SIAM J. COMPUT.
Vol. 36, No. 5, pp. 1231–1247

DETERMINISTIC EXTRACTORS FOR BIT-FIXING SOURCES AND EXPOSURE-RESILIENT CRYPTOGRAPHY*

JESSE KAMP† AND DAVID ZUCKERMAN‡

Abstract. We give an efficient deterministic algorithm that extracts $\Omega(n^{2\gamma})$ almost-random bits from sources where $n^{\frac{1}{2}+\gamma}$ of the n bits are uniformly random and the rest are fixed in advance. This improves upon previous constructions, which required that at least $n/2$ of the bits be random in order to extract many bits. Our construction also has applications in exposure-resilient cryptography, giving explicit adaptive exposure-resilient functions and, in turn, adaptive all-or-nothing transforms. For sources where instead of bits the values are chosen from $[d]$, for $d > 2$, we give an algorithm that extracts a constant fraction of the randomness. We also give bounds on extracting randomness for sources where the fixed bits can depend on the random bits.

Key words. extractors, randomness, deterministic, bit-fixing sources, exposure-resilient, cryptography, resilient function, random walks

AMS subject classifications. 68Q10, 94A60, 68W20

DOI. 10.1137/S0097539705446846

1. Introduction.

True randomness is needed for many applications, such as cryptography. However, most physical sources of randomness are not even close to being truly random, and may in fact seem quite weak in that they can have substantial biases and correlations. A natural approach to dealing with the problem of weak physical sources is to apply a *randomness extractor*—a function that transforms a weak random source into an almost uniformly random source. For certain natural notions of such random sources, it has been shown that it is impossible to devise a single function that extracts even one bit of randomness [32]. One way to combat this problem is to allow the use of a small number of uniformly random bits as a catalyst in addition to the bits from the weak random source. Objects constructed in this manner, known as seeded extractors [28], have been shown to extract almost all of the randomness from general weak random sources (see [33] for a recent survey).

However, we would like to eliminate the need for the random catalyst by restricting the class of weak random sources for which we need our function to work. Following the lead of Trevisan and Vadhan [34], we call such functions deterministic extractors for the given class of sources. More formally, we say that a function is an ϵ-extractor for a class of sources if the output of the function is ϵ-close to uniform (in variation distance) for all sources in the class.

1.1. Bit-fixing and symbol-fixing sources.

The particular class of sources that we are interested in are bit-fixing sources, in which some subset of the bits are fixed and the rest are chosen at random. There are two classes of bit-fixing sources,

*Received by the editors January 19, 2005; accepted for publication (in revised form) March 12, 2006; published electronically December 21, 2006. A preliminary version of this paper has appeared in *IEEE Symposium on Foundations of Computer Science*, 2003, pp. 92–101.
http://www.siam.org/journals/sicomp/36-5/44684.html

†Department of Computer Science, University of Texas, Austin, TX 78712 (kamp@cs.utexas.edu).
The research of this author was supported in part by NSF grants CCR-9912428 and CCR-0310960.

‡Department of Computer Science, University of Texas, Austin, TX 78712 (diz@cs.utexas.edu).
The research of this author was supported in part by a David and Lucile Packard Fellowship for Science and Engineering, NSF grants CCR-9912428 and CCR-0310960, a Radcliffe Institute Fellowship, and a Guggenheim Fellowship.

depending on whether the fixed bits are chosen before or after the random bits are determined, known respectively as oblivious and nonoblivious bit-fixing sources. We will construct extractors for both classes.

Extractors for oblivious bit-fixing sources were first studied in [11], in which they considered the case of exactly uniform output. They proved that at least $n/3$ random bits are needed to extract even two bits from an input of length n. Friedman generalized this result to obtain bounds on the number of random bits needed for longer outputs [16]. The large amount of randomness needed to obtain exactly uniform resilient functions led to the consideration of relaxing this restriction to allow for almost uniform output. We note that even when we allow the extractor to have small error, the best previous constructions still required that at least half of the bits be random [23, 4].

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 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 $\epsilon = 2^{-cm}$. 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 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 $\epsilon = \frac{1}{2}k^{\frac{1}{4}}\exp(-\frac{\pi^2\sqrt{k}}{2})$. This extractor is computable in a linear number of arithmetic operations on $\frac{1}{4}\log k$ bits.*

In addition to studying oblivious bit-fixing sources, we introduce the related model of d-ary oblivious symbol-fixing sources (SF sources). Such a source consists of a string of symbols over a d symbol alphabet where k of the symbols are random and 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 SF sources with $d > 2$, we get much better results than for oblivious bit-fixing sources. We extract 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 we can extract almost all of the randomness.

THEOREM 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 SF sources with k random symbols that outputs $m = c_d\ k - O(\log_d(1/\epsilon))$ 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.*

Another interesting related class of sources for which deterministic extraction is possible are nonoblivious bit-fixing sources [3, 21]. In such sources, the fixed bits can depend on the random bits chosen. This problem was originally studied in the context of collective coin flipping [3], which can be viewed as extraction of a single bit. For the single bit case, nearly optimal lower [21] and upper [2] bounds are known, though the upper bound is not completely constructive. However, little attention has previously been given to generalizing these results to the case of multiple output bits. We give bounds for this case. If $\ell = n - k$ is the number of fixed bits in the source, we show that at most n/ℓ bits can be extracted from these sources, which is likely to be nearly optimal. We also give a construction of an ϵ-extractor for nonoblivious

bit-fixing sources which outputs $\Omega((\epsilon/\ell)^{\log_2 3} \cdot n)$ bits.

1.2. Exposure-resilient cryptography. Our work has applications in 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), introduced by Rivest [30], address this issue. 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 [10], and increasing the efficiency of block ciphers [27, 20, 5].

Boyko used the random-oracle model to give the first formalizations and constructions of the AONT [9]. Canetti et al. gave the first constructions in the standard computational model [10]. 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, Sahai, and Smith [15] 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 [10] and [15] is that we can 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 [15] gives a probabilistic construction of adaptive statistical ERF's, the problem of giving an explicit construction was left open (see also [14]).

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 in [23]. 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 [15], 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.

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

Our extractor for d-ary SF sources involves using the input symbols to take a random walk on a d-regular expander graph, starting from an arbitrary 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. Because of the rapid mixing properties of expanders, this output will be close to uniform, and we can extract a linear fraction of the entropy, thus proving Theorem 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, proving Theorem 1.2; we give these constructions in section 3.1. We also note that similar types of random walks have been used in previous pseudorandomness constructions [1, 12, 19].

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 SF sources, which we call approximate symbol-fixing (approx-SF) sources, and then applying the expander walk extractor. This gives the extractor from Theorem 1.1. We show in section 3.2 that our extractor for SF sources also works for approx-SF sources. To convert the oblivious bit-fixing source into a d-ary approx-SF 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 1.1. In section 4, we show how to do the conversion and prove that the extractor works.

In section 5, we show the relation between our extractors for oblivious bit-fixing sources and exposure-resilient cryptography.

We give our results for nonoblivious bit-fixing sources in section 6. For such sources, let $\ell = n - k$ be the number of fixed bits. We show that at most n/ℓ bits can be extracted from these sources using a generalization of the edge isoperimetric inequality on the cube. This is likely to be nearly optimal, as it almost corresponds to applying known single bit functions to blocks of the input. In particular, we can use any function with low "influence" [3]. Our best explicit construction uses the iterated majority function of Ben-Or and Linial [3] and outputs $\Omega((\epsilon/\ell)^{\log_2 3} \cdot n)$ bits. However, there are nonexplicit constructions that give bounds within a polylogarithmic factor of our edge isoperimetric bound [2].

1.4. Subsequent work. Since this paper first appeared, Gabizon, Raz, and Shaltiel [18] have improved upon our constructions of extractors for oblivious bit-fixing sources. Using our extractors as building blocks, they are able to extract almost all of the randomness from oblivious bit-fixing sources. Unfortunately, however, the error they achieve is not good enough for our application of constructing adaptive ERF's.

2. Preliminaries. For ease of notation, we sometimes assign noninteger 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 for f to represent a family of functions over all input lengths, so asymptotic notions make sense.

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. Extractor definitions. Trevisan and Vadhan studied what would happen if you removed the random catalyst from ordinary extractors, and they called such functions deterministic extractors [34]. Deterministic extractors for general weak sources are impossible, and they're even impossible for semirandom sources [32]. However, if we restrict our attention to certain classes of weak sources, then the problem becomes tractable. The following definition of a deterministic extractor is taken from [14], which is implicit in the definitions of [34].

DEFINITION 2.1. *An efficiently computable function $f : \{0,1\}^n \to \{0,1\}^m$ is an ϵ-extractor for a set of random sources $\mathcal{X}$, if for every $X \in \mathcal{X}$, $f(X)$ is within variation distance ϵ of uniform.*

The sets of sources we use are the sets of oblivious bit-fixing [11], symbol-fixing, and nonoblivious bit-fixing sources [3]. Oblivious bit-fixing sources are the easiest to handle, since the fixed bits do not depend on the random bits.

DEFINITION 2.2 (see [11]). *An (n, k) oblivious bit-fixing source X is a source with n bits, of which all but k are fixed and the rest are then chosen uniformly at random.*

DEFINITION 2.3. *An (n, k, d) oblivious SF source X 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 $d = 2^t$, SF 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.

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

DEFINITION 2.4 (see [3]). *An (n, k) nonoblivious bit-fixing source X is a source with n bits, of which k are chosen uniformly at random and then the remaining $n - k$ bits are chosen, possibly depending on the random bits.*

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

DEFINITION 2.5. *An (n, k, d, ϵ) approximate oblivious symbol-fixing (approx-SF) source X is a source with n symbols independently chosen from $[d]$, of which k have distributions within an ℓ_2 distance of ϵ of uniform.*

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 nonzero 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 will look at, 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 2.6. *A family of expander graphs is an infinite set of regular graphs G with uniform transition matrix P that have $\lambda(P) = 1 - \epsilon$ for some constant $\epsilon > 0$.*

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., [17, 25, 26, 29]. 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.

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

3.1. Extracting from SF sources. In this section, we prove the following generalization of Theorem 1.3 to show that we can extract a constant fraction of the randomness from SF sources.

THEOREM 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 $m \leq 2\alpha k - \frac{2}{\log d} \log \frac{1}{2\epsilon}$, then there exists an efficiently computable ϵ-extractor for the set of (n, k, d) SF sources which outputs m 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 [26, 25]. 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 3.2. *For odd d, there exists an ϵ-extractor $f : \{0,1\}^n \to [d]$, for the set of (n, k) oblivious bit-fixing sources, where $\epsilon = \frac{1}{2}\sqrt{d}\exp(-\frac{\pi^2 k}{2d^2})$. 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 1.2.

Both Theorems 3.1 and 3.2 arise from the following key lemma.

LEMMA 3.3. *Let* P *be a uniform transition matrix with stationary distribution* π *for an undirected nonbipartite* d*-regular graph* G *on* M *vertices. Consider an* n *step walk on* G*, with the steps taken according to the symbols from an* (n, k, d) *SF source* X*. For any initial probability distribution* $p = v + \pi$*, the distance from uniform at the end of the walk is bounded by*

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

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 [24].

LEMMA 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 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 3.3.

Proof of Lemma 3.3. For the random symbols we can apply Lemma 3.4. Since there are k random symbols, this gives us the $\lambda(P)^k$ factor. We also use that by Lemma 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 3.3 to prove Theorem 3.1.

Proof of Theorem 3.1. We can apply Lemma 3.3, where in this case $\lambda(P) \leq d^{-\alpha}$ and $M = d^m$. Thus the error $\epsilon \leq \frac{1}{2} d^{-\alpha k + (m/2)}$. Taking logarithms and solving for m, we get the stated bound on m. □

Now, using Lemma 3.3, we can prove Theorem 3.2. We first separate out the following lemma which will be useful later.

LEMMA 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* (n, k) *oblivious bit fixing source* X*. For any initial probability distribution* $p = v + \pi$*, the distance from uniform at the end of the walk is bounded by*

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

Proof. The lemma follows from Lemma 3.3 and the fact that the d-cycle has $\lambda(P) = \cos(\pi/d)$ (see [13]). □

Proof of Theorem 3.2. The extractor outputs the result of a random walk on the d-cycle. By Lemma 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 [13, 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 3.7. *If a random variable X is within ϵ of uniform over $[N]$, then the random variable $Y = X \mod M$ is within $\epsilon + 1/r$ 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 $M/N \leq 1/r$. □

3.2. Extracting from approx-SF sources. We now show how the previous construction can be extended to handle the case of approx-SF sources. Our main result in this section is the following variant of Lemma 3.3 for approx-SF sources.

LEMMA 3.8. *Let P be a uniform transition matrix with stationary distribution π for an undirected nonbipartite d-regular graph G on M vertices. Suppose we take a walk on G for n steps, with the steps taken according to the symbols from an (n, k, d, ϵ) approx-SF source X. For any initial probability distribution $p = v + \pi$, the distance from uniform at the end of the walk is bounded by*

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

In the case of approx-SF 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 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 nonuniform random steps, we can modify the bound from Lemma 3.4 slightly to get the following lemma.

LEMMA 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 $p = v + \pi$,*

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

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

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

where the first line is simply from the definition, and noting that we only need to sum over all nonzero 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 nonzero 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 SF sources, the nonrandom 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 nonrandom 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 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 3.8.

Proof. Let P_i be the transition matrix of the random walk at the ith step. By Lemma 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 3.10, $\|v_{j-1}P_j\| \leq (\lambda(P) + \epsilon\sqrt{d})\|v_{j-1}\|$. For the other steps, we still have by Lemma 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

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

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

4. From SF sources to oblivious bit-fixing sources. In this section, we show how to extend our results for SF sources to oblivious bit-fixing sources to get the following theorem, which is basically a restatement of Theorem 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 4.1. *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 $(n, \delta n)$ oblivious bit-fixing sources, where $m = \Omega(\delta^2 n)$ and $\epsilon = 2^{-cm}$. 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 approx-SF 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 approx-SF source is then the concatenation of these outputs. Then we use the expander walk extractor from the previous section to extract from the approx-SF source.

We start by applying Lemma 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 approx-SF source.

LEMMA 4.2. *For any odd d, any $(n, \delta n)$ oblivious bit-fixing source can be deterministically converted into a $(\frac{\delta n}{2t}, \frac{\delta^2 n}{4t}, d, \epsilon)$ approx-SF source, where $t = \lceil \frac{\log \epsilon}{\log(\cos(\pi/d))} \rceil$.*

The almost random symbols in the approx-SF 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 4.3. *Suppose we have n bits from an (n, k) oblivious bit-fixing source, where $k = \delta n$. 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 4.2.

Proof of Lemma 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 approx-SF 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 3.6, if there are at least t random bits in the block the ℓ_2 distance from uniform is at most $(\cos(\pi/d))^t \leq \epsilon$, which means all such blocks are good. Then by Lemma 4.3, the number of good blocks r satisfies $r > \frac{\delta^2 n}{4t}$. Thus the output source is an approx-SF source with the appropriate parameters. ∎

The symbols from the approx-SF source then correspond to our almost random steps in the expander graph, so we can apply Lemma 3.8 to the expander walk to get

that the final distribution is close to uniform.

Proof of Theorem 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 $\lambda_{\epsilon'} = \lambda(P) + \epsilon'\sqrt{d} < 1$. Then use the procedure in Lemma 4.2 to convert the $(n, \delta n)$ oblivious bit-fixing source to a $(\frac{\delta n}{2t}, \frac{\delta^2 n}{4t}, d, \epsilon')$ approx-SF source, where $t = \lceil (\log \epsilon')/(\log(\cos(\pi/d))) \rceil$.

Now we use the approx-SF 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 3.8, which states that the variation distance from uniform of $f(r)$ is at most

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

We want this to be at most $\epsilon = 2^{-cm}$, so setting $m = b\delta^2 n$ for some constant $b > 0$ and taking the logarithm, we get $\frac{1}{4t}\log\frac{1}{\lambda_{\epsilon'}} \geq b\left(c + \frac{1}{2}\right)$. 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 3.8 is tighter than the bound on the variation distance, but this difference only affects the constants in the theorem.

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 [15], 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 $f : \{0,1\}^n \to \{0,1\}^m$. 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 $\epsilon(n)$ is said to be *negligible* if $\epsilon(n) = O(\frac{1}{n^c})$ 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 5.1 (see [15]). *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, $|\Pr[\mathcal{A}^r(f(r)) = 1] - \Pr[\mathcal{A}^r(R) = 1]| \leq \epsilon(n)$ for some negligible function $\epsilon(n)$.*

Our goal is to construct adaptive ERF's. We might first think that any $\epsilon(n)$-extractor for oblivious bit-fixing sources would work as long as $\epsilon(n)$ is negligible. However, [15] 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 the APRFs (first stated in section 1.2), introduced by Kurosawa, Johansson, and Stinson [23].

DEFINITION 5.2 (see [23]). *A $k = k(n)$ 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 $|p_i - 2^{-m}| < 2^{-m}\epsilon(n)$ for all i and for some negligible function $\epsilon(n)$.

THEOREM 5.3 (see [15]). *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 5.4. *Any $2^{-m}\epsilon(n)$-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of (n,k) oblivious bit-fixing sources, where $\epsilon(n)$ 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 $2^{-m}\epsilon(n)$. Thus the distance of any possible output from uniform must also be within $2^{-m}\epsilon(n)$, and the APRF property is satisfied. ∎

Now using this lemma we get the following theorem.

THEOREM 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 5.4 to the extractor from Theorem 4.1, choosing $c > 1$. ∎

We can use adaptive ERFs to construct AONTs, which were introduced by Rivest [30] and extended to adaptive adversaries by Dodis, Sahai, and Smith [15]. 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 5.6 (see [15]). *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 $\epsilon(s+p)$:*

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

The following lemma from [15] relates adaptive k-ERF's to adaptive k-AONT's, and shows that our construction gives adaptive k-AONT's.

THEOREM 5.7 (see [15]). *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 $x \oplus f(r)$.*

By combining Theorem 5.7 with Theorem 5.5, we get the following theorem.

THEOREM 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 $x \oplus f(r)$.*

6. Extracting from nonoblivious bit-fixing sources. In this section, we switch our focus to nonoblivious 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 nonoblivious bit-fixing sources have been defined in terms of the "influence" of variables on a function [3]. 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. The following two lemmas show that the influence of a function is related to the variation distance of the function from uniform when the input comes from a nonoblivious bit-fixing source. The first lemma shows that having low influence for all sets of a given size implies that a function is an extractor, while the second lemma shows that a function that has a set with high influence cannot be an extractor.

LEMMA 6.1. *Suppose $f : \{0,1\}^n \to \{0,1\}^m$ maps the uniform distribution U_n to U_m and $I_f(S) \leq \epsilon$ for all sets S of ℓ variables. Then f is an ϵ-extractor for the set of $(n, n-\ell)$ nonoblivious bit-fixing sources.*

Proof. Let X be an $(n, n-\ell)$ nonoblivious bit-fixing source and let S denote the set of fixed variables of X. Since $I_f(S) \leq \epsilon$, 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. □

LEMMA 6.2. *Let S be a set of ℓ variables. If, for some $\epsilon > 0$, $I_f(S) = \epsilon$, then there exists an $(n, n-\ell)$ nonoblivious bit-fixing source X with set of fixed variables S so that*

$$|f(X) - U_m| \geq \epsilon/4.$$

Proof. View the possible outputs as vertices of a hypergraph on 2^m vertices. Look at all possible values of the $n-\ell$ bits not in S. Since $I_f(S) = \epsilon$, 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 $(n, n-\ell)$ nonoblivious 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 $\epsilon/2$, these sources will differ by at least $\epsilon/2$. Thus at least one of them will differ by at least $\epsilon/4$ from the uniform distribution. □

Using Lemma 6.1, we immediately see that known constructions of Boolean functions with low influence [3, 2] 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 6.3. *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 $\ell(s,\epsilon) = \epsilon r(s)$ variables has $I_g(S) \leq \epsilon$ for all $\epsilon > 0$. Then there exists an ϵ-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of $(n, n-\ell(s,\epsilon))$ nonoblivious bit-fixing sources that extracts $m = n/s$ 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 6.1 we need to show that f has $I_f(S) \leq \epsilon$ for all sets S of $\ell = \ell(s,\epsilon)$ variables. Let ℓ_i be the number of bits in S in block i and set $\epsilon_i = \ell_i / r(s)$. 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 $\sum_{i=1}^{m} \epsilon_i = (\sum_{i=1}^{m} \ell_i)/r(s) = \ell/r(s) = \epsilon$, $I_f(S) \leq \epsilon$. □

We can apply this lemma to the iterated majority function of Ben-Or and Linial [3] to get an explicit extractor for nonoblivious bit-fixing sources.

THEOREM 6.4 (see [3]). *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 $\ell(s,\epsilon) = \epsilon\left(\frac{s}{3}\right)^{\alpha}$ variables has $I_g(S) \leq \epsilon$ for every $\epsilon > 0$, where $\alpha = \log_3 2$.*

THEOREM 6.5. *For every n, we can construct an ϵ-extractor $f : \{0,1\}^n \to \{0,1\}^m$ for the set of $(n, n-\ell)$ nonoblivious bit-fixing sources that extracts $m = \frac{1}{3}(\epsilon/\ell)^{1/\alpha} n$ bits, where $\alpha = \log_3 2$.*

Proof. Apply Lemma 6.3 using the function from Theorem 6.4. □

Ajtai and Linial [2] give hope for improvement since their functions allow $\Omega(s/\log^2 s)$ fixed bits. However, their construction is nonexplicit, and a bound like that in Lemma 6.3 is only known to hold for $\epsilon \geq 1/\operatorname{polylog}(s)$ [31].

In the other direction, we now show that at most n/ℓ bits can be extracted from nonoblivious bit-fixing sources. To do so, we generalize the edge-isoperimetric bound from [3].

LEMMA 6.6. *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*

$$1 - 2\frac{\binom{n-m+1}{l}}{\binom{n}{l}} - 2\epsilon.$$

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 $t \leq \epsilon 2^m$. Since the distance from uniform is at most ϵ, the total number of vertices with large colors is at most $\epsilon 2^n + t2^{n-m} \leq 2\epsilon 2^n$. 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 $n-m+1$ [7, 6]. 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 $1 - 2\frac{\binom{n-m+1}{l}}{\binom{n}{l}} - 2\epsilon$, 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 Lemmas 6.6 and 6.2, 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 6.7. *No function $f : \{0,1\}^n \to \{0,1\}^m$ is an ϵ-extractor for $(n, n-\ell)$ nonoblivious bit-fixing sources for any $\epsilon \leq \frac{1}{10}\min\{\frac{\ell\cdot(m-1)}{n}, 1\}$.*

Proof. Suppose f is an ϵ-extractor. First note that f must be within ϵ of uniform on uniform input. So by Lemma 6.6, there is a set of variables S of size ℓ with

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

By Lemma 6.2, there is an $(n, n-\ell)$ nonoblivious bit-fixing source X so that $f(X)$ is of distance at least $I_f(S)/4$ from uniform, so $\epsilon > I_f(S)/4$. Thus $\epsilon > (1 - e^{-\ell\cdot(m-1)/n})/6$. If $\ell \cdot (m-1)/n \geq 1$, then $\epsilon > (1-e^{-1})/6 > 1/10$. 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 $\epsilon > \frac{(1-e^{-1})}{6}\frac{\ell\cdot(m-1)}{n} > \frac{1}{10}\frac{\ell\cdot(m-1)}{n}$. ∎

7. Open questions. There remains some work to be done in order to get truly optimal deterministic extractors for oblivious bit-fixing sources. Though we can get nearly optimal results for the d-ary case, for $d > 2$, we lose a factor of δ in the binary case because of the need to take the random walks on the cycle. Ideally, we would like to improve the output length from $\Omega(\delta^2 n)$ to $\Omega(\delta n)$, to match the number of random bits in the input. The extractor of [18] is able to extract almost all of the randomness; however, the error is not as good. In particular, their extractor is not useful for the application to exposure-resilient cryptography. Can we construct an extractor that extracts a linear fraction of the randomness and has small error?

For nonoblivious bit-fixing sources, there also remains more work to be done. It would be nice to eliminate some of the difference between the lower and upper bounds. For the single bit case, Kahn, Kalai, and Linial [21] give a lower bound that improves upon the edge isoperimetric bound by a factor of $\log n$ using a harmonic analysis argument. Perhaps similar techniques could be applied to the general case of many output bits. Also, we could get better extractors if we could modify the construction of Ajtai and Linial [2] to work for smaller error and make it explicit.

Another interesting future direction would be to identify additional classes of sources that have deterministic extractors. One interesting possibility is the set of affine sources, where k bits are chosen uniformly at random and the n bits of the source are affine combinations of these bits. Affine sources are a special case of nonoblivious bit-fixing sources, so our constructions apply to affine sources as well. Other methods allow us to extract when $k > n/2$, but it would be interesting to construct extractors for affine sources that work for $k \leq n/2$. Recently, Bourgain [8] has overcome this barrier by constructing extractors that work for affine sources with $k = \delta n$ for any constant δ. However, there is still room for improvement,

since probabilistic arguments show that affine source extractors exist even when k is logarithmic in n.

Another interesting model is sources generated using a small amount of space. Recently, in joint work with Rao and Vadhan [22], we have given the first explicit constructions of deterministic extractors for such sources.

8. Acknowledgments. We thank Peter Bro Miltersen for suggesting the problem of extractors for oblivious bit-fixing sources and Anindya Patthak and Vladimir Trifonov for helpful discussions.

REFERENCES

[1] M. Ajtai, J. Komlós, and E. Szemerédi, *Deterministic simulation in Logspace*, in the 19th ACM Symposium on Theory of Computing, New York, NY, 1987, pp. 132–140.

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

[3] M. Ben-Or and N. Linial, *Collective coin flipping*, in Randomness and Computation, S. Micali, ed., Academic Press, New York, 1990, pp. 91–115.

[4] J Bierbrauer and H. Schellwat, *Almost independent and weakly biased arrays: Efficient constructions and cryptologic applications*, in Advances in Cryptology—CRYPTO 2000, Lecture Notes in Comput. Sci. 1880, Springer-Verlag, Berlin, 2000, pp. 531–543.

[5] M. Blaze, *High-bandwidth encryption with low-bandwidth smartcards*, in Fast Software Encryption, Third International Workshop, Cambridge, UK, Lecture Notes in Comput. Sci 1039, Springer-Verlag, Berlin, 1996, pp. 33–40.

[6] B. Bollobás and I. Leader, *Exact edge-isoperimetric inequalities*, European J. Combin., 11 (1990), pp. 335–340.

[7] B. Bollobás and A. J. Radcliffe, *Isoperimetric inequalities for faces of the cube and the grid*, European J. Combin., 11 (1990), pp. 323–333.

[8] J. Bourgain, *On the Construction of Affine Extractors*, Geom. Funct. Anal., to appear.

[9] V. Boyko, *On the security properties of the oaep as an all-or-nothing transform*, in Advances in Cryptology—CRYPTO 1999, M. Wiener, ed., Lecture Notes in Comput. Sci. 1666, Springer-Verlag, Berlin, 1999, pp. 503–518.

[10] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai, *Exposure-resilient functions and all-or-nothing transforms*, in Advances in Cryptology—EUROCRYPT 2000, B. Preneel, ed., Lecture Notes in Comput. Sci. 1807, Springer-Verlag, Berlin, 2000, pp. 453–469.

[11] B. Chor, J. Friedman, O. Goldreich, J. Håstad, S. Rudich, and R. Smolensky, *The bit extraction problem or t–resilient functions*, in 26th Annual Symposium on Foundations of Computer Science, Portland, OR, 1985, pp. 396–407.

[12] A. Cohen and A. Wigderson, *Dispersers, deterministic amplification, and weak random sources*, in 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 14–19.

[13] P. Diaconis, *Group Representations in Probability and Statistics*, Lecture Notes—Monograph Series 11, Institute of Mathematical Statistics, Hayward, CA, 1988.

[14] Y. Dodis, *Exposure-Resilient Cryptography*, Ph.D. thesis, MIT, Cambridge, MA, 2000.

[15] Y. Dodis, A. Sahai, and A. Smith, *On perfect and adaptive security in exposure-resilient cryptography*, in Advances in Cryptology—EUROCRYPT 2001, Birgit Pfitzmann, ed., Lecture Notes in Computer Sci. 2045, Springer-Verlag, 2001, pp. 301–324.

[16] J. Friedman, *On the bit extraction problem*, in 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 314–319.

[17] O. Gabber and Z. Galil, *Explicit construction of linear sized superconcentrators*, J. Comput. System Sci., 22 (1981), pp. 407–420.

[18] A. Gabizon, R. Raz, and R. Shaltiel, *Deterministic extractors for bit-fixing sources by obtaining an independent seed*, in 45th Annual Symposium on Foundations of Computer Science, Rome, Italy, 2004, pp. 394–403.

[19] R. Impagliazzo and D. Zuckerman, *How to recycle random bits*, in the 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 248–253.

[20] M. Jakobsson, J. P. Stern, and M. Yung, *Scramble all, encrypt small*, Lecture Notes in Comput. Sci. 1636 (1999), pp. 95–111.

[21] J. Kahn, G. Kalai, and N. Linial, *The influence of variables on Boolean functions*, in the 29th Annual Symposium on Foundations of Computer Science, White Plains, NY, 1988,

pp. 68–80.

[22] J. Kamp, A. Rao, S. Vadhan, and D. Zuckerman, *Deterministic extractors for small space sources*, in the 38th ACM Symposium on Theory of Computing, Seattle, WA, 2006, pp. 691–700.

[23] K. Kurosawa, T. Johansson, and D. R. Stinson, *Almost k-wise independent sample spaces and their cryptologic applications*, J. Cryptology, 14 (2001), pp. 231–253.

[24] L. Lovász, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdős is Eighty, Vol. 2, D. Miklós, V. T. Sós, and T. Szőnyi, eds., János Bolyai Math. Soc., Budapest, 1996, pp. 353–398.

[25] A. Lubotzky, *Discrete Groups, Expanding Graphs and Invariant Measures*, Birkhäuser-Verlag, Basel, Switzerland, 1994.

[26] A. Lubotzky, R. Philips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–277.

[27] 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.

[28] N. Nisan and D. Zuckerman, *Randomness is linear in space*, J. Comput. System Sci., 52 (1996), pp. 43–52.

[29] O. Reingold, S. Vadhan, and A. Wigderson, *Entropy waves, the zig-zag product, and new constant-degree expanders and extractors*, Ann. of Math. (2), 155 (2002), pp. 155–187.

[30] R. L. Rivest, *All-or-nothing encryption and the package transform*, Lecture Notes in Comput. Sci., 1267 (1997), pp. 210–218.

[31] A. Russell and D. Zuckerman, *Perfect-information leader election in* $\log^* n + O(1)$ *rounds*, J. Comput. System Sci., 63 (2001), pp. 612–626.

[32] M. Santha and U. V. Vazirani, *Generating quasi-random sequences from semi-random sources*, J. Comput. System Sci., 33 (1986), pp. 75–87.

[33] R. Shaltiel, *Recent developments in explicit constructions of extractors*, Bull. Eu. Assoc. Theor. Comput. Sci., (2002), pp. 67–95.

[34] L. Trevisan and S. P. Vadhan, *Extracting randomness from samplable distributions*, in the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, IEEE Comput. Soc. Press, Los Alamitos, CA, 2000, pp. 32–42.