the function Cond from Corollary 6.5.11 to each X_i, choosing $c < \frac{1}{B}$. Then the output $\langle \mathsf{Cond}(X_1), \mathsf{Cond}(X_2), \ldots, \mathsf{Cond}(X_B) \rangle$ is $2^{-\Omega(\delta\ell)}$ close to a convex combination of distributions $Y = \sum_j \beta_j Y^{(j)}$, where $Y^{(j)} = \langle Y_1^{(j)}, Y_2^{(j)}, \ldots, Y_B^{(j)} \rangle$ and $\sum_j \beta_j = 1$. Each smaller source $Y_i^{(j)}$ has the property that at least a $(1-c)$ fraction of its rows have min-entropy rate at least 2δ. Now we restrict our attention to a single source in the convex combination $Y^{(j)}$. In this source at most $cB < 1$ fraction of the rows have a smaller source $Y_i^{(j)}$ with min-entropy rate less than 2δ in that row. Thus there is at least one row where the min-entropy rate for all the smaller sources is at least 2δ, i.e., the output is aligned with somewhere-min-entropy rate 2δ. Now we recursively apply Cond to each row in each output source. When we apply it to the aligned row, we'll get another aligned row with min-entropy rate 4δ. If we recursively do this t times, we end up close to a convex combination of a set of aligned sources with somewhere-min-entropy

rate $2^t\delta$. If we let $t = \log(0.9/\delta) = \log(0.9\ell^\alpha)$, then these sources have somewhere-min-entropy rate 0.9. If we choose α small enough (depending on the constants in Corollary 6.5.11), then we can achieve $\mu > 2\gamma$, as desired. □

6.5.3 Extracting From Independent Sources, A Few Of Which Are Aligned SR-Sources

Here we will prove Theorem 6.5.4. Our extractor will be obtained by condensing the aligned SR-sources, closely following a similar construction of Rao [Rao06]. The additional complication is that whereas in [Rao06] every source was assumed to have a random row, in our model only some of the sources contain a random row and the rest may be arbitrary. We will build a condenser that when given u independent sources, v of which are aligned SR-sources, outputs a distribution that is statistically close to a convex combination of sources of the same type, with far fewer rows in each SR-source. Our condenser can handle an arbitrarily large u and some small universal constant v.

Iterating our condenser, we will eventually obtain just one row in our SR-sources, at which point we can use BGK from Corollary 6.4.5 to extract from the sources, or even simply XOR all the sources together.

To condense a single source from the input, we will take a small slice of bits from all other sources in the input. We will use these slices to generate a short list of candidate seeds that are independent of the source we are trying to condense. Then we will use these seeds with a strong seeded extractor to extract from the source we are trying to condense. In this way we reduce the number of rows of one source. In particular, the seeded extractor we use is as follows.

Theorem 6.5.12. *[Tre01, RRV02] For every $n, k \in \mathbb{N}$, $\varepsilon > 0$, there is a polynomial-*

time computable strong seeded ε*-extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{k-O(\log^3(n/\varepsilon))}$ *for sources with min-entropy k, with* $t = O(\log^3(n/\varepsilon))$.

To condense all of the sources, we repeat the same construction with all sources: each source is condensed using seeds generated from slices of the other sources. The output of all this condensing is u sources that are no longer independent. Still, we will argue that if we fix all the slices of bits we used to generate the seeds, the output is the distribution of independent sources of the type that we want.

Remark 6.5.13. Although we do not include the details here, it is not hard to modify the construction in this subsection to extract even when $v = 2$ and u is arbitrarily large, by replacing the function BGK from Corollary 6.4.5 in the composition below with a generalization of Bourgain's extractor for two independent sources [Bou05]. We can also show that our construction is *strong*, i.e. the output of our extractor is statistically close to being independent of any one source from the input.

Now we describe our condenser in detail.

Construction: $\mathsf{Cond}(x^1, \ldots, x^u)$

Input: $x^1, \ldots, x^u$, strings each divided into t rows of length r.

Output: $z^1, \ldots, z^u$.

Let w, l be parameters that we will set later.

Let BGK be as in Corollary 6.4.5 - an extractor for independent sources when $v-1$ of them have min-entropy rate 0.2. Let a be the output length of BGK. Let ε_1 be the error of BGK.

Let Ext be the strong seeded extractor promised by Theorem 6.5.12. We will set up Ext to extract from sources on $\{0,1\}^{ta}$ with min-entropy at least $a-l$ and to have output length m, using seed length a. Let ε_2 be the error of Ext.

1. For each source, partition its rows into pairs of rows.

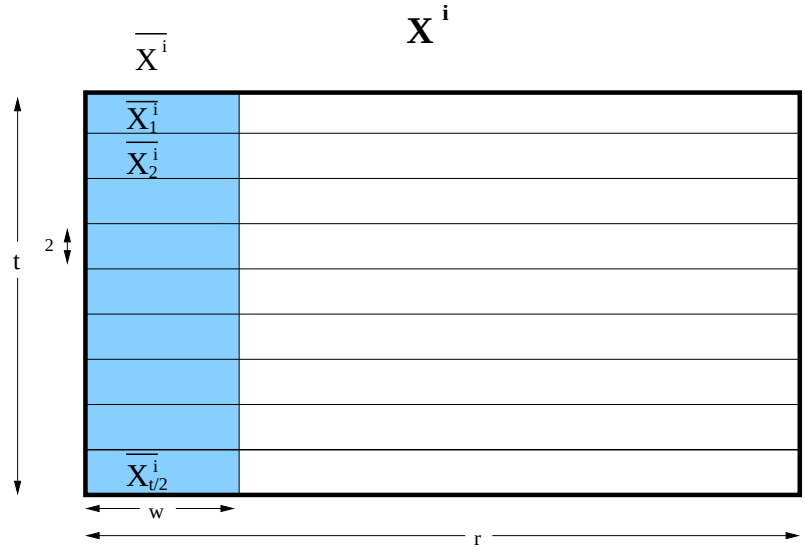

Figure 6.1: Notation in one source

2. For $i = 1,2,\ldots,u$ and $j = 1,2,\ldots,t/2$ let $\overline{x}^i_j$ denote the first w bits of the $j'th$ pair of rows in the string x^i. Let $\overline{x}^i$ denote the first w bits of every row of x^i. Let $\overline{x}^{\neq i}_j$ denote the concatenation of the first w bits of the j'th pair rows of all sources except the i'th source.

3. For every $i = 1,2,\ldots,u$, and $j = 1,2,\ldots,t/2$, let $z^i_j = \mathsf{Ext}(x^i, \mathsf{BGK}(x^{\neq i}_j))$.

4. For every $i = 1,2,\ldots,u$, let z^i be the concatenation of $z^i_1 \circ \cdots \circ z^i_{t/2}$

Lemma 6.5.14. *Let* Cond *be as above. If $X^1, X^2, \ldots, X^u$ are independent sources, with v of them being aligned $(t \times a)$ SR-sources, then $Z^1, Z^2, \ldots, Z^u$ are $v(\varepsilon_1 + 2\sqrt{\varepsilon_2} + 2^{-(l-tw)})$-close to a convex combination of independent sources, v of which are aligned $(t/2 \times m)$ SR-sources.*

Proof. Let h be such that the h'th pair of rows in $X^1, \ldots, X^u$ contains a random row. We will argue that the h'th row of the output distribution is statistically close to uniform.

To see this, consider the random variable $\overline{X} = \overline{X}^1 \circ \cdots \circ \overline{X}^u$, the concatenation of all the slices that are used to generate the various seeds.

We will partition the support of this variable into two sets, a *good* set and a *bad* set. We will then make the following two claims, which clearly imply the lemma.

Claim 6.5.15. *For good $\overline{x}$, $(Z^1 \circ \cdots \circ Z^u)|\overline{X} = \overline{x}$ is the distribution of u independent sources, with v of them being $v\sqrt{\varepsilon_2}$ close to aligned SR-sources.*

Claim 6.5.16. $\Pr[\overline{X}$ *is not good* $] < v\varepsilon_1 + v\sqrt{\varepsilon_2} + v2^{tw-l}$

To ensure these claims, the notion of good we will use is this one: call $\overline{x}$ *good for source* X^i if

1. $X^i|\overline{X} = \overline{x}$ has min-entropy at least $r - l$
2. $\mathsf{BGK}(\overline{x}_h^{\neq i})$ is a good seed to extract from $X^i|\overline{X} = \overline{x}$, i.e.

$$\| \mathsf{Ext}(X^i|\overline{X} = \overline{x}, \mathsf{BGK}(\overline{x}_h^{\neq i})) - U_m \| < \sqrt{\varepsilon_2}$$

We will say that $\overline{x}$ is *good* if it is good for all the v sources that contain a random row. Claim 6.5.15 immediately follows from this notion of good. All we have left to prove is Claim 6.5.16. The proof requires the following simple proposition.

Proposition 6.5.17. *Let X be a random variable with $H_\infty(X) = k$. Let A be any event in the same probability space. Then $H_\infty(X|A) < k' \Rightarrow \Pr[A] < 2^{k'-k}$.*

Proof. (of Claim 6.5.16) Fix an i so that X^i is one of the v aligned SR-sources that contains a random row. We will first argue that $\overline{X}$ is good for X^i with high

probability. Then we will use the union bound to claim that it is good with high probability.

$\overline{X}$ is good for X^i when two events occur:

1. Event T: $X^i|\overline{X}=\bar{x}$ has min-entropy at least $a-l$. This event is equivalent to the event $X^i|\overline{X}^i=\bar{x}^i$ has min-entropy at least $a-l$, since X^i only depends on those bits of $\overline{X}$.

2. Event U: $\mathsf{BGK}(\bar{x}_h^{\neq i})$ is a good seed to extract from $X^i|\overline{X}=\bar{x}$, i.e.

$$\| \mathsf{Ext}(X^i|\overline{X}=\bar{x}, \mathsf{BGK}(\bar{x}_h^{\neq i})) - U_m \| < \sqrt{\varepsilon_2}$$

By Proposition 6.5.17, the probability that event T does not occur is at most $2^{-l}2^{tw}$. This is because there are 2^{tw} possible settings for $\bar{x}^i$. Every bad setting occurs with probability at most 2^{-l}, thus by the union bound, the probability that any bad setting occurs is at most 2^{tw-l}.

Now given that T does occur, event U does not occur with probability at most $\sqrt{\varepsilon_2}+\varepsilon_1$. This is because the output of BGK is ε_1-close to uniform and for a uniformly chosen seed the probability that Ext fails to extract from the source is at most $\sqrt{\varepsilon_2}$ by the strong extractor property and Markov's inequality.

Thus by the union bound, the probability that either T or U do not occur is at most $2^{tw-l}+\sqrt{\varepsilon_2}+\varepsilon_1$.

Applying the union bound again, $\overline{X}$ is good for all the X^i's we care about with probability at least $1-v(2^{tw-l}+\sqrt{\varepsilon_2}+\varepsilon_1)$.

□

This concludes the proof of the lemma.

□

Now we can prove the main theorem of this section.

Proof. (of Theorem 6.5.4)

We will use the condenser Cond repeatedly. In each step we reduce the number of rows in each of the sources by a factor of 2. We need to repeat the condensation step at most $\lceil \gamma \log \ell \rceil$ times. By Lemma 6.5.14 the error in each step is $v(\varepsilon_1 + 2\sqrt{\varepsilon_2} + 2^{-(l-tw)})$.

Recall that ε_1 is the error of BGK from Corollary 6.4.5. Thus $\varepsilon_1 = 2^{-\Omega(w)}$ in every step, since w is the length of the inputs to BGK. ε_2 was the error of Ext from Theorem 6.5.12. Since the seed length $a = \Omega(w)$, the error ε_2 is at most $2^{-w^{\Omega(1)}}$ in every step.

Setting $l = 2\ell^{(1+\gamma)/2}$, $w = l/(2t) = \ell^{\Omega(1)}$, we get a total error of $2^{-\ell^{\Omega(1)}}$.

In each step the length of the sources drops by $\ell^{\beta'}$ for some small constant β'. Thus the final output length is at least $\ell - \ell^{\beta}$ for some constant $\beta \in (0,1)$.

□

6.6 Better Extractors For Total-Entropy Independent Sources With Many Smaller Sources

Now we show how for sources consisting of many smaller sources of length ℓ we can do better than the constructions in the previous sections by generalizing our previous constructions for symbol-fixing sources. The base extractor simply takes the sum of the smaller sources modulo p for some prime $p > 2^{\ell}$. Unlike in the symbol-fixing case, we cannot use an expander random walk directly. To see this, think of the extreme case where each symbol is uniformly distributed over only two values. In this case we run into the same problems as in the oblivious bit-fixing

case, since then we'd be essentially taking a random walk on a degree two graph. To overcome this problem, we use similar techniques to the oblivious bit-fixing source case. We start with taking a random walk on the p-cycle for some prime $p > 2^\ell$, or equivalently, taking the sum modulo p. Using the relation between the distance to the uniform distribution and Fourier coefficients over $\mathbb{Z}_p$, we're able to show that for any prime p this is an extractor. As in the bit-fixing case, we can then extract $\Omega(\log k)$ bits for any min-entropy k. Then we divide the source into blocks, apply the base extractor to each block, and then use the result to take a random walk on an expander as we did in Chapter 4.

Recall the following (slightly rephrased) definition from Section 2.2.

Definition 6.6.1. An independent source on $(\{0,1\}^\ell)^r$ is a (k,ε)-*approximate symbol-fixing source* if k of the r smaller sources have distributions within an ℓ_2 distance ε of uniform.

As before, these sources will be used as intermediate sources. We will transform the sources we wish to extract from into approximate symbol-fixing sources and then use the results from Chapter 4 to extract from these sources. In particular, we will need the following proposition, which follows from Lemma 4.3.8.

Proposition 6.6.2. *Let G be an undirected non-bipartite d-regular graph on 2^m vertices with uniform transition matrix P. Then we can construct a polynomial-time computable ε'-extractor for the set of (k,ε)-approximate oblivious symbol-fixing sources on $[d]^r$, where $\varepsilon' = \frac{1}{2}(\lambda(P) + \varepsilon\sqrt{d})^k 2^{m/2}$. This extractor simply uses the input from the source to take a random walk on G and outputs the label of the final vertex.*

6.6.1 Reducing to Flat Total-Entropy Independent Sources

It will be simpler to analyze our extractor for flat total-entropy independent sources. We show that any extractor that works for flat total-entropy independent sources also works for general total-entropy independent sources because any total-entropy independent source is close to a convex combination of flat independent sources with high total-entropy.

Lemma 6.6.3. *Any ε-extractor for the set of flat independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy $k/(2\log 3)$ is also an $(\varepsilon + e^{-k/9})$-extractor for the set of independent sources on $(\{0,1\}^\ell)^r$ with min-entropy k.*

This lemma follows directly from the following lemma.

Lemma 6.6.4. *Any independent source $X = X_1, \ldots, X_r$ on $(\{0,1\}^\ell)^r$ with total min-entropy k is $e^{-k/9}$-close to a convex combination of flat independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy $k/(2\log 3)$.*

Proof. Let $H_\infty(X_i) = k_i$ for all i. If $k_i \geq 1$, we can write X_i as a convex combination of flat sources with support size $\lfloor 2^{k_i} \rfloor$. Each of these flat sources has min-entropy $\log \lfloor 2^{k_i} \rfloor > \frac{k_i}{\log 3}$, since we lose the largest fraction of min-entropy from taking the floor when 2^{k_i} is nearly 3.

If $k_i < 1$, then we must have constant sources in our convex combination, so if we did as above, we'd lose up to a bit of entropy for each such i. Instead, suppose k' of the total entropy is contained in X_i with less than a bit of entropy each. Call this set $S \subseteq [r]$. Now suppose $k' \leq k/2$. In this case, we can write X_S as a convex combination of constant sources and we are still left with $(k-k')/\log 3 \geq k/(2\log 3)$ bits of entropy in each of our sources, as desired.

From now on we will assume $k' \geq k/2$. We will show we can write X_S as a convex combination of sources that with probability $1-\varepsilon$ have min-entropy $k'/3$.

For each $i \in S$, we can write X_i as a convex combination of flat sources with one or zero bits of entropy. The one bit sources are obtained by choosing uniformly between the most probable value and each of the other values for X_i. Each of these sources occurs with probability equal to twice the probability of the less probable value. Since the most probable value occurs with probability 2^{-k_i}, we get one bit of entropy with probability $2(1-2^{-k_i})$. Otherwise, X_i is fixed to the most probable value.

Now we can use a Chernoff bound to bound the entropy in the sources in the overall convex combination of sources for X_S. Let Y_i be an indicator random variable for the ith source having one bit of entropy. Then $Y = \sum Y_i$ is a random variable representing the total entropy. Note that $\mathbb{E}[Y] = \sum \mathbb{E}[Y_i] = \sum 2(1-2^{-k_i}) \geq \sum k_i = k'$, where the inequality is true because $k_i < 1$. Now we are ready to apply the Chernoff bound (Theorem A.1.13 in Alon and Spencer [AS00]).

$$\Pr[Y < (1-\lambda)k'] \leq \Pr[Y < (1-\lambda)\mathbb{E}[Y]] < e^{-\lambda^2(\sum(1-2^{-k_i}))} \leq e^{-\lambda^2 \frac{k'}{2}} \leq e^{-\lambda^2 \frac{k}{4}}$$

Setting $\lambda = 2/3$ we get the desired error bound $\varepsilon = e^{-\frac{k}{9}}$. Then with probability $1-\varepsilon$ we have at least $(k-k')/\log 3 + k'/3 \geq k/(2\log 3)$ bits of entropy, as desired. □

6.6.2 Extracting From Flat Total-Entropy Independent Sources

Now we show how to extract from flat total-entropy independent sources for small ℓ. Our initial extractor simply takes the sum modulo p of the individual sources, for some prime $p \geq 2^\ell$

Theorem 6.6.5. *Let $\ell \geq 1$ and $p \geq 2^\ell$ a prime. Then $Sum_p : (\{0,1\}^\ell)^r \to [p]$, where $Sum_p(x) = \sum_i x_i \mod p$, is an ε-extractor for the set of flat independent sources on*

$(\{0,1\}^\ell)^r$ *with total min-entropy k, where* $\varepsilon = \frac{1}{2}2^{-2k/p^2}\sqrt{p}$.

Combining Theorem 6.6.5 with Lemma 6.6.3 we get an extractor for total-entropy independent sources.

Corollary 6.6.6. *Suppose* $p \geq 2^\ell$ *is a prime. Then* Sum_p *is an* ε*-extractor for the set of independent sources on* $(\{0,1\}^\ell)^r$ *with total min-entropy* $k \geq \Omega(p^2 \log p)$*, where* $\varepsilon = 2^{-\Omega(k/p^2)}$.

We will prove Theorem 6.6.5 via the following lemma, which will be useful later.

Lemma 6.6.7. *Let* $\ell \geq 1$ *and* $p \geq 2^\ell$ *a prime. Then for all sets of flat independent sources* $X = X_1, \ldots, X_r$ *on* $(\{0,1\}^\ell)^r$ *with min-entropy k,* $Sum_p(x)$ *has* ℓ_2 *distance from uniform at most* $2^{-2k/p^2}$.

It is well known that if X and Y are both distributed over a universe of size p, then $|X - Y| \leq \frac{1}{2}\sqrt{p}||X - Y||$. Theorem 6.6.5 then follows by combining this lemma with this relation between ℓ_2 and variation distance.

To analyze the distance from uniform of the sum modulo p, we use the following lemma that relates this distance to the additive characters of $\mathbb{Z}_p$. For $\mathbb{Z}_p$, where p is a prime, the ith additive character is defined as $\chi_j(a) = e^{\frac{2\pi i j a}{p}}$.

Lemma 6.6.8. *For any function* $f : \{0,1\}^r \to \mathbb{Z}_p$ *and random variable* X *over* $\{0,1\}^r$,

$$||f(X) - U_p||^2 = \frac{1}{p}\sum_{j=1}^{p-1} |\mathbb{E}[\chi_j(f(X))]|^2 < \max_{j \neq 0} |\mathbb{E}[\chi_j(f(X))]|^2,$$

where U_p *denotes the uniform distribution over* $\mathbb{Z}_p$.

Proof. Let $Y = f(X) - U_p$. The jth Fourier coefficient of Y is given by $\hat{Y}_j = \sum_{y=0}^{p-1} Y(y)\chi_j(y)$. By Parseval's Identity and using the fact that $\sum_{y=0}^{p-1} \chi_j(y) = 0$ when $j \neq 0$ we get

$$\begin{aligned}
||Y||^2 = \frac{1}{p}\sum_{j=0}^{p-1} |\hat{Y}_j|^2 &= \frac{1}{p}\sum_{j=0}^{p-1} \left|\sum_{y=0}^{p-1} Y(y)\chi_j(y)\right|^2 \\
&= \frac{1}{p}\sum_{j=0}^{p-1} \left|\sum_{y=0}^{p-1} \Pr[f(X)=y]\chi_j(y) - \frac{1}{p}\sum_{y=0}^{p-1}\chi_j(y)\right|^2 \\
&= \frac{1}{p}\sum_{j=1}^{p-1} |\mathbb{E}[\chi_j(f(X))]|^2 \\
&< \max_{j\neq 0} |\mathbb{E}[\chi_j(f(X))]|^2.
\end{aligned}$$

□

Using the previous lemma we can now prove Theorem 6.6.5.

Proof. Let $f(X) = \sum_{i=1}^{r} X_i$ and fix $j \neq 0$. Then $|\mathbb{E}[\chi_j(f(X))]|^2 = \prod_{i=1}^{r} |\mathbb{E}[\chi_j(X_i)]|^2$. Suppose X_i has min-entropy k_i, so $k = \sum_i k_i$. Then since each X_i is a flat source, X_i is uniformly distributed over $K_i = 2^{k_i}$ values. Our goal is to upper bound $|\mathbb{E}[\chi_j(X_i)]|^2$

over all possible choices of X_i. Doing so, we get

$$\begin{aligned}
\left|\mathbb{E}[\chi_j(X_i)]\right|^2 &\leq \max_{X_i:\mathbb{Z}_p\to\{0,1/K_i\},\sum_x X_i(x)=1} \left|\mathbb{E}[\chi_j(X_i)]\right|^2 \\
&= \max_{X_i:\mathbb{Z}_p\to\{0,1/K_i\},\sum_x X_i(x)=1} \left|\sum_{x\in\mathbb{Z}_p} X_i(x)\chi_j(x)\right|^2 \\
&= \max_{y,|y|=1} \left(\max_{X_i:\mathbb{Z}_p\to\{0,1/K_i\},\sum_x X_i(x)=1} \left(\left(\sum_{x\in\mathbb{Z}_p} X_i(x)\chi_j(x)\right)\odot y\right)^2\right) \\
&= \max_{X_i:\mathbb{Z}_p\to\{0,1/K_i\},\sum_x X_i(x)=1} \left(\max_{y,|y|=1} \left(\sum_{x\in\mathbb{Z}_p} X_i(x)(\chi_j(x)\odot y)\right)^2\right),
\end{aligned}$$

where $\odot$ denotes the complex dot product, where the complex numbers are viewed as two dimensional vectors, and the third line follows from the observation that the dot product is maximized when y is in the same direction as $(\sum_{x\in\mathbb{Z}_p} X_i(x)\chi_j(x))$, in which case we get exactly the square of the length. Now we further note that $\chi_j(x)\odot y$ is greatest for values of x for which $\chi_j(x)$ is closest to y. Thus we achieve the maximum when X_i is distributed over the K_i values closest to y. Without loss of generality we can assume these values correspond to $x = 0$ to $K_i - 1$ (since we only

care about the magnitude). Thus

$$\begin{aligned}
|\mathbb{E}[\chi_j(X_i)]|^2 &\leq \left| \frac{1}{K_i} \sum_{j=0}^{K_i-1} e^{\frac{2\pi i j}{p}} \right|^2 \\
&= \left| \frac{1}{K_i} \frac{1-e^{\frac{2\pi i j K_i}{p}}}{1-e^{\frac{2\pi i}{p}}} \right|^2 \\
&= \left| \frac{1}{K_i} \frac{e^{\frac{\pi i K_i}{p}}(e^{\frac{-\pi i K_i}{p}} + e^{\frac{\pi i K_i}{p}})}{e^{\frac{\pi i}{p}}(e^{\frac{-\pi i}{p}} + e^{\frac{\pi i}{p}})} \right|^2 \\
&= \left(\frac{1}{K_i} \frac{\sin(\frac{\pi K_i}{p})}{\sin(\frac{\pi}{p})} \right)^2 \\
&= \left(\frac{1}{K_i} \frac{\frac{\pi K_i}{p} \prod_{m=1}^{\infty}(1-\frac{K_i^2}{p^2m^2})}{\frac{\pi}{p} \prod_{m=1}^{\infty}(1-\frac{1}{p^2m^2})} \right)^2 \\
&= \left(\prod_{m=1}^{\infty} \left(1 - \frac{K_i^2-1}{p^2m^2-1} \right) \right)^2 \\
&< \left(1 - \frac{K_i^2-1}{p^2-1} \right)^2 \\
&< e^{-2(K_i^2-1)(p^2-1)},
\end{aligned}$$

where in the fifth line we use the infinite product representation of sine.

So

$$\begin{aligned}
|\mathbb{E}[\chi_j(f(X))]|^2 &= \prod_{i=1}^{r} |\mathbb{E}[\chi_j(X_i)]|^2 \\
&< \prod_{i=1}^{r} e^{-2(K_i^2-1)/(p^2-1)} \\
&< e^{2r/p^2} e^{-2(\sum_i K_i^2)/p^2}.
\end{aligned}$$

By the power mean inequality, $\sum_{i=1}^{r} K_i^2 \geq r \cdot (\prod_{i=1}^{r} K_i)^{2/r} = r2^{2k/r}$. Thus

$$|\mathbb{E}[\chi_j(f(X))]|^2 < e^{-\frac{2r(2^{2k/r}-1)}{p^2}}$$

Let $k = \delta r$. Then this quantity is $e^{-(2k/p^2)((2^{2\delta}-1)/\delta)}$. Since $(2^{2\delta}-1)/\delta$ is an increasing function of δ and goes to $2\ln 2$ as δ goes to 0, we have

$$|\mathbb{E}[\chi_j(f(X))]|^2 < e^{-(2k/p^2)((2^{2\delta}-1)/\delta)} < e^{-4(\ln 2)k/p^2} = 2^{-4\frac{k}{p^2}}$$

Then since by Lemma 6.6.8 $||f(X) - U_p||^2 < \max_{j\neq 0} |\mathbb{E}[\chi_j(f(X))]|^2$, $||f(X) - U_p|| < 2^{-2k/p^2}$. □

Now we show that if we divide the source into blocks and take the sum modulo p for each block, we get a convex combination of approximate symbol-fixing sources, which we can then use an expander walk to extract from.

Lemma 6.6.9. *For any prime $p \geq 2^\ell$ and any t, any flat independent source X on $(\{0,1\}^\ell)^r$ with total min-entropy k can be transformed in polynomial-time into a $(k', 1/p)$-approximate oblivious symbol-fixing source $f(X)$ on $[p]^{r'}$, where $r' = k/(2p^2 \log p)$ and $k' = k^2/(4np^2 \log^2 p)$.*

Proof. First divide X into $\frac{k}{2t}$ blocks consisting of $\frac{2t}{k}r$ smaller sources, for $t = p^2 \log p$. Then for each block take the sum modulo p of the smaller sources in the block. Then $f(X)$ is the concatenation of the resulting symbols for each block.

By Lemma 4.4.3, the number of blocks with min-entropy at least t is greater than $\frac{k^2}{4tr\ell} > \frac{k^2}{4tr\log p}$. For each of these blocks, by Lemma 6.6.7, we mix within $2^{-t/p^2} = \frac{1}{p}$ of uniform. □

Now, as we did for oblivious symbol-fixing sources, we use $f(X)$ as defined

above to take a random walk on an expander graph, which will mix to uniform by Lemma 4.3.8 and thus give us our extractor.

Theorem 6.6.10. *There exists an ε-extractor for the set of flat independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k that outputs $m = \Omega(k^2/(r2^{2\ell}\ell))$ bits and has error $\varepsilon = 2^{-m}$. This extractor is computable in time $\mathrm{poly}(r, 2^\ell)$.*

Proof. Let p be the least prime greater than 2^ℓ. Since by Bertrand's Postulate $p < 2\cdot 2^\ell$, this can easily be done in polynomial time in 2^ℓ by exhaustive search. Given a source X, first apply $f(X)$ from Lemma 6.6.9 to get a $(k', 1/p)$-approximate oblivious symbol-fixing source on $[p]^{r'}$, where $r' = k/(2p^2 \log p)$ and $k' = k^2/(4rp^2 \log^2 p)$. Then apply the extractor from Proposition 6.6.2 to $f(X)$, taking the graph G to be a p regular expander graph on 2^m vertices (for m to be given later). Specifically, assume G has $\lambda(G) \le \frac{1}{p^\alpha} - \frac{1}{\sqrt{p}}$ for some constant $\alpha < 1/2$. This can be achieved, for example, by taking G to be an $O(\log p)$ power of a constant degree expander with self loops added to make it degree p. Then by Proposition 6.6.2 $f(X)$ is within

$$\begin{aligned}
\varepsilon &\le \frac{1}{2}\left(\lambda(G) + \frac{1}{\sqrt{p}}\right)^{(k^2/4rp^2\log^2 p)} 2^{m/2} \\
&< p^{-(\alpha k^2/4rp^2\log^2 p)} 2^{m/2} \\
&= 2^{-((\alpha k^2/4rp^2\log p) - (m/2))}
\end{aligned}$$

of uniform. Then let $m = \alpha k^2/6rp^2 \log p$ so then $\varepsilon < 2^{-m}$. □

Combining this theorem with our reduction from general to flat sources, we get that this same extractor works for general total-entropy independent sources.

Theorem 6.6.11. *There exists an ε-extractor for the set of independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k that outputs $m = \Omega(k^2/r2^{2\ell}\ell)$ bits and has error $\varepsilon = 2^{-m}$. This extractor is computable in time $\mathrm{poly}(r, 2^\ell)$.*

Proof. Combine Theorem 6.6.10 and Lemma 6.6.3. □

6.7 Extracting More Bits From Total-Entropy Independent Sources

6.7.1 Seed Obtainers

Now that we have extractors for total-entropy independent sources, we can extract even more bits using the techniques that Gabizon et al. [GRS04] used to extract more bits out of oblivious bit-fixing sources. Assuming the entropy is high enough to use the extractors from Theorem 6.6.11, Theorem 6.4.6, or Corollary 6.5.2, we can extract almost all of the entropy. Their construction works by using an extractor for bit-fixing sources and a sampler to construct a seed obtainer. This seed obtainer outputs a source and a seed that is close to a convex combination of independent bit-fixing sources and uniform seeds. We generalize their definition of seed obtainer to total-entropy independent sources.

Definition 6.7.1. A function $F : (\{0,1\}^\ell)^r \to (\{0,1\}^\ell)^r \times \{0,1\}^d$ is a (k', ρ)-seed obtainer for all independent sources X on $(\{0,1\}^\ell)^r$ with total min-entropy k if the distribution $R = F(X)$ can be expressed as a convex combination of distributions $R = \eta Q + \sum_a \alpha_a R_a$ (where the coefficients η and α_a are nonnegative and $\eta + \sum_a \alpha_a = 1$) such that $\eta \leq \rho$ and for every a there exists an independent source Z_a on $(\{0,1\}^\ell)^r$ with min-entropy k' such that R_a is ρ-close to $Z_a \otimes U_d$.

Now, as in the bit-fixing case, we can use a seeded extractor for total-entropy independent sources together with a seed obtainer to construct a deterministic extractor for total-entropy independent sources. The proof for the following Theorem

is the same as the proof for the bit-fixing case in [GRS04]. We include it here for the sake of completeness.

Theorem 6.7.2. *Let $F : (\{0,1\}^\ell)^r \to (\{0,1\}^\ell)^r \times \{0,1\}^t$ be a (k',ρ)-seed obtainer for independent sources X on $(\{0,1\}^\ell)^r$ with total min-entropy k. Let $E_1 : (\{0,1\}^\ell)^r \times \{0,1\}^d \to \{0,1\}^m$ be a seeded ε-extractor for independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k. Then $E : (\{0,1\}^\ell)^r \to \{0,1\}^m$ defined by: $E(x) = E_1(F(x))$ is a deterministic $(\varepsilon+2\rho)$-extractor for independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k.*

Proof. By the definition of a seed obtainer we have that $E(X) = \eta E_1(Q) + \sum_a \alpha_a E_1(R_a)$ for some $\eta \leq \rho$. For each a we have that R_a is ρ-close to $Z_a \otimes U_d$, so $E_1(R_a)$ is ρ-close to $E_1(Z_a \otimes U_d)$, which is itself ε-close to U_m since E_1 is an ε-extractor. Thus $E_1(R_a)$ is $(\varepsilon+\rho)$-close to U_m, which implies that $E(X)$ is $(\varepsilon+\rho)$-close to $\eta E_1(Q) + (1-\eta)U_m$. Therefore by Lemma 2.4.4 we have that $E(X)$ is $(\eta+\varepsilon+\rho)$-close to uniform. The lemma follows because $\eta \leq \rho$. □

To construct seed obtainers, we need to extend the definition of averaging samplers from [GRS04] to general functions as follows. This definition is similar in spirit to that of Vadhan in [Vad04], except the sample size is not fixed and we both upper and lower bound the total value of the sample.

Definition 6.7.3. A function $Samp : \{0,1\}^t \to P([r])$ is a $(\delta,\theta_1,\theta_2,\gamma)$ averaging sampler if for every function $f : [r] \to [0,1]$ with average value $\frac{1}{r}\sum_i f(i) = \delta$, it holds that

$$\Pr_{w \leftarrow U_t}\left[\theta_1 \leq \sum_{i \in Samp(w)} f(i) \leq \theta_2\right] \geq 1-\gamma.$$

When applying these samplers to total-entropy independent sources, we get the following lemma.

Lemma 6.7.4. *Let $Samp : \{0,1\}^t \to P([r])$ be a $(\delta, \delta_1 r, \delta_2 r, \gamma)$ averaging sampler. Then for any independent source X on $(\{0,1\}^\ell)^r$ with total min-entropy $k = \delta r \ell$, we have*

$$\Pr_{w \leftarrow U_t}[\delta_1 r\ell \leq H_\infty(X_{Samp(w)}) \leq \delta_2 r\ell] \geq 1 - \gamma.$$

Proof. Let $f(i) = H_\infty(X_i)/\ell$. □

Given these definitions, we can show that essentially the same construction from Gabizon et al. [GRS04] for bit-fixing seed obtainers works for total-entropy independent source seed obtainers.

Theorem 6.7.5. *Let $Samp : \{0,1\}^t \to P([r])$ be a $(\delta, \delta_1 r, \delta_2 r, \gamma)$ averaging sampler and $E : (\{0,1\}^\ell)^r \to \{0,1\}^m$ be an ε-extractor for independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy $k = \delta_1 r\ell$. Then $F : (\{0,1\}^\ell)^r \to (\{0,1\}^\ell)^r \times \{0,1\}^{m-t}$ defined as follows is a (k', ρ)-seed obtainer for independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy $k = \delta r\ell$ with $k' = (\delta - \delta_2) r\ell$ and $\rho = \max(\varepsilon + \gamma, \varepsilon \cdot 2^{t+1})$.*

The Construction of F:

- *Given $x \in (\{0,1\}^\ell)^r$ compute $z = E(x)$. Let $E_1(x)$ denote the first t bits of $E(x)$ and $E_2(x)$ denote the remaining $m - t$ bits.*
- *Let $T = Samp(E_1(x))$.*
- *Let $x' = x_{[r]\setminus T}$. If $|x'| < n$ we pad it with zeroes to get an r source long string.*
- *Let $y = E_2(x)$. Output x', y.*

The proof of this theorem is almost exactly the same as the proof in [GRS04], except substituting independent sources and the associated sampler and extractor for bit-fixing sources, so we omit it here. This theorem also follows from the main theorem of [Sha06].

6.7.2 Constructing Samplers

In order to use the seed obtainer construction to extract more bits, we first need a good averaging sampler. We will show that the same sampler construction given in Gabizon et al. [GRS04] generalizes to our definition. Our sampler works by generating d-wise independent variables $Z_1, \ldots, Z_r \in [b]$ and letting $Samp(U_t) = \{i | Z_i = 1\}$.

Lemma 6.7.6. *For all δ and integers r, b, t such that $b/r \leq \delta \leq 1$ and $6 \log r \leq t \leq \frac{\delta r \log r}{20b}$ there is a polynomial-time computable $(\delta, \frac{\delta r}{2b}, \frac{3\delta r}{b}, 2^{-\Omega(t/\log r)})$ averaging sampler $Samp : \{0,1\}^t \to P([r])$*

The following tail inequality for d-wise independent variables is due to Bellare and Rompel [BR94].

Theorem 6.7.7. *[BR94] Let $d \geq 6$ be an even integer. Suppose that $X_1, \ldots, X_r$ are d-wise independent random variables taking values in $[0,1]$. Let $Y = \sum_{1 \leq i \leq r} Y_i$, $\mu = \mathbb{E}[Y]$, and $A > 0$. Then*

$$\Pr[|Y - \mu| \geq A] \leq 8 \left(\frac{d\mu + d^2}{A^2} \right)^{d/2}$$

Proof. (of Lemma 6.7.6) Let d be the largest even integer such that $d \log r \leq t$ and let $q = \lfloor \log b \rfloor \leq \log r$. Use $d \log r$ random bits to generate r d-wise independent random variables $Z_1, \ldots, Z_r \in \{0,1\}^q$ using the construction from [CW79]. Fix $a \in \{0,1\}^q$. Let the random variable denoting the output of the sampler be $Samp(U_t) = \{i | Z_i = a\}$. For $1 \leq i \leq r$, define a random variable Y_i that is set to $f(i)$ if $i \in Samp(U_t)$ and 0 otherwise. Let $Y = \sum_i Y_i$ (note that Y is exactly the sum we wish to bound). Note that $\mu = \mathbb{E}[Y] = \delta r / 2^q$ and that the random variables $Y_1, \ldots, Y_r$ are

d-wise independent. Applying Theorem 6.7.7 with $A = \delta r/2b$,

$$\Pr[|Y-\mu| \geq A] \leq 8\left(\frac{d\frac{\delta r}{2^q}+d^2}{A^2}\right)^{d/2}.$$

Note that

$$\begin{aligned}\{|Y-\mu|<A\} &\subseteq \{\frac{\delta r}{2^q}-A<Y<\frac{\delta r}{2^q}+A\} \subseteq \{\frac{\delta r}{b}-A<Y<\frac{2\delta r}{b}+A\}\\ &\subseteq \{\frac{\delta r}{2b}\leq Y\leq \frac{3\delta r}{b}\} = \{\frac{\delta r}{2b}\leq \sum_{i\in Samp(w)} f(i)\leq \frac{3\delta n}{b}\}.\end{aligned}$$

Note that $d \leq \frac{t}{\log r} \leq \frac{\delta r}{20b}$ by assumption. We conclude that

$$\begin{aligned}\Pr_{w\leftarrow U_t}\left[\frac{\delta r}{2b}\leq \sum_{i\in Samp(w)} f(i)\leq \frac{3\delta r}{b}\right] &\geq 1-8\left(\frac{d\frac{\delta r}{2^q}+d^2}{(\delta r/2b)^2}\right)^{d/2}\\ &\geq 1-8\left(\frac{4b^2}{(\delta r)^2}\left(\frac{2d\delta r}{b}+\frac{d\delta r}{20b}\right)\right)^{d/2}\\ &\geq 1-8\left(\frac{10db}{\delta r}\right)^{d/2}\\ &\geq 1-2^{-(d/2+3)} \geq 1-2^{-\Omega(t/\log r)}.\end{aligned}$$

□

6.7.3 Extractors From Seed Obtainers

As in [GRS04] it will be convenient to combine Theorem 6.7.2 and Theorem 6.7.5 to get the following theorem.

Theorem 6.7.8. *Assume we have the following:*

- *A* $(\delta, \delta_1 r, \delta_2 r, \gamma)$ *averaging sampler* $Samp : \{0,1\}^t \to P([r])$.

- *A deterministic ε^*-extractor for total-rate δ_1 independent sources* $E^* : (\{0,1\}^\ell)^r \to \{0,1\}^{m'}$.

- *A seeded ε_1-extractor for total-rate $\delta - \delta_2$ independent sources* $E_1 : (\{0,1\}^\ell)^r \times \{0,1\}^s \to \{0,1\}^m$, *where* $m' \geq s+t$.

Then we get a deterministic ε-extractor for total-rate δ independent sources $E : (\{0,1\}^\ell)^r \to \{0,1\}^m$ *where* $\varepsilon = \varepsilon_1 + 3 \cdot \max(\varepsilon^* + \gamma, \varepsilon^* \cdot 2^{t+1})$.

We will use the following seeded extractor from Raz, Reingold, and Vadhan [RRV99].

Theorem 6.7.9. *[RRV99] For any r,k, and $\varepsilon > 0$, there exists a seeded ε-extractor $Ext : \{0,1\}^r \times \{0,1\}^s \to \{0,1\}^m$ for all sources with min-entropy k, where $m = k$ and $s = \Theta(\log^2 r \cdot \log(1/\varepsilon) \cdot \log m)$.*

Combining the extractor from [RRV99] with the sampler from the previous section, we get the following general corollary, which shows how to transform a deterministic extractor that extracts just some of the min-entropy into one that extracts almost all of the min-entropy.

Corollary 6.7.10. *Let $\delta, \delta_1, \varepsilon_1$ and integers r,t be such that $\delta_1 \geq 1/2r$ and $6 \log r \leq t \leq \frac{\delta_1 r \log r}{10}$. Also let $m = (\delta - 6\delta_1) r\ell$ and $s = \Theta(\log^2(r\ell) \cdot \log(1/\varepsilon_1) \cdot \log m)$. Then given any deterministic ε^*-extractor for total-rate δ_1 independent sources $E^* : (\{0,1\}^\ell)^r \to \{0,1\}^{m'}$ with $m' \geq s+t$, we can construct an ε-extractor for total-rate δ independent sources $E : (\{0,1\}^\ell)^r \to \{0,1\}^m$ where $\varepsilon = \varepsilon_1 + 3 \cdot \max(\varepsilon^* + 2^{-\Omega(t/\log r)}, \varepsilon^* \cdot 2^{t+1})$.*

Proof. Combine Lemma 6.7.6 with $b = \delta/2\delta_1$, Theorem 6.7.9, and Theorem 6.7.8. □

Now we can use Corollary 6.7.10 together with our previous deterministic extractor construction from Theorem 6.6.11 to show how we can extract nearly all of the entropy from total-entropy independent sources with sufficiently high min-entropy, proving Theorem 6.1.7.

Proof. (Of Theorem 6.1.7.) Use the construction from Corollary 6.7.10 with the extractor from Theorem 6.6.11 as E^* and let $\varepsilon_1 = 2^{-\Omega((\delta_1^2 r\ell)(2^{2\ell}\log^3 r))}$ and $t = \Omega(\frac{\delta_1^2}{2^{2\ell}} r\ell)$. Then it's not hard to see that (choosing appropriate constants) these values satisfy $6\log r \leq t \leq \frac{\delta_1 r \log r}{10}$ and $m' \geq s + t$ for sufficiently large r. □

The extractor for small-space sources from Theorem 6.1.4 is then obtained by combining Theorem 6.1.7 with Lemma 6.3.1.

We could also use a seed obtainer together with the extractor for constant rate sources from Theorem 6.4.6. This lets us extract any constant fraction of the entropy and proves Theorem 6.1.6.

Proof. (Of Theorem 6.1.6.) Use the construction from Corollary 6.7.10 with the extractor from Theorem 6.4.6 as E^* and let $\varepsilon_1 = 2^{-\Omega((r\ell)/(\log^3(r\ell)))}$ and $t = \Theta(r\log(\min(2^\ell, r)))$. Then it's not hard to see that (choosing appropriate constants) these values satisfy $6\log r \leq t \leq \frac{\delta_1 r \log r}{10}$ and $m' \geq s + t$ for sufficiently large r. □

The extractor for small-space sources from Theorem 6.1.3 is then obtained by combining Theorem 6.1.7 with Lemma 6.3.1.

We can also apply this construction to the polynomial entropy rate extractor from Corollary 6.5.2, which proves Theorem 6.1.5.

Proof. (Of Theorem 6.1.5.) Use the construction from Corollary 6.7.10 with the extractor from Corollary 6.5.2 as E^* and let $\varepsilon_1 = 2^{-(\delta_1^2 r\ell)^{\Omega(1)}/(\log^3(r\ell))}$ and $t = (\delta_1^2 r\ell)^{\Omega(1)}$.

Then it's not hard to see that (choosing appropriate constants) these values satisfy $6\log r \leq t \leq \frac{\delta_1 r \log r}{10}$ and $m' \geq s+t$ for sufficiently large r. □

The extractor for small-space sources from Theorem 6.1.1 is then obtained by combining Theorem 6.1.5 with Lemma 6.3.1.

6.7.4 Extractors For Smaller Entropy

Gabizon et. al [GRS04] also showed how to use seed obtainers to extract more bits even when the initial extractor only extracts $\Theta(\log k)$ bits, which they're able to get from the cycle walk extractor described in Chapter 4. We can generalize their construction to work for total-entropy independent sources, which together with our generalization of the cycle walk extractor allows us to extract more bits from smaller entropy rates.

In order to get a seed obtainer that can use only $\Theta(\log k)$ bits, we need both a sampler and a seeded extractor for total-entropy independent sources. To do so, as in [GRS04], we use d-wise ε-dependent random variables to both sample and partition. The proofs of the following two lemmas easily generalize the construction from [GRS04] in a similar way to our earlier sampler construction.

Lemma 6.7.11. *For any constant $0 < \alpha < 1$, there exist constants $c > 0$ and $0 < b < 1/2$ (both depending on α) such that for any $r \geq 16$ and $k = \delta r\ell \geq \log^c r$, the following holds. There is a polynomial-time computable $(\delta, \delta r/2k^b, 3\delta r/k^b, O(k^{-b}))$ sampler $Samp : \{0,1\}^t \to P([r])$ where $t = \alpha \cdot \log k$.*

Lemma 6.7.12. *Fix any constant $0 < \alpha < 1$. There exist constants $c > 0$ and $0 < b < 1/2$ (both depending on α) such that for any $r \geq 16$ and $k = \delta r\ell \geq \log^c r$, we can use $\alpha \cdot \log k$ random bits to explicitly partition $[r]$ into $m = \Theta(k^b)$ sets $T_1, \ldots, T_m$*

such that for every function $f : [r] \to [0,1]$ *with average value* $\frac{1}{r}\sum_i f(i) = \delta$,

$$\Pr\left[\forall i, \delta r/2k^b \leq \sum_{j \in T_i} f(j) \leq 3\delta r/k^b\right] \geq 1 - O(k^{-b}).$$

As in Lemma 6.7.6, this lemma implies that if we partition a total-rate δ independent source, with high probability each T_i has some min-entropy.

Corollary 6.7.13. *For any constant* $0 < \alpha < 1$*, there exist constants* $c > 0$ *and* $0 < b < 1/2$ *(both depending on* α*) such that for any* $r \geq 16$ *and* $k \geq \log^c r$*, the following holds. We can use* $\alpha \cdot \log k$ *random bits to explicitly partition* $[r]$ *into* $m = \Theta(k^b)$ *sets* $T_1, \ldots, T_m$ *such that for any set of independent sources* X *on* $(\{0,1\}^\ell)^r$ *with total min-entropy* k*,*

$$\Pr\left[\forall i, k^{1-b}/2 \leq H_\infty(X_{T_i}) \leq 3k^{1-b}\right] \geq 1 - O(k^{-b}).$$

Now we will use this partitioning to construct a seeded extractor for total-entropy independent sources that uses a small seed. As in [GRS04] once we partition the source, we apply an extractor to each part. The extractor we will use is our sum mod p extractor.

Theorem 6.7.14. *For any constant* $0 < \alpha < 1$*, there exist constants* $c > 0$ *and* $0 < b < 1/2$ *(both depending on* α*) such that for any* $r \geq 16$*,* $k \geq \log^c r$*,* $0 < \delta \leq 1$ *and* $\ell \leq \log(k^{(1-b)/2}/\sqrt{\log k^{2b}})$*, the following holds. There is a polynomial-time computable seeded* ε*-extractor* $E : (\{0,1\}^\ell)^r \times \{0,1\}^s \to \{0,1\}^m$ *for independent sources on* $(\{0,1\}^\ell)^r$ *with total min-entropy* $k = \delta r\ell$*, with* $s = \alpha \cdot \log k$*,* $m = \Theta(k^b\ell)$ *and* $\varepsilon = O(k^{-b})$*.*

Proof. As stated above, E works by first partitioning the input x into $m' = \Theta(k^b)$

parts $T_1, \ldots, T_{m'}$ using Corollary 6.7.13. Next we find the next largest prime $p \geq 2^\ell$, which by Bertrand's postulate is at most $2 \cdot 2^\ell$, so we can find it efficiently by brute force search. Then for each T_i we compute $z_i = \sum_{j \in T_i} x_j \mod p$ and output $z = z_i, \ldots, z_{m]}$.

Let Z be the distribution of the output string z. Let A be the "good" event that all sets T_i have entropy at least $k^{1-b}/2$. Then we decompose Z as

$$Z = \Pr[A^c] \cdot (Z|A^c) + \Pr[A] \cdot (Z|A).$$

Now by Corollary 6.7.13, $\Pr[A] \geq 1 - O(k^{-b})$. By Corollary 6.6.6, $(Z|A)$ is $m' \cdot 2^{-\Omega(k^{1-b}/2^{2\ell})}$ close to uniform. Since $\ell \leq \log(k^{(1-b)/2}/\sqrt{\log k^{2b}})$, $(Z|A)$ is $O(k^{-b})$ close to uniform. Thus by Lemma 2.4.4, Z is $O(k^{-b})$ close to uniform. □

Now we are ready to combine these ingredients using Theorem 6.7.8 to get an improved extractor.

Theorem 6.7.15. *There exist constants $c > 0$ and $0 < b < 1/2$ such that for $k \geq \log^c r$ and $2^\ell \leq O(k^{(1-b)/2}/\sqrt{\log k^{2b}})$, the following holds. There exists a polynomial-time computable ε-extractor $E : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for independent sources on $(\{0,1\}^\ell)^r$ with min-entropy k, where $m = \Theta(k^b \ell)$ and $\varepsilon = O(k^{-b})$.*

Proof. Use Theorem 6.7.8 together with the sampler from Lemma 6.7.11, the deterministic extractor from Corollary 6.6.6, and the seeded extractor from Theorem 6.7.14 □

This still doesn't get all of the entropy out of the source, but now we have a long enough output that we can use the seeded extractor from Theorem 6.7.9 to get the rest of the entropy, which proves Theorem 6.1.8.

Proof. (Of Theorem 6.1.8.) Use Theorem 6.7.8 together with the sampler from Lemma 6.7.11, the deterministic extractor from Theorem 6.7.15, and the seeded extractor from Theorem 6.7.9. □

6.8 Doing Better For Width Two

In this section we consider the case of space 1 (width 2) sources where the output bit is restricted to be the same as the label of the next state, which we will call *restricted width two sources*. For such sources, we can improve our results by decreasing the alphabet size in the total-entropy independent sources. This will allow us to extract from smaller entropy rates. We will need the following class of sources.

Definition 6.8.1. A previous-bit source on $\{0,1\}^n$ with min-entropy k has at least k uniformly random bits and the rest of the bits are functions of the previous bit.

We will show that restricted width two sources are close to a convex combination of previous-bit sources, and then show that these previous bit sources can be converted into total-entropy independent sources with small alphabet size.

6.8.1 Extracting From Previous-Bit Sources

To convert a previous-bit source to a total-entropy independent source, we first divide the source into blocks as before, but instead of simply viewing each block as a binary number, we apply a function to reduce the alphabet size while still maintaining some of the entropy. Specifically, we will show that if a block has at least one random bit, then the output symbol will have at least one bit of entropy. The main lemma is as follows.

Lemma 6.8.2. *Any length n previous-bit source X with min-entropy k can be converted in polynomial time to a convex combination of flat independent sources on $(\{0,1\}^\ell)^r$ with min-entropy k', where $r = \frac{k}{2}$, $k' = k^2/4n$ and $\ell = \lceil \log\left(\frac{2n}{k} + 1\right) \rceil$.*

The following lemma shows that any block that contains at least one random bit will give a random source.

Lemma 6.8.3. *We can construct a function $f : \{0,1\}^t \to \{0,1\}^{\lceil \log(t+1) \rceil}$ so that for any previous-bit source Y on $\{0,1\}^t$ with exactly one random bit, f attains different values depending on whether the random bit in Y is set to 0 or 1.*

Proof. For $0 \leq i \leq t$, let $z_i \in \mathbb{Z}_2^{\lceil \log(t+1) \rceil}$ be the standard representation of i as a vector over $\mathbb{Z}_2$. (More generally, we only require the z_i to be distinct vectors.) Then $f(y) = \sum_{i=1}^{t} y_i(z_i - z_{i-1})$.

Let y_0 (y_1) be Y with the random bit set to 0 (1). Now we show that $f(y_0) \neq f(y_1)$. We see that

$$f(y_0) - f(y_1) = \sum_{i=1}^{t} (y_{0i} - y_{1i})(z_i - z_{i-1}).$$

It's easy to see that $y_{0i} - y_{1i}$ will be 0 for all fixed bits and 1 whenever the random bit or its negation appears. For our sources, all appearances of the random bit must appear consecutively. This means that if the random bit appears from positions j through k, $f(y_0) - f(y_1) = z_k - z_{j-1}$, since all of the other terms cancel. Thus since $z_k \neq z_{j-1}$, $f(y_0) - f(y_1) \neq 0$. □

Now we can prove Lemma 6.8.2.

Proof. Divide X into $r = k/2$ blocks of size $n/r = 2n/k$. Then apply the function f from Lemma 6.8.3 to each block to get Y.

To see that this works, fix all of the random bits that cross between blocks. Also, for each block fix all but one of the random bits that are contained within the block. Now X is a convex combination of all of the sources given by every possible such fixing. Let X' be a source corresponding to one particular fixing. We will show that if we apply f to every block of X', we will get a source with enough random blocks. Any block of X' with a random source is a previous-bit source with one random bit, so we can apply Lemma 6.8.3 to see that the output of f on this block is uniformly chosen from among two different sources, as desired.

Now we just need to see how many blocks with at least one random bit there are. There can be at most r random bits that cross between blocks. So removing those bits we are left with at least $k - r = k/2$ random bits. These $k/2$ random bits must be contained in at least $k' = (k/2)/(n/r) = k^2/4n$ different blocks, which gives us the desired bound. □

Now we can combine Theorem 6.1.7 and Lemma 6.8.2 to get an extractor for previous-bit sources.

Theorem 6.8.4. *There exists a polynomial-time computable ε-extractor for the set of previous-bit sources of length n with min-entropy k that outputs $m = \frac{k^2}{8n}$ bits and has error $\varepsilon = \exp(-\Omega(k^5/(n^4 \log(n/k) \log^3 k)))$.*

Proof. Given a source X, apply Lemma 6.8.2 to convert X into a convex combination of flat independent sources on $(\{0,1\}^\ell)^r$ with total min-entropy k', where $r = \frac{k}{2}$, $k' = \frac{k^2}{4n}$, and $\ell' = \lceil \log\left(\frac{2n}{k} + 1\right) \rceil$. Then apply the extractor from Theorem 6.1.7 with $\zeta = k^2/(48n \cdot r\ell)$. □

6.8.2 Restricted Width Two Sources As Convex Combinations Of Previous-Bit Sources

To show we can extract from restricted width two sources, we will prove that these sources can be viewed as convex combinations of previous bit sources. With high probability, these previous-bit sources will have sufficient entropy so that our extractor from the previous section will work.

Lemma 6.8.5. *Any length n restricted width two source X with min-entropy k is a convex combination of length n previous bit sources Z_j so that with probability at least $1-2^{-k/4}-e^{-9k'^2/2n}$, the sources Z_j have at least $k' = \min(k/48\log(n/k), k/96)$ random bits.*

To get our extractor, we just combine this lemma with the extractor from Theorem 6.8.4.

Theorem 6.8.6. *There exists a polynomial-time computable ε-extractor for the set of length n restricted width two sources with min-entropy k that outputs $m = \Omega(k^2/n(\max(\log(n/k),1))^2)$ bits and has error $\varepsilon = 2^{-\Omega((k')^5/(n^4\log(n/k')\log^3 k')}$, where $k' = \min(k/48\log(n/k), k/96)$.*

Proof. By Lemma 6.8.5 our source X is $2^{-k/4}+e^{-9k'^2/2n}$ close to a convex combination of length n previous-bit sources with $k' = \min(k/48\log(n/k), k/96)$ random bits. We can then apply the extractor from Theorem 6.8.4 to get out $m = \frac{(k')^2}{8n} = \Omega(k^2/n(\max(\log(n/k),1))^2)$ bits. □

Notice that here we only need $k \gg n^{4/5}$ whereas before we required $k \gg n^{1-\eta}$ for some small constant η.

Now we describe how we express the restricted width two source X as a convex combination of previous-bit sources Z_j. This is done recursively on the

layers of the branching program for the source. We say we are in a given state at each layer; either "open", "closed at 0", or "closed at 1". Each sequence of states corresponds to a previous-bit source. The way we divide the next layer up depends on the state we are in. The high level picture is that each random bit corresponds to going into the open state, which we are in until we get a fixed bit, which takes us to the corresponding closed state. We stay closed until another random bit occurs. An example is shown in Figure 6.8.2.

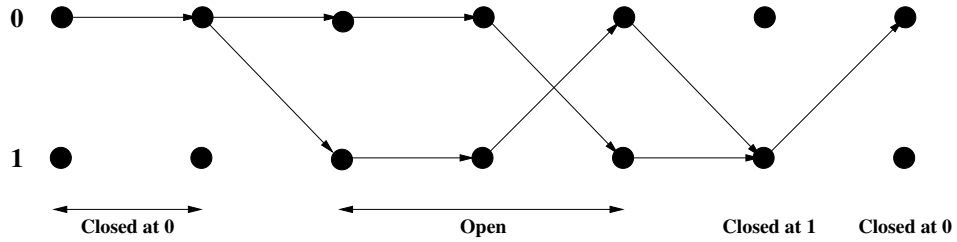

Figure 6.2: A previous-bit source viewed as a restricted width two source. This source consists of the bits $0, 0, r, r, \bar{r}, 1, 0$, where r is a random bit.

More formally, we define the following probabilities, shown in Figure 6.8.2.

$$p_{i0} = \Pr[X_i = 0 | X_{i-1} = 0]$$
$$p_{i1} = \Pr[X_i = 1 | X_{i-1} = 0]$$
$$q_{i0} = \Pr[X_i = 0 | X_{i-1} = 1]$$
$$q_{i1} = \Pr[X_i = 1 | X_{i-1} = 1]$$

First, we describe what happens if we are currently in the open state. The next bit is fixed to 0 (resp. 1) and the state becomes closed at 0 (1) with probability $\min(p_{i0}, q_{i0})$ ($\min(p_{i1}, q_{i1})$). Else we stay in the open state and the next bit is either equal to the previous bit or the negation of the previous bit depending on which edges have the remaining probability.

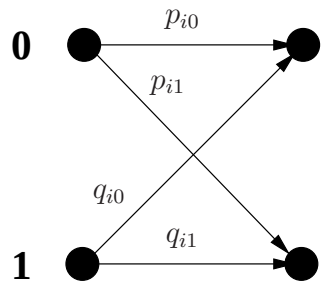

Figure 6.3: The probabilities for a single bit of a restricted width two source.

If we are closed at 0, the next bit is random and we go into the open state with probability $2\min(p_{i0}, p_{i1})$. If $p_{i0} < p_{i1}$, the next bit is fixed to 1 and we go into the closed at 1 state with probability $1 - 2p_{i0}$. Else the next bit is fixed to 0 and we go into the closed at 0 state with probability $1 - 2p_{i1}$.

If we are closed at 1, the next bit is random and we go into the open state with probability $2\min(q_{i0}, q_{i1})$. If $q_{i0} < q_{i1}$, the next bit is fixed to 1 and we go into the closed at 1 state with probability $1 - 2q_{i0}$. Else the next bit is fixed to 0 and we go into the closed at 0 state with probability $1 - 2q_{i1}$.

Now we show that with high probability, the sources in the convex combination have sufficient min-entropy. We do this by looking at the relationships between paths in the original source X and the min-entropy of the Z_j. First, note that each path in the branching program corresponds to an output value of X, so each path has probability at most 2^{-k}. Note that the min-entropy of Z_j is equal to the number of openings in Z_j.

Each path can be divided into edges that are the most probable edge coming out of a node and those that are the least probable. We will show how the number of least probable edges on a path in X relates to the min-entropy of a Z_j that contains this path. First note that every least probable edge corresponds to either an

opening, a closing, or what we call a "false closing". A false closing is defined as transitioning from the open state to the open state yet still taking a least probable edge. Let $C(Z_j)$ denote the number of closings in Z_j, $A(Z_j)$ denote the number of openings, and $B(Z_j)$ denote the number of false closings.

If we could ignore the false closings, showing that with high probability we take the least probable edge a large number of times would be enough. Since $C(Z_j) \leq A(Z_j)$, this would imply that with high probability $A(Z_j)$ is large, and thus the Z_j have large min-entropy with high probability. To take account of the false closings, we also have to show that there aren't too many of them, which we will do by a martingale argument.

First, we show that with high probability over all paths in X, we take the least probable edge a large number of times.

Lemma 6.8.7. *For any length n restricted width two source with min-entropy k, the total probability of all paths that have at most* $t = \min(k/8\log(n/k), k/16)$ *least probable edges is less than* $2^{-k/4}$.

Proof. Since the source has min-entropy k, each path has probability at most 2^{-k}. There are $\binom{n}{i}$ paths that have i least probable edges. Thus the total probability of all paths that have at most t least probable edges is at most

$$2^{-k}\sum_{i=0}^{t}\binom{n}{i} \leq 2^{-k}2^{nH(t/n)} < 2^{-k+2t\log(n/t)}$$

where $H(t/n)$ is the standard Shannon entropy $H(p) = -p\log p - (1-p)\log(1-p)$.

Suppose $k \leq n/4$. Then $s = k/8\log(n/k)$, so

$$2s\log\frac{n}{t} = \frac{k}{4}\left(1 + \frac{\log(8\log\frac{n}{k})}{\log\frac{n}{k}}\right) \leq \frac{3k}{4}.$$

If $k > n/4$, then $t = \frac{k}{16}$, so

$$2t\log\frac{n}{t} = \frac{k}{8}\left(4 + \log\frac{n}{k}\right) \leq \frac{3k}{4}.$$

Thus the probability of taking at most t least probable edges is at most $2^{-k+2t\log(n/t)} \leq 2^{-k/4}$. □

To show that the number of false closings is small, we first define a submartingale that is equal to the number of closings minus the number of false closings after the first i bits. Then we use the following simple variant of Azuma's inequality for submartingales (see [Wor99] for a proof).

Definition 6.8.8. A submartingale with respect to a random process $G_0, G_1, \ldots$, with G_0 fixed, is a sequence $Y_0, Y_1, \ldots$ of random variable defined on the random process such that

$$\mathbb{E}[Y_{i+1}|G_0, G_1, \ldots, G_i] \geq Y_i$$

for all $i \geq 0$.

Lemma 6.8.9. *Let $Y_0, Y_1, \ldots, Y_n$ be a submartingale with respect to $G_0, G_1, \ldots, G_n$ where $Y_0 = 0$ and $|Y_i - Y_{i-1}| \leq 1$ for $i \geq 1$. Then for all $\alpha > 0$,*

$$\Pr[Y_n \leq -\alpha] \leq e^{-\alpha^2/2n}.$$

Now we are ready to prove that with high probability the number of false

closings can't be too large.

Lemma 6.8.10. *For all* $\alpha > 0$*,*

$$\Pr[B(Z_j) \geq C(Z_j) + \alpha] \leq e^{-\alpha^2/2n}.$$

Proof. Let Y_i be the number of closings from $X_1, \ldots, X_i$ minus the number of false closings from $X_1, \ldots, X_i$ and let $Y_0 = 0$. Let $G_0, G_1, \ldots, G_n$ be the random process for dividing X into previous-bit sources, so G_i is the state after the first i bits have been divided.

Now we show that $Y_0, \ldots, Y_n$ is a submartingale with respect to $G_0, G_1, \ldots, G_n$. If we are in a closed state after i bits, then we have no closings or false closings at $i+1$, so $\mathbb{E}[Y_{i+1}|G_0, G_1, \ldots, G_i] = Y_i$. If we are in an open state at i, we show that if we have the possibility of a false closing at $i+1$, then the probability of closing is greater than $1/2$, and in particular is greater than the probability of a false closing. This would imply that $\mathbb{E}[Y_{i+1}|G_0, G_1, \ldots, G_i] \geq Y_i$, as desired. First, note that the probability of closing at $i+1$ is

$$\min(p_{i+1,0}, q_{i+1,0}) + \min(p_{i+1,1}, q_{i+1,1}) = \min(p_{i+1,0} + q_{i+1,1}, q_{i+1,0} + p_{i+1,1}).$$

Suppose without loss of generality that $p_{i+1,0} + q_{i+1,1} \geq q_{i+1,0} + p_{i+1,1}$, so we close with probability $q_{i+1,0} + p_{i+1,1}$. In this case, the edges we would take in a false closing are the 00 and 11 edges. So if we have a false closing, either $p_{i+1,0} \leq 1/2$ or $q_{i+1,1} \leq 1/2$, which implies either $p_{i+1,1} \geq 1/2$ or $q_{i+1,0} \geq 1/2$, and thus the probability of closing is at least $1/2$.

By the definition of Y_i, $|Y_i - Y_{i-1}| \leq 1$, so we can apply Lemma 6.8.9 to get

$$\Pr[Y_n \leq -\alpha] \leq e^{-\alpha^2/2n},$$

which implies the desired result.

□

Now we are finally ready to prove Lemma 6.8.5.

Proof. (Of Lemma 6.8.5.)

First, express the restricted width two source X as a convex combination of previous-bit sources Z_j as described previously, so $X = \sum_j \alpha_j Z_j$. Now consider a randomly chosen Z_j, chosen with probability α_j. The number of random bits in Z_j is equal to the number of openings $A(Z_j)$. Since the number of closings is either equal to or one less than the number of openings, either $C(Z_j) = A(Z_j)$ or $C(Z_j) = A(Z_j) - 1$. So if we can prove with high probability that $C(Z_j)$ is large, then with high probability the number of random bits in Z_j is also large. For every path in Z_j, every least probable edge on the path corresponds to either an opening, a closing, or a false closing. Thus the probability that $A(Z_j) + B(Z_j) + C(Z_j) \geq s$ is at least the probability over all paths that the path has at least s least probable edges. Thus we can apply Lemma 6.8.7 and get

$$\Pr[B(Z_j) + 2C(Z_j) \geq s - 1] \geq \Pr[A(Z_j) + B(Z_j) + C(Z_j) \geq s] > 1 - 2^{-k/4}$$

for $s = \min(k/8\log(n/k), k/16)$.

By Lemma 6.8.10,

$$\Pr[B(Z_j) < C(Z_j) + \frac{s}{2}] \geq 1 - e^{-s^2/8n}.$$

With high probability both of these events occur, so

$$\Pr[C(Z_j) \geq \frac{s}{6}] \geq 1 - 2^{-k/4} - e^{-s^2/8n}.$$

□

6.9 Open Questions

The main area for improvement to our constructions for total-entropy independent sources is in getting more bits out for smaller min-entropies. For large source lengths ℓ, we can only extract when the min-entropy rate is at least $(r\ell)^{-\eta}$ for some constant η. Our extractors based on our extractors for oblivious bit-fixing sources work for smaller min-entropy, but require that $\ell < \frac{1}{2}\log r$. It would be nice to get extractors which worked for large source lengths and smaller min-entropy. We note that the constant η in our construction is small. So even constructing an extractor that works for min-entropy rate $(r\ell)^{-1/2}$ for large ℓ, roughly matching our extractor for small ℓ, would be very interesting. Nonconstructively we can even achieve arbitrarily long source lengths ℓ for even much smaller min-entropies $k = \Omega(\ell + \log r)$ (see Theorem 3.3.4), so our current bounds fall well short of what is possible.

For small-space sources, even our best construction requires that the min-entropy is at least $n^{-\eta}$ for some small constant η. This min-entropy requirement is quite high, and unlike in other cases, we can't even do anything for smaller min-entropies. Thus the main area for future improvement is in decreasing the min-entropy requirement for extracting from small space sources. Even if we could extract fewer bits for smaller min-entropy it would still be a great improvement, although ideally we would like to extract as close to k bits as possible.

The other area for improvement is in the space requirement for our extractors. Non explicitly, we can get extractors for space even up to k, so we would like to get as close to this bound as possible. For constant rate sources, our constructions are within a constant factor of the non-explicit bound, but for sub-constant rate sources, we have larger gaps.

Chapter 7

Affine Sources

7.1 Overview Of Our Results

Recall from Section 2.2 that affine sources are sources distributed uniformly over a k dimensional affine subspace of $\{0,1\}^n$. To extract from such sources, we use functions known as bent functions [Rot76, Dil74]. These are boolean functions which have maximum distance to any boolean affine function. They have previously been studied primarily for their cryptographic properties. In Section 7.3, we show that any bent function is also a $\frac{1}{2}2^{n/2-k}$-extractor for dimension k affine sources on $\{0,1\}^n$.

In Section 7.4, we show how to get more bits out. To do so, we use a generalization of bent functions to multiple output bits introduced by Nyberg [Nyb91], which he calls "perfect non-linear" functions. Such functions consist of m bent functions such that any non-zero linear combination of these functions is also bent. Because each of these bent functions is also an extractor, we get a construction of m functions such that any non-zero linear combination of these functions is a

$2^{n/2-k-1}$-extractor for dimension k affine sources on $\{0,1\}^n$. Thus the output of these functions on the source forms a $2^{n/2-k}$-biased space, and so using the relation between ε-bias and closeness in variation distance [Vaz86, AGHP92], we get that these perfect nonlinear functions are $2^{n/2-k+m/2}$-extractors with output length m for dimension k affine sources on $\{0,1\}^n$.

7.2 Preliminaries

Before we get to our results, we review some important definitions.

Definition 7.2.1. The *Fourier transform* of a boolean function $f : \{0,1\}^n \to \{0,1\}$ is given by

$$F(u) = \sum_{x \in \{0,1\}^n} (-1)^{f(x)+u \cdot x}.$$

Definition 7.2.2. A *bent function* is a boolean function $f : \{0,1\}^n \to \{0,1\}$ with $|F(u)| = 2^{n/2}$ for all $u \in \{0,1\}^n$.

An equivalent characterization of bent functions is that they are the set of boolean functions that have maximum distance to any affine function. That is, for any boolean affine function $u \cdot x + a$, $\Pr[f(x) = u \cdot x + a] = 1/2 \pm 2^{-n/2}$. This relationship between bent functions and affine functions is similar in spirit to our results.

In particular, Maiorana (unpublished, see [Dil74]) and McFarland [McF73] gave the following general method to construct bent functions.

Theorem 7.2.3. *Let $g : \{0,1\}^{n/2} \to \{0,1\}$ be any function and $\pi : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ be any permutation. Then the function $f : \{0,1\}^n = \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ defined as*

$$f(x_1, x_2) = \pi(x_1) \cdot x_2 + g(x_1)$$

is a bent function.

In particular, if π is the identity permuation and g is the zero constant function, then we see that the function $f(x_1, x_2) = x_1 \cdot x_2$ is an extremely simple example of a bent function.

We'll also need the definition of an ε-biased random variable, which has the property that every non-zero linear combination of its bits is close to uniform.

Definition 7.2.4. A random variable X over $\{0,1\}^m$ is ε-biased, if for all non-zero $a \in \{0,1\}^m$,

$$|\Pr[a \cdot X = 0] - \Pr[a \cdot X = 1]| \leq \varepsilon.$$

Equivalently, $a \cdot X$ is $\varepsilon/2$ close to uniform in variation distance.

Vazirani [Vaz86] showed that if X is ε-biased, then X is also close to uniform (see also [AGHP92]).

Theorem 7.2.5. *[Vaz86] If the random variable X over $\{0,1\}^m$ is ε-biased, then X is $2^{m/2}\varepsilon$ close to uniform in variation distance.*

7.3 Extracting A Single Bit

The following theorem states that any bent function is a single bit extractor for affine sources. In particular, any of the Maiorana-McFarland bent functions from Theorem 7.2.3 is an extractor.

Theorem 7.3.1. *Any bent function $f : \{0,1\}^n \to \{0,1\}$ is an ε-extractor for dimension k affine sources on $\{0,1\}^n$ for $\varepsilon = 2^{n/2-k-1}$.*

The proof is a straightforward application of the following well known lemma (see e.g. [Car02]).

Lemma 7.3.2. *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function and E be any vector subspace of $\{0,1\}^n$ and $a \in \{0,1\}^n$. Then*

$$\sum_{x \in a \oplus E} (-1)^{f(x)} = \frac{1}{|E^\perp|} \sum_{u \in E^\perp} F(u).$$

Proof. (Of Theorem 7.3.1.) Let $a \oplus E$ be the affine space corresponding to the affine source. Then the variation distance from uniform is given by

$$\begin{aligned}
|\Pr_{x \in a \oplus E}[f(x) = 0] - 1/2| &= \frac{1}{2} |\Pr_{x \in a \oplus E}[f(x) = 0] - \Pr_{x \in a \oplus E}[f(x) = 1]| \\
&= \frac{1}{2} \left| \frac{1}{|E|} \sum_{x \in a \oplus E} (-1)^{f(x)} \right| \\
&= \frac{1}{2|E||E^\perp|} \left| \sum_{u \in E^\perp} F(u) \right| \\
&\leq \frac{1}{2|E||E^\perp|} \sum_{u \in E^\perp} |F(u)| \\
&= \frac{|E^\perp| 2^{n/2}}{2|E||E^\perp|} = \frac{1}{2} 2^{n/2-k}
\end{aligned}$$

where Lemma 7.3.2 is used in the third line. □

7.4 Extracting Multiple Bits

Now we show how we can use the fact that any boolean bent function is an extractor to get an extractor that outputs many bits. We use the results of Nyberg [Nyb91] for constructing sets of bent functions such that any non-zero linear combination of these functions is also bent, which he calls "perfect nonlinear". Nyberg gives a few different constructions. The one that we give here is based on the Maiorana-McFarland method discussed in Section 7.2.

Theorem 7.4.1. *[Nyb91] Let $n > 0$ and $m \leq n/2$, and for $1 \leq i \leq m$ let $f_i : \{0,1\}^n \to \{0,1\}$ be defined as*

$$f_i(x) = f_i(x_1, x_2) = \pi_i(x_1) \cdot x_2 + g_i(x_1),$$

where π_i is a permutation of $\{0,1\}^{n/2}$ and g_i is a function from $\{0,1\}^{n/2}$ to $\{0,1\}$. Then $f : \{0,1\}^n \to \{0,1\}^m$ defined as

$$f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$

has the property that the function defined by taking any non-zero linear combination of its output bits is also bent if and only if every non-zero linear combination of the permutations π_i is a permutation of $\{0,1\}^{n/2}$. In particular, if for $1 \leq i \leq m$ $\pi_i(x_1) = \alpha^i x_1$, where α is a primitive element of $GF(2^{n/2})$, then f has the desired property. (Note that in π_i we implicitly map x_1 from $\{0,1\}^{n/2}$ to $GF(2^{n/2})$ so we can perform the multiplication by α^i in this field. Then we map the result back into $\{0,1\}^{n/2}$ so we can perform the dot product multiplication in f_i.)

Since any bent function is also an extractor for affine sources, we have the following theorem.

Theorem 7.4.2. *For $m \leq n/2$, we can construct a function $f : \{0,1\}^n \to \{0,1\}^m$ such that the function defined by taking any non-zero linear combination of its output bits is an ε-extractor for dimension k affine sources on $\{0,1\}^n$ for $\varepsilon = 2^{n/2-k-1}$. Thus the output of f forms a $2^{n/2-k}$-biased space.*

Proof. Let f be the function defined in Theorem 7.4.1. Since the function defined by taking any non-zero linear combination of the output bits of f is bent, by Theorem 7.3.1 this function is also an ε-extractor. □

Now using the fact that ε-bias implies closeness to uniform, we get the following theorem.

Theorem 7.4.3. *For $m \leq n/2$, we can construct a function $f : \{0,1\}^n \to \{0,1\}^m$ that is a $2^{n/2-k+m/2}$-extractor for dimension k affine sources on $\{0,1\}^n$.*

Proof. Let f be the function defined in Theorem 7.4.1 and X be a dimension k affine source on $\{0,1\}^n$. By Theorem 7.4.2, every non-zero linear combination of the output bits of $f(X)$ is ε-close to uniform, for $\varepsilon = 2^{n/2-k-1}$. Hence, $f(X)$ is 2ε-biased, so by Theorem 7.2.5, $f(X)$ is $2^{n/2-k+m/2}$-close to uniform. □

7.5 Subsequent Work and Open Questions

Recently, Bourgain [Bou07] has improved upon our results by giving a construction of extractors which work for k any constant fraction of n. However, as we saw in Theorem 3.1.3, this is still far from what can be achieved nonconstructively, where we can extract even when the entropy is logarithmic in n.

Besides improving these previous constructions, an interesting open area is in generalizing affine sources. Perhaps the most obvious generalization in light of our previous study of symbol sources is to have d-ary affine sources. In this case the source is uniformly distributed over an affine subspace of $\mathbb{Z}_d^n$ instead of $\mathbb{Z}_2^n$. Recently Gabizon and Raz [GR05] have constructed extractors which extract almost all of the entropy from such sources when each input symbol is taken from a large finite field. However, this still leaves a large gap of alphabet sizes for which we don't know how to extract. In particular, it would be interesting to construct extractors for small constant $d > 2$. Another generalization would be to have each bit be given by a low-degree multi-variate polynomial of the random bits instead of

an affine function. So far, no results are known for such sources, even for degree two.

Bibliography

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k–wise independent random variables. *Random Structures and Algorithms*, 3(3):289–303, 1992.

[AKS87] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic simulation in Logspace. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 132–140, 1987.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of Mathematics*, 160(2):781–793, 2004.

[AL93] M. Ajtai and N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[AS00] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley–Interscience Series, John Wiley & Sons, Inc., New York, 2000.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, April 1988.

[BGK06] J. Bourgain, A. Glibichuk, and S. Konyagin. Estimates for the number of sums and products and for exponential sums in fields of prime order. *J. London Math. Soc.*, 73(2):380–398, 2006.

[BIW04] Boaz Barak, Russell 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.

[BL90] B. Bollobás and I. Leader. Exact edge-isoperimetric inequalities. *Europ. J. Combinatorics*, 11:335–340, 1990.

[Bla96] Matt Blaze. High-bandwidth encryption with low-bandwidth smartcards. In *Fast Software Encryption*, pages 33–40, 1996.

[Blu86] M. Blum. Independent unbiased coin flips from a correlated biased source: a finite Markov chain. *Combinatorica*, 6(2):97–108, 1986.

[BOL90] M. Ben-Or and N. Linial. Collective coin flipping. In S. Micali, editor, *Randomness and Computation*, pages 91–115. Academic Press, New York, 1990.

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

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

[Boy99] V. Boyko. On the security properties of the oaep as an all-or-nothing transform. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 503–518. Springer-Verlag, 1999.

[BR90] B. Bollobás and A. J. Radcliffe. Isoperimetric inequalities for faces of the cube and the grid. *Europ. J. Combinatorics*, 11:323–333, 1990.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 276–287, 1994.

[BS00] Jürgen Bierbrauer and Holger Schellwat. Almost independent and weakly biased arrays: Efficient constructions and cryptologic applications. In *Advances in Cryptology — CRYPTO '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 531–543, 2000.

[Car02] C. Carlet. On cryptographic complexity of boolean functions. In G.L. Mullen, H. Stichtenoth, and H. Tapia Recillas, editors, *Proceedings of the Sixth Conference on Finite Fields with Applications to Coding Theory, Cryptography and Related Areas*, pages 53–69. Springer, 2002.

[CDH+00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469. Springer-Verlag, May 2000.

[CFG+85] B. Chor, J. Friedman, O. Goldreich, J. Håstad, S. Rudich, and R. Smolensky. The bit extraction problem or t–resilient functions. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

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

[Cra37] H. Cramer. On the order of magnitude of the difference between consecutive prime numbers. *Acta Arithmetica*, pages 23–46, 1937.

[CW79] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[CW89] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[Dia88] Persi Diaconis. *Group Representations in Probability and Statistics*, volume 11 of *Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1988.

[Dil74] J. F. Dillon. *Elementary Hadamard Difference Sets*. PhD thesis, University of Maryland, 1974.

[Dod00a] Yevgeniy Dodis. *Exposure-Resilient Cryptography*. PhD thesis, MIT, 2000.

[Dod00b] Yevgeniy Dodis. Impossibility of black-box reduction from non-adaptively to adaptively secure coin-flipping. Unpublished manuscript, April 2000.

[DR05] Z. Dvir and R. Raz. Analyzing linear mergers. Technical Report TR05-25, ECCC: Electronic Colloquium on Computational Complexity, 2005.

[DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT '01*, volume 2045 of *Lecture Notes in Computer Science*, pages 301–324. Springer-Verlag, May 2001.

[Fri92] J. Friedman. On the bit extraction problem. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 314–319, 1992.

[GG81] O. Gabber and Z. Galil. Explicit construction of linear sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[GR05] A. Gabizon and R. Raz. Deterministic extractors for affine sources over large fields. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 407–416, 2005.

[GRS04] A. Gabizon, R. Raz, and R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 394–403, 2004.

[IZ89] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989.

[JK99] B. Jun and P. Kocher. The intel random number generator, 1999. http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf.

[JSY99] Markus Jakobsson, Julien P. Stern, and Moti Yung. Scramble all, encrypt small. *Lecture Notes in Computer Science*, 1636:95–111, 1999.

[KJS01] Kaoru Kurosawa, Thomas Johansson, and Douglas R. Stinson. Almost k-wise independent sample spaces and their cryptologic applications. *Journal of Cryptology*, 14(4):231–253, 2001.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–80, 1988.

[KM04] Robert Koenig and Ueli Maurer. Extracting randomness from generalized symbol-fixing and markov sources. In *Proceedings of 2004 IEEE International Symposium on Information Theory*, page 232, June 2004.

[KM05] Robert Koenig and Ueli Maurer. Generalized strong extractors and deterministic privacy amplification. In Nigel Smart, editor, *Cryptography and Coding 2005*, volume 3796 of *Lecture Notes in Computer Science*, pages 322–339. Springer-Verlag, December 2005.

[KRVZ06] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small space sources. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 691–700, 2006.

[KZ06] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2006.

[LLS89] D. Lichtenstein, N. Linial, and M. Saks. Some extremal problems arising from discrete control processes. *Combinatorica*, 9(3):269–287, 1989.

[Lov96] L. Lovász. Random walks on graphs: A survey. In D. Miklós, V. T. Sós, and T. Szőnyi, editors, *Combinatorics, Paul Erdős is Eighty, Vol. 2*, pages 353–398. J. Bolyai Math. Soc., Budapest, 1996.

[LPS88] A. Lubotzky, R. Philips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[Lub94] A. Lubotzky. *Discrete Groups, Expanding Graphs and Invariant Measures*. Birkhäuser, 1994.

[McF73] R. L. McFarland. A family of difference sets in non-cyclic groups. *J. Combinatorial Theory, Ser. A*, 15:1–10, 1973.

[MPR] S. Matyas, M. Peyravian, and A. Roginsky. Encryption of long blocks using a short-block encryption procedure. http://grouper.ieee.org/groups/1363/P1363a/LongBlock.html.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer-Verlag, August 1997.

[NTS99] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58:148–173, 1999.

[Nyb91] K. Nyberg. Perfect non-linear s-boxes. In *Advances in Cryptology – Eurocrypt '91*, volume 547 of *Lecture Notes in Computer Science*, pages 378–386. Springer Verlag, 1991.

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

[Rao06] A. 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*, pages 497–506, 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.

[Riv97] Ronald L. Rivest. All-or-nothing encryption and the package transform. *Lecture Notes in Computer Science*, 1267:210–218, 1997.

[Rot76] O. S. Rothaus. On "bent" functions. *Journal of Combinatorial Theory, series A*, 20:300–305, 1976.

[RRV99] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 149–158, 1999.

[RRV02] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *Journal of Computer and System Sciences*, 65(1):97–128, 2002.

[RVW02] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag product, and new constant-degree expanders and extractors. *Annals of Mathematics*, 155(1):155–187, 2002.

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

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the European Association for Theoretical Computer Science*, (77):67–95, June 2002.

[Sha06] R. Shaltiel. How to get more mileage from randomness extractors. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, pages 49–60, 2006.

[SV86] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 276–285, 1996.

[TV00] Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 32–42, 2000.

[Vad04] S. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, Winter 2004.

[Vaz86] U. V. Vazirani. *Randomness, Adversaries and Computation*. PhD thesis, EECS, University of California at Berkeley, 1986.

[Vaz87] Umesh V. Vazirani. Strong communication complexity or generating quasirandom sequences from two communicating semirandom sources. *Combinatorica*, 7(4):375–392, 1987.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Mathematics Series*, 12:36–38, 1951.

[VV85] U. V. Vazirani and V. V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 417–428, 1985.

[Wor99] N. C. Wormald. *The differential equation method for random graph processes and greedy algorithms*, pages 73–155. PWN, Warsaw, 1999.

[WZ99] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

[Zuc96] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16:367–391, 1996.

[Zuc06] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 681–690, 2006.

Vita

Jesse John Kamp was born in Bartlesville, Oklahoma on January 9, 1979, the son of Jan Mary Kamp and William Paul Kamp. After completing his work at Apple Valley High School, Apple Valley, Minnesota, in 1997, he entered The Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts. He received the degree of Bachelor of Science from MIT in June 2001. In August 2001 he entered the Graduate School of The University of Texas at Austin.

Permanent Address: 12904 Hamlet Ave, Apple Valley, MN 55124

This dissertation was typeset with LaTeX 2ε[1] by the author.

[1] LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.