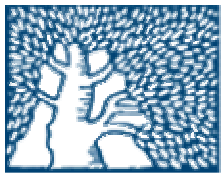

מכון ויצמן למדע

Weizmann Institute of Science

Thesis for the degree
Master of Science

חיבור לשם קבלת התואר
מוסמך למדעים

By
Ariel Elbaz

מאת
אריאל אלבז

Improved Constructions for Extracting Quasi-Random Bits from Sources of Weak Randomness

בניות משופרות של פונקציות למיצוי ביטים קוואזי-אקראיים ממקורות אקראיות חלשה

Supervisor: **Prof. Ran Raz**
Department of Computer Science and Applied Mathematics

מנחה: **פרופ' רן רז**
המחלקה למדעי המחשב ולמתמטיקה שימושית

August, 2003

אב, תשס"ג

Submitted to the Scientific Council of the
Weizmann Institute of Science
Rehovot, Israel

מוגש למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

Acknowledgements

I am thankful to my advisor, Ran Raz, for introducing me to this subject.

I am also thankful to Amir Shpilka, for many talks we had in the early stages of this thesis, and to Adi Avidor, who commented on an early version of this thesis.

I am grateful to Yuval Filmus, Eden Clamtac, Eran Ofek, Udi Wieder, and especially to Eran Tromer, for many interesting discussions.

I also want to thank Ronen Shaltiel for suggesting many modifications that improved the readability of this thesis.

Above all I want to thank my family, Yael, and my friends, for their support throughout my studies.

Abstract

We present an improved technique for extracting quasi-random bits from two sources of weak randomness. Given random variables X, Y over $\{0,1\}^n$, we denote the min-entropy of X by b_X and the min-entropy of Y by b_Y. We show how to extract $b_X + \Omega(b_X + b_Y - n)$ quasi-random bits, when $b_X + b_Y$ is at least $n + \Omega(polylog(n))$. This is the first explicit construction that extracts $\Omega(n)$ quasi-random bits even when $b_X + b_Y = n + o(n)$.

For the proof, we show that the extraction method of Vazirani ([Vaz87b]) extracts $\Omega(b_X + b_Y - n)$ bits that are almost independent of X. We use these bits as a seed for an extractor and extract all the randomness from X.

Contents

1 Introduction

1.1 Randomness in Computer Science

Randomness is a central tool in computer science. For many problems, randomized algorithms give more efficient or simpler solutions than the best known deterministic algorithms for these problems (for examples, see [AS91] and the references therein).

Randomized algorithms and protocols assume an auxiliary source of truly random input bits, i.e. independent and unbiased bits.

There are physical phenomena that are known to be random. Sampling a random physical phenomenon can give a random source. However, such sources are usually biased, and may have dependencies between the sampled bits. Furthermore, the bias of such sources increases with the sampling rate (see [Mur70], and the discussion in [Rit]).

Denote by U_n the uniform distribution on $\{0,1\}^n$. For a randomized algorithm that uses n random bits, we assume they are coming from U_n. If the bits are coming from a distribution D_n, then the bias of D_n (defined as half the statistical distance between D_n from U_n) may add up to the error probability of the algorithm. Sources with super-polynomially small (in n) bias are called *quasi-random sources*. Quasi-random sources are statistically indistinguishable from truly random sources, and can be used instead (see also the discussion at [Gol02], pages 70-71). On the other hand, sources with distributions that are "far" from uniform are called *weak random sources*.

1.2 Extracting Randomness from Imperfect Sources

Several classes of weak random sources were defined and studied in the past. The motivation for studying such sources is to find a general characterization of weak random sources, and show that such sources can be used efficiently in randomized algorithms.

One way to use a weak random source in randomized algorithms is by transforming the weak random source into a strong source (e.g., quasi-random source). This problem has been studied extensively in the last decade, although some works date as early as 1951 [vN51].

We view a random source as a random variable over $\{0,1\}^n$. An extraction function is a function that transforms a weak random source (or possibly more than one random source) into a stronger random source. To be able to extract randomness from a source, we first have to define more clearly the randomness that is present in the source.

Consider, for example, von Neumann's work ([vN51]). The class of random sources he studied was those sources that can be produced by biased coins; For every $0 < \delta < 1$, let $D_{n,\delta}$ be the following distribution over $\{0,1\}^n$. A random variable $X = (X_1, \ldots, X_n)$ with distribution $D_{n,\delta}$ has $\Pr[X_i = 1 | X_1, \ldots, X_{i-1} = \vec{z}] = \delta$, for every $1 \leq i \leq n$ and for every $\vec{z} \in \{0,1\}^{i-1}$.

von Neumann's protocol extracts truly random bits from any source X with distribution $D_{n,\delta}$, for every $0 < \delta < 1$. It divides the input string into $\frac{n}{2}$ pairs of bits. For every pair, it outputs 0 if the pair is (01) or 1 if the pair is (10), and outputs nothing if the pair is (00) or (11). The probability of outputting 1 and the probability of outputting 0 is equal to $\delta \cdot (1-\delta)$, and as we assume the input bits are independent, so are the output bits. This algorithm works for any δ, although the expected number of output bits is not optimal. Elias [Eli72], and later Peres [Per92], showed how to extract truly random bits at optimal rate, which is the entropy rate of the source.

More general types of sources were studied by Blum [Blu84]. He studied sources where the bits are assumed to be the output of a finite Markov chain, with unknown (fixed) transition probabilities. He showed an algorithm to extract perfect random bits at optimal rate from such sources.

Cohen and Wigderson [CW89] studied bit fixing sources, where a subset of the bits have fixed values, and the rest of the bits are unbiased and independent.

Santha and Vazirani ([SV86]) studied sources that are now called *SV-sources*. For $0 \leq \delta \leq \frac{1}{2}$, a δ-SV-source $X = (X_1, \ldots, X_n)$ is a random variable over $\{0,1\}^n$, such that for every $i = 1, \ldots, n$ and for every $\vec{z} \in \{0,1\}^{i-1}$, $\quad \delta \leq \Pr[X_i = 1 | X_1, \ldots, X_{i-1} = \vec{z}] \leq 1 - \delta$.

We think of such a δ-SV-source as being generated by an adversary, who, after seeing the values of $x_1, \ldots, x_{i-1}$, can set the bias of x_i (as long as that bias is smaller than $\frac{1}{2} - \delta$).

A somewhat surprising result is that it is impossible to extract even a single bit with bias smaller than $(\frac{1}{2} - \delta)$ from a general δ-SV-source ([SV86]).

However, it is possible to extract quasi-random bits from *two independent* δ-SV-sources. In a series of works ([SV86, Vaz85, VV85, Vaz87a, Vaz87b]) Santha & Vazirani, Vazirani & Vazirani, and Vazirani showed that $\Omega(n\delta^2)$ quasi-random bits can be extracted from two independent δ-SV-sources.

Chor and Goldreich [CG88] suggested a more general characterization of random sources, which they called probability bounded (PRB) sources. For PRB sources, every string $\vec{z} \in \{0,1\}^n$ has "small" probability to be the output of the source.

We say that X is a k-source if the maximum probability of any string $\vec{z} \in \{0,1\}^n$ is 2^{-k} (k is called the *min-entropy* of this source). [1]

PRB sources (as defined above) strictly generalize SV-sources; Every δ-SV-source is an $(n \cdot \log_2(\frac{1}{1-\delta}))$-source. But there are PRB sources that are not SV-sources. For example, an $(n-1)$-source that is uniform on 2^{n-1} of the strings in $\{0,1\}^n$ is not an SV-source. Thus, it is not surprising that it is also impossible to extract even a single unbiased random bit from a single PRB source (see section 4.1).

As with SV sources, it is sometimes possible to extract random bits from *two independent* PRB sources: Chor and Goldreich ([CG88]) show how to extract a single random bit from two PRB sources, if their min-entropies are high enough: If X is a b_X-source, and Y is a b_Y-source, and $b_X + b_Y$ are at least $n+2+2\log_2(\frac{1}{\varepsilon})$, then the bias of the extracted bit is ε.

They also give a non-constructive proof that even for a very low value of k, it is possible to extract random bits from *every* pair of k-sources. [2]

We review some of their techniques in detail in section 4.4.

PRB sources not only generalize all the sources mentioned above, but are also relatively simple to define and to manipulate.

[1] We note that the definition in [CG88] is different: They define a PRB source X as an infinite sequence of k-sources, $X_1, X_2, \ldots$, where for every positive integer i, for every $\vec{z} \in \{0,1\}^n$ and for every $\vec{w} \in \{0,1\}^{i \cdot n}$, the probability that $X_{i+1} = \vec{z}$, given that $(X_1, \ldots, X_i) = \vec{w}$, is at most 2^{-k}. This definition allows every problem in $\mathcal{RP}$ and in $\mathcal{BPP}$ to be decided using only a single weak source (see [VV85, CG88])

[2] They use counting arguments to show that almost all functions are good for this purpose, and give tradeoffs between the number of extracted bits, the min-entropies of the sources, and the bias of the extracted bits. For example, they show that all but 2^{-2^b} fraction of the functions $f : \{0,1\}^{2n} \to \{0,1\}$ have bias of $\varepsilon \leq 2^{-\frac{b-3-\log_2(2n+1)}{2}}$ on *all* pairs of independent b-sources X, Y. The min-entropy b must be at least $\log_2(2n+1) + 2\log(\frac{1}{\varepsilon}) + 3$. Note that b is smaller than n; Constructing an explicit function that achieves this is an open problem (also see section 6)

1.2.1 Extractors

Another approach for extracting randomness from weak random sources is to use a "short" truly random seed, in addition to the weak random source (as opposed to the methods we described above, that use *weak* random sources; Here the seed is truly random). Such constructions are called extractors, and were first defined by Nisan and Zuckerman ([NZ96]). A (k, ϵ)-extractor E is a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, such that $E(X, U_d)$ is ϵ-close to U_m, when X is a random variable with min-entropy of at least k. (Note that k, d, m, ϵ can be functions of n.) A probabilistic argument shows that for every k there exists a (k, ϵ)-extractor, for a constant ϵ, with $d = \log(n) + O(1)$ and $m = k + d - O(1)$. That is, the number of extracted bits is almost equal to the sum of min-entropies of the inputs.

A long line of research in the past decade focused on finding explicit constructions of extractors. Ta-Shma was the first to show an explicit construction of an extractor that has seed length $d = \log^{O(1)} n$ (for a constant ϵ). His extractor extracts all the randomness from the source, that is $m = k$ ([TS96]). Later explicit constructions improved on his results, with the best parameters to date achieved by [LRVW03]: They give an explicit construction of (k, ϵ)-extractor, for every k and for every constant ϵ, that has seed length $d = O(\log(n))$, and extracts almost all the randomness from the weak source; That is, $m = (1 - \alpha)k$, for any constant $\alpha > 0$. The error can be reduced to any ϵ' (for example, ϵ' that is inverse-super-polynomial in n) using the technique of [RRV02]. Using this technique, additional $O(\log(1/\epsilon') + \log(m))$ truly random bits are used, in order to reduce the bias to ϵ'. That is, additional $O(\log^2(n))$ truly random bits in the seed guarantees the output to be quasi-random.

There are several surveys and texts on extractors, such as [Nis96, NTS99] and a recent survey by Shaltiel [Sha02].

Extractors with the additional property that the output is almost independent from the seed are called strong extractors; For a strong extractor SE, the distribution of $(SE(X, U), U)$ is ϵ-close to the distribution of (U_m, U), when U has uniform distribution over $\{0,1\}^d$.

1.3 Our Contribution

It is commonly known that the extraction algorithm presented by Vazirani [Vaz87b], that works for SV-sources, also applies to the more general PRB sources. However, we have failed to find references that describe extraction of many quasi-random bits from two PRB sources. As far as we know, this is the first time where this algorithm is described in detail. In section 4.5 we show how to extract $\Omega(b_X + b_Y - n)$ bits, with bias of $2^{-\Omega(b_X+b_Y-n)}$, from two independent sources X, Y, that are b_X and b_Y-flat, respectively.

In section 5, we show how to go beyond these results, and extract $b_X + \Omega(b_X + b_Y - n)$ bits from such sources, with bias of $2^{-\Omega(b_X+b_Y-n)}$, when $b_X + b_Y - n = \Omega(polylog(n))$. This is the first explicit construction of a function that extracts $\Omega(n)$ quasi-random bits even when $b_X + b_Y - n = o(n)$.

Our construction uses two steps; We first show that the bits extracted in section 4.5 are independent of X. That is, the bits look random even to someone who has access to X. We then use an extractor to extract the randomness from X, and note that even for the extractor, the random seed can be replaced by the bits we extracted in the first step.

Proving that the bits extracted in the first step are independent of X was first shown by Dodis and Oliveira [DO], and communicated to us by them. Using probabilistic arguments, Dodis and Oliveira show that for inputs X, Y, which are b_X and b_Y-flat, respectively, it is possible to extract $m = b_Y - 2\log(\frac{1}{\epsilon})$ bits that are at most ϵ-biased (independent of X), when $b_X, b_Y \geq \log(n) + 2\log(\frac{1}{\epsilon}) + O(1)$. They also give an explicit construction that extracts $\Omega(b_X + b_Y - n)$ bits.

In our work, we give two explicit constructions of functions that extract $\Omega(b_X + b_Y - n)$ quasi-random bits, which are independent of X (and are different from the construction used at [DO]). Our proof works only when $b_X + b_Y - n = \Omega(polylog(n))$, and the extracted bits have bias of $2^{-\Omega(b_X+b_Y-n)}$.

We then show how to use an extractor that extracts all the randomness from X (for example, the extractor in [RRV02]) to get $b_X + \Omega(b_X + b_Y - n)$ bits.

2 Definitions

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. A set $S \subset [n]$ is also represented as a vector $S \in \{0,1\}^n$, with $S_i = 1 \iff i \in S$. A vector S is also a function $S : [n] \to \{0,1\}$, with $S(i) = S_i$. A vector S can also carry an integer value of $\sum_{i=1}^{n} 2^{i-1} S_i$ (that is, S is an index between 0 and $2^n - 1$).

When a variable is used mostly as a set, we will denote it by a capital letter such as S or T. When a variable is used mostly as an index we use x, i, j, etc. When we want to emphasize the vector meaning of an element $v \in \{0,1\}^n$ we denote it with an arrow, $\vec{v}$. We denote random variables by capital letters such as X, Y.

When f is a function on $S \times T$ and $i \in S$, we denote by f_i the function f where the first argument is set to i. That is, $f_i(j) \stackrel{\text{def}}{=} f(i,j)$ for every $j \in T$.

Let X be a random variable getting values in S. Without loss of generality, assume $|S| = 2^n$, for n integer, and furthermore that $S = \{0,1\}^n$. The probability function of X is the function $P_X : S \to \mathbb{R}$ such that $P_X(j)$ is the probability that X gets the value j (for every $j \in S$).

It is useful to think of P_X as a vector in $\mathbb{R}^{2^n}$, where the j-th index $(P_X)_j$ is equal to $P_X(j)$.

Recall that the L_1-norm of a vector $\vec{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$ (denoted $||\vec{v}||_1$) is defined as

$$||\vec{v}||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{n} |v_i|$$

Note that the L_1-norm of the vector P_X is 1.

We also recall that the L_2 norm of the vector $\vec{v}$ (denoted $||\vec{v}||_2$) is defined as

$$||\vec{v}||_2 \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{n} (v_i)^2}$$

For two vectors $\vec{v}, \vec{u} \in \mathbb{R}^n$, we denote by $\langle \vec{v}, \vec{u} \rangle$ their inner product, i.e.

$$\langle \vec{v}, \vec{u} \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{n} v_i \cdot u_i$$

For two vectors $\vec{v}, \vec{u} \in GF(2)^n$, we denote their inner product (mod 2) by $\langle \vec{v}, \vec{u} \rangle_2$, i.e.

$$\langle \vec{v}, \vec{u} \rangle_2 \stackrel{\text{def}}{=} \sum_{i=1}^{n} v_i \cdot u_i \text{ (mod 2)}$$

Let $\vec{v} = (v_1, \ldots, v_n)$ be a vector in $\{0,1\}^n$. For every $S \subset [n]$, denote by $\vec{v}_S$ the xor of $\{v_i\}_{i \in S}$, i.e.

$$\vec{v}_S \stackrel{\text{def}}{=} \bigoplus_{i \in S} v_i$$

For a random variable X getting values in the set T, we denote bt $\mathbb{X}$ the set of elements in T that have non-zero probability; That is, $\mathbb{X} = \{x \in T : P_X(x) > 0\}$

If X is a random variable over T and f is a function from T to W, then $f(X)$ is a random variable getting values in W, such that

$$\Pr[f(X) = w] \;\; = \sum_{t \,:\, f(t)=w} \Pr[X = t]$$

Definition 1 (Min Entropy). *Min-Entropy of a random variable X is denoted $H_\infty(X)$, and is defined as*

$$H_\infty(X) = -\log_2 \left(\max_x \left\{ \Pr[X = x] \right\} \right)$$

If the most probable event happens with probability 2^{-k}, then the min-entropy of X is k.

Since $\sum_{x \in X} \Pr[X = x] = 1$, any random variable with min-entropy k has at least 2^k different values with non-zero probability (that is, $|\mathbb{X}| \geq 2^k$).

Min-entropy can serve as a measure of the information present in a sample from the random variable. We can compare the min-entropy to the (Shanon) entropy: in some sense, min-entropy "measures" the information in the *worst case*, that is, when the value sampled is the most probable value, as opposed to Shanon's entropy, which measures the information in the *average case*.

X is called *flat* on S if for every $\alpha, \beta \in S$, $\Pr[X = \alpha] = \Pr[X = \beta]$ and for every $\alpha \notin S$, $\Pr[X = \alpha] = 0$.

As mentioned in the introduction, it is not always possible to extract truly random bits. The "quality" of the extracted bits is measured by their statistical distance (bias) from truly random bits.

Definition 2 (Statistical Distance). *Let P_X, P_Y be two probability functions on $\{0,1\}^n$. The statistical distance $\Delta(P_X, P_Y)$ is defined as*

$$\Delta(P_X, P_Y) = \frac{1}{2} \sum_{i \in \{0,1\}^n} \left| P_X(i) - P_Y(i) \right|$$

The statistical distance $\Delta(X, Y)$ of two random variables X, Y, is the statistical distance $\Delta(P_X, P_Y)$ of their probability functions, P_X, P_Y.

Two probability functions (random variables) that have statistical distance at most ϵ are called ϵ-close.

The Bias of X is the statistical distance of X from U_n.

We also define the bias of a matrix over $\{\pm 1\}$. Let C be an $n \times m$ matrix over $\{\pm 1\}$, then

$$\mathrm{Bias}(C) \stackrel{\text{def}}{=} \frac{1}{2} \cdot \frac{\left| \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} \right|}{n \cdot m}$$

We call the term $\left| \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} \right|$ the *elements sum* of C. Matrices with zero elements sum are called *balanced matrices* or unbiased matrices.

The bias of a row (respectively, column) of C is the bias of the submatrix of C that consists of this row (resp., column).

C is at most ε-biased if $\mathrm{Bias}(C) \leq \varepsilon$, and at least ε-biased if $\mathrm{Bias}(C) \geq \varepsilon$.

3 Preliminaries

3.1 Flat Distributions

It is relatively easy to deal with flat distributions, as we shall see in Section 4.4. Also, the "worst behavior" of extraction functions occurs on flat distributions, as the following lemma shows

Claim 1 ([CG88], Lemma 5). *For every function* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ *and every* $\alpha \in \{0,1\}^k$

$$\sup_{\substack{H_\infty(X)=b_X \\ H_\infty(Y)=b_Y \\ X,Y \text{ are independent}}} \{\Pr[f(X,Y)=\alpha]\} = \max_{\substack{H_\infty(X)=b_X, \text{ and } X \text{ is flat} \\ H_\infty(Y)=b_Y, \text{ and } Y \text{ is flat} \\ X,Y \text{ are independent}}} \{\Pr[f(X,Y)=\alpha]\}$$

and

$$\inf_{\substack{H_\infty(X)=b_X \\ H_\infty(Y)=b_Y \\ X,Y \text{ are independent}}} \{\Pr[f(X,Y)=\alpha]\} = \min_{\substack{H_\infty(X)=b_X, \text{ and } X \text{ is flat} \\ H_\infty(Y)=b_Y, \text{ and } Y \text{ is flat} \\ X,Y \text{ are independent}}} \{\Pr[f(X,Y)=\alpha]\}$$

Thus, when extracting randomness from two PRB sources over $\{0,1\}^n$, we can assume these sources are flat.

3.2 Fourier Transform

We will use some properties of the Fourier representation of a function. Consider an arbitrary function $g : \{0,1\}^n \to \mathbb{R}$. It will be convenient to view g as a vector in $\mathbb{R}^{2^n}$, with the x-th coordinate being the value of $g(x)$ (for every $1 \leq x \leq 2^n$). This mapping of functions to vectors is 1-to-1. The *Fourier base* is an orthonormal base for $\mathbb{R}^{2^n}$, and the (discrete) Fourier transform is the transformation from a representation of a vector in the standard base, into a representation of this vector in the Fourier base.

The Fourier representation and the Fourier transform for functions are implicitly defined by the Fourier representation and the Fourier transform for vectors.

Let K (after the Kronecker delta functions) be the *standard base* for $\mathbb{R}^{2^n}$:

$$K = \{1_x\}_{x \in \{0,1\}^n}$$

where $1_x = \overbrace{(0, \ldots, 0, 1, 0, \ldots, 0)}^{\text{length } 2^n}$ has a single 1 in the location $x{+}1$ (here we take $x \in \{0,1\}^n$ as an index $0 \leq x \leq 2^n - 1$).

As a function, $1_x(y) = 1$ iff $y = x$. We write the function g as

$$g(y) = \sum_{x \in \{0,1\}^n} g_x \cdot 1_x(y)$$

where

$$g_x = \langle g, 1_x \rangle = \sum_{z \in \{0,1\}^n} g(z) \cdot 1_x(z) = g(x)$$

Let χ be the Fourier base for $\mathbb{R}^{2^n}$:

$$\chi = \{\chi_S\}_{S \subset [n]}$$

where for every $T \subset [n]$, $\chi_S(T) = 2^{-n/2} \cdot (-1)^{|S \cap T|}$ (Here we take the indices S and T as subsets of $[n]$, although they can also be treated as elements of $\{0,1\}^n$)

As a function, $\chi_S(T) = \begin{cases} 2^{-\frac{n}{2}} & \text{when } |S \cap T| \text{ is even} \\ -2^{-\frac{n}{2}} & \text{when } |S \cap T| \text{ is odd} \end{cases}$

The representation of g in Fourier base is

$$g(T) = \sum_{S \subset [n]} \widehat{g_S} \cdot \chi_S(T)$$

for every $T \subset [n]$, where the coefficients $\widehat{g_S}$ are

$$\widehat{g_S} = \langle g, \chi_S \rangle = \sum_{T \subset [n]} g(T) \cdot \chi_S(T) = 2^{-\frac{n}{2}} \left(\sum_{T : |T \cap S| \text{ even}} g(T) - \sum_{T : |T \cap S| \text{ odd}} g(T) \right) \quad (1)$$

It is easy to verify that both bases are indeed orthonormal, and that both representations of g give the same vector (i.e., compute the same function).

For orthonormal bases, it is well known that the L_2 norm of a vector is invariant of the base the vector is represented in. This is known as the *Parseval equality*:

$$\sum_{x \in \{0,1\}^n} (g_x)^2 = \sum_{S \subset [n]} (\widehat{g_S})^2 \quad (2)$$

3.3 The xor-lemma

Let $X = (X_1, \ldots, X_n)$ be a random variable getting values in $\{0,1\}^n$ and let $P = P_X$ be the probability function of X.

For every $S \subset [n]$, denote by X_S the xor of $\{X_i\}_{i \in S}$, i.e.

$$X_S \stackrel{\text{def}}{=} \bigoplus_{i \in S} X_i$$

There is an equivalence between the following two conditions (see also [NN90])

1. X is uniformly distributed on $\{0,1\}^n$.
2. For every non empty $S \subset [n]$, X_S is unbiased.

When the values $\{X_S\}_{S \neq \emptyset}$ are slightly biased, X is still close to uniform. Vazirani's xor-lemma relates the bias of X to the maximal bias of X_S, when $S \subset [n]$.

Lemma 2 (xor-lemma). *For X as above*

$$\text{Bias}(X) \leq \sqrt{\sum_{\emptyset \neq S \subset [n]} (\text{Bias}(X_S))^2} \leq 2^{n/2} \cdot \max_{\emptyset \neq S \subset [n]} \{\text{Bias}(X_S)\}$$

The proof is immediate from the following two claims:

Claim 3. *For X, P as above,*

$$2 \cdot \text{Bias}(X) = \sum_{x \in \{0,1\}^n} \left| P(x) - \frac{1}{2^n} \right| \leq 2^{n/2} \sqrt{\sum_{\emptyset \neq S \subset [n]} (\widehat{P_S})^2}$$

Proof. By definition,

$$\text{Bias}(X) = \frac{1}{2} \cdot \sum_{x \in \{0,1\}^n} \left| \Pr[X = x] - \frac{1}{2^n} \right|$$

Denote $g(x) \stackrel{\text{def}}{=} P(x) - \frac{1}{2^n}$, and note that we can write

$$\sum_{x \in \{0,1\}^n} |g(x)| = \sum_{x \in \{0,1\}^n} |g(x)| \cdot 1$$

We use the Cauchy-Schwartz inequality: for two vectors $u, v \in \mathbb{R}^n$, $\langle u, v\rangle \leq ||u||_2 \cdot ||v||_2$.

We get

$$\sum_{x\in\{0,1\}^n} \left| \Pr[X = x] - \frac{1}{2^n} \right| \leq 2^{n/2} \sqrt{\sum_{x\in\{0,1\}^n} (g(x))^2}$$

Using Parseval equality, this is equal to

$$2^{n/2} \sqrt{\sum_{S\subset[n]} (\widehat{g_S})^2}$$

Note that $\widehat{g_S} = \begin{cases} \widehat{P_S} & S \neq \emptyset \\ 0 & S = \emptyset \end{cases}$

Therefore we can write

$$2 \cdot \mathrm{Bias}(X) \;=\; \sum_{x\in\{0,1\}^n} \left| P(x) - \frac{1}{2^n} \right| \;\leq\; 2^{n/2} \sqrt{\sum_{\emptyset\neq S\subset[n]} (\widehat{P_S})^2}$$

□

Claim 4. *For X, P as above, $|\widehat{P_S}| = \frac{2}{2^{n/2}} \cdot \mathrm{Bias}(X_S)$.*

Proof. From equation 1 (page 13), we know

$$\widehat{P_S} = 2^{-n/2} \left(\sum_{T:|S\cap T| \text{ even}} P(T) - \sum_{T:|S\cap T| \text{ odd}} P(T) \right)$$

But

$$X_S = 0 \text{ when } |X \cap S| \text{ is even, and } X_S = 1 \text{ otherwise.}$$

therefore

$$|\widehat{P_S}| = \frac{1}{2^{n/2}} \cdot \left| \Pr[X_S = 0] - \Pr[X_S = 1] \right| = \frac{2}{2^{n/2}} \cdot \mathrm{Bias}(X_S)$$

□

4 Previous work

In section 4 we describe the works of Vazirani [Vaz87b] and Chor and Goldreich [CG88], on extracting quasi-random bits from weak random sources. In section 4.4 we bring the proof of [CG88] that the given two weak random sources, the inner product modulo 2 of them can give a quasi-random bit; Denote the random sources by X, Y, and assume that X is b_X-distributed and Y is b_Y-distributed. If $b_X + b_Y - n = \Omega(polylog(n))$ then $\langle X, Y \rangle_2$ is a quasi-random variable.

Vazirani showed how to extract $\Omega(n\delta^2)$ quasi-random bits from two δ-SV-sources, by taking $\Omega(n\delta^2)$-bits from a convolution between the two sources. It is consider common knowledge that his technique also applies to PRB-sources, but we did not find references that describe explicit constructions of functions that extract more than one bit from two PRB-sources. In section 4.5 we present two explicit constructions of functions that extract $\Omega(b_X + b_Y - n)$ quasi-random bits from two such PRB-sources, when $b_X + b_Y - n = \Omega(polylog(n))$. One of these functions is the one used in [Vaz87b]. We use the xor-lemma to show that the extracted bits are quasi-random; we show that the xor of every subset of the extracted bits has small bias, since it is the inner product of two random variables that have "high-enough" min-entropy.

4.1 Impossibility Results

As noted in the introduction, every extraction function designed to extract even a single unbiased bit, from a k-source, for $k \leq n-1$, will fail to extract randomness for some distributions. It is easy to see that, using the following argument of [CG88]:

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be an arbitrary Boolean function, then there exists a $\sigma \in \{0,1\}$ such that f outputs σ for at least half the inputs $x \in \{0,1\}^n$. Let S be a set of size $2^k \leq 2^{n-1}$ on which f outputs σ, and let X be a random variable uniformly distributed on S. Then X is a k-source, and $f(X)$ is identically σ.

The situation improves if we use several independent sources.

4.2 Extracting from Two Independent Sources

Given two strings $x, y \in \{0,1\}^n$, let $CS(x,y)$ be the *convolution of* x, y, i.e. a string $z \in \{0,1\}^n$, such that z_i is the inner product (mod 2) of x and $\Pi^{i-1}y$, where $\Pi^{i-1}y$ is the cyclic shift of y by $i-1$ positions to the right.

Vazirani [Vaz87b] showed that if X, Y are independent δ-SV-sources, then $k = O(n\delta^2)$ bits taken from $CS(X,Y)$ are quasi-random (for example, the first k bits of $CS(X,Y)$ are quasi-random).

We use the same method when X, Y are PRB sources.

Notation: From now on, we will assume X is a b_X-source and Y is a b_Y-source, both over $\{0,1\}^n$, and we assume $b_X \geq b_Y$. We write for short $H_\infty(X) + H_\infty(Y) = b_X + b_Y$.

Using Vazirani's extraction function, we show that we can extract $k = \Omega(b_X + b_Y - n)$ bits with bias $2^{-\frac{b_X+b_Y-n-k-1}{2}}$ from X, Y.

We also consider a similar function, that extracts bits at a slightly slower rate. Define $RS(x,y) = \vec{z}$, where $z_i = \langle x, \tilde{\Pi}^{i-1}y \rangle_2$, and $\tilde{\Pi}^{i-1}y$ is the (non-cyclic) right shift of y by $i-1$ positions. With this function, we extract $k' = \Omega(b_X + b_Y - n)$ bits with bias $2^{-\frac{b_X+b_Y-n-2k'+2}{2}}$. That is, for the same bias, we can extract k bits using $CS(X,Y)$ or $k' = (k+3)/2$ bits using $RS(X,Y)$. Note that for the purpose of using the extracted bits as a seed for an extractor, a constant factor in the number of bits does not make a significant difference. However the proof that CS is a good extraction function involves a non-trivial argument and is limited to values of n that are primes with 2 as a primitive root modulo n. Thus we prefer to think of RS as the building block we use for the next step of our algorithm.

A main part of the proof (for both functions RS and CS), is a claim from [CG88], saying that the inner product (mod 2) of X, Y (i.e., $\langle X, Y \rangle_2$) is a random bit with bias of $2^{-\frac{b_X+b_Y-n}{2}-1}$ (see section 4.4).

The analysis of the extraction functions that we study, shows that for for every subset of the extracted bits, the parity of these bits corresponds to an inner product of X and Y', where X, Y are the inputs and Y' has min-entropy that is almost the min-entropy of Y. Therefor if Y has high min-entropy, we have small bias for the parity of every subset of bits in the output, and we use the xor-lemma to bound the bias of the output bits.

	00	01	10	11
00	1	1	1	1
01	1	-1	1	-1
10	1	1	-1	-1
11	1	-1	-1	1

Figure 1: A Matrix F describing the function $f(x,y) = \langle x,y \rangle_2$

4.3 Describing a Boolean Function by a $\{\pm 1\}$ Matrix

For every Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let $F = (F_{x,y})$ be a $2^n \times 2^n$ table with entries in $\{\pm 1\}$, such that $F_{x,y} = (-1)^{f(x,y)}$ (for an example, see figure 1). We call F *the matrix describing* f.

When X and Y are flat, the probability of $f(X,Y) = b$ (for $b \in \{0,1\}$) is the probability of $(-1)^b$ in the submatrix $\mathbb{X} \times \mathbb{Y}$ of F.

Consequently,

$$\begin{aligned} \mathrm{Bias}(f(X,Y)) &= \frac{1}{2} \cdot \Big| \Pr_{(x,y)\in\mathbb{X}\times\mathbb{Y}}[F_{x,y} = 1] - \Pr_{(x,y)\in\mathbb{X}\times\mathbb{Y}}[F_{x,y} = -1] \Big| \\ &= \frac{1}{2} \cdot \frac{\left| \sum_{(x,y)\in\mathbb{X}\times\mathbb{Y}} F_{x,y} \right|}{|\mathbb{X}| \cdot |\mathbb{Y}|} \end{aligned} \tag{3}$$

the last term is the *bias of the submatrix* $(\mathbb{X} \times \mathbb{Y})$ *of* C.

4.4 Extracting a Single Bit

A special class of almost balanced matrices is the Hadamard matrices. Hadamard matrices are square matrices with entries in $\{\pm 1\}$, where every two distinct rows (respectively, columns) are orthogonal.

When f is the inner-product (mod 2) function, $F_{i,j} = (-1)^{\langle i,j \rangle_2}$. For every $i \neq j$, rows i and j (resp., columns i and j) of F are orthogonal. That is, F is a $2^n \times 2^n$ Hadamard matrix.

A submatrix of size $r \times s$ of a $2^n \times 2^n$ Hadamard matrix is at most $\frac{1}{2}\sqrt{\frac{2^n}{rs}}$-biased.

Claim 5 ([CG88], Lemma 8). *Let $H = (h_{i,j})$ be a $t \times t$ Hadamard matrix. Then the sum of elements in every $r \times s$ submatrix of H is at most $\sqrt{s \cdot r \cdot t}$.*

Proof. Let $\vec{h}_i$ be row i of H. Assume wlog that the submatrix consists of the first r rows and s columns of H, and let $\vec{I} = (\overbrace{1, \ldots, 1}^{\text{s ones}}, \overbrace{0, \ldots, 0}^{t-s \text{ zeros}})$. The inner-product of $\vec{h}_i$ and $\vec{I}$ is the sum

$$\langle \vec{h}_i, \vec{I} \rangle = \sum_{j=1}^{t} (\vec{h}_i)_j \cdot I_j = \sum_{j : I_j = 1} (\vec{h}_i)_j$$

The sum of elements in the $r \times s$ submatrix of H is

$$\begin{aligned}
\Big| \sum_{i=1}^{r} \sum_{j=1}^{s} h_{i,j} \Big| &= \Big| \sum_{i=1}^{r} \langle \vec{h}_i, \vec{I} \rangle \Big| = \Big| \langle \sum_{i=1}^{r} \vec{h}_i, \vec{I} \rangle \Big| \\
\text{(using Cauchy Schwartz inequality)} \quad &\leq || \sum_{i=1}^{r} \vec{h}_i ||_2 \cdot ||\vec{I}||_2 \\
&= \sqrt{\langle \sum_{i=1}^{r} \vec{h}_i, \sum_{i=1}^{r} \vec{h}_i \rangle} \cdot \sqrt{s} \\
&= \sqrt{\sum_{i=1}^{r} \sum_{j=1}^{r} \langle \vec{h}_i, \vec{h}_j \rangle} \cdot \sqrt{s} \\
\text{(by orthogonality of the rows)} \quad &= \sqrt{r \cdot t \cdot s}
\end{aligned}$$

□

Corollary 6. *If F is a $2^n \times 2^n$ matrix that describes the inner-product function (mod 2), then every submatrix of size $r \times s$ of F has bias at most $\frac{1}{2}\sqrt{\frac{2^n}{rs}}$.*

Proof. Let $\mathbb{X}$ be an arbitrary set of r rows of F and let $\mathbb{Y}$ be an arbitrary set of s columns of F.

By equation 3,

$$\text{Bias}(\text{subset } \mathbb{X} \times \mathbb{Y} \text{ of } F) = \frac{1}{2} \frac{|\sum_{x,y \in \mathbb{X} \times \mathbb{Y}} F_{x,y}|}{rs}$$

and by claim 5

$$\Big|\sum_{x,y\in\mathbb{X}\times\mathbb{Y}} F_{x,y}\Big| \leq \sqrt{2^n rs}$$

Therefore

$$\mathrm{Bias}(\text{submatrix } (\mathbb{X}\times\mathbb{Y}) \text{ of } F) \leq \frac{1}{2}\sqrt{\frac{2^n}{rs}}$$

□

Note that the results of claim 5 and of corollary 6 depend on the product $r \cdot s$, rather than on r and s separately. When we use these results in the context of flat random variables, we require that the sum of the min-entropies of X and of Y is high, rather than that both X and Y have high min-entropy.

Corollary 7 ([CG88], Theorem 9). *Let X, Y be flat random variables over $\{0,1\}^n$, such that $H_\infty(X) + H_\infty(Y) = b_X + b_Y$, then*

$$\mathrm{Bias}(\langle X, Y\rangle_2) \leq 2^{-\frac{b_X+b_Y-n}{2}-1}$$

4.5 Extracting $\Omega(\mathbf{b_X + b_Y - n})$ Bits

Consider a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ that is given by

$$f(x,y) \stackrel{\text{def}}{=} \left(\langle x, A^0 y\rangle_2 \ , \ \langle x, A^1 y\rangle_2 \ , \ \ldots \ , \ \langle x, A^{k-1} y\rangle_2\right)$$

for every $x, y \in \{0,1\}^n$, where A is an $n \times n$ matrix over $GF(2)$.

We use the xor-lemma to bound the bias of $f(X,Y)$. We first must bound the bias of $f(X,Y)_S$ (for every non-empty set $S \subset [k]$). We note that

$$f(X,Y)_S = \bigoplus_{i\in S} \langle X, A^{i-1}Y\rangle_2 \ = \langle X, \overbrace{\sum_{i\in S} A^{i-1}}^{\tilde{A}(S)} Y\rangle_2$$

If A were a matrix such that $\tilde{A}(S)$ is regular for every non-empty S, then the min-entropy of $\tilde{A}(S)Y$ would be the same as the min-entropy of Y, and the bias of $f(X,Y)_S$ would be the same as the bias of $\langle X, Y\rangle_2$. The bias of $\langle X, Y\rangle_2$ is guaranteed to be small if X, Y have enough min-entropy.

Following this motivation, we note that if A is such that $\tilde{A}(S)$ has high rank, then the min-entropy of $\tilde{A}(S)Y$ is close to the min-entropy of Y. Therefore, we can expect that if X, Y have enough min-entropy, then $f(X,Y)_S$ has bias that is only slightly bigger than the bias of $\langle X, Y \rangle_2$.

We study two such linear transformations A; One is the cyclic right-shift and the other is the non-cyclic right-shift.

4.5.1 An Extraction Function based on Right-Shift

Let $A \in \{0,1\}^{n \times n}$ be the matrix of right-shift by 1; For every $\vec{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$ $A\vec{b} = (0, b_1, \ldots, b_{n-1})$.
A^j is the matrix of right shift by j, for $j \geq 0$ integer. Let S be a non-empty subset of $[k]$, and let $\tilde{A} = \tilde{A}(S) \stackrel{\text{def}}{=} \sum_{i \in S} A^{i-1}$. $\tilde{A}$ is a matrix whose upper-right triangle is all zeros, and has $|S|$ diagonals, all in a band of width k in the lower-left triangle (see figure 2).

Every non-zero row of $\tilde{A}$ is independent of the other rows of $\tilde{A}$ (and the same holds for columns of $\tilde{A}$). For proving this, it is enough to look at the maximal non-zero coordinate of row i, vs. the maximal non-zero coordinates of rows $1, \ldots, i-1$.

Let $r = \min_i\{i \in S\} \leq k$. It is easy to see that there are $n - r + 1$ non-zero rows (resp., columns) in $\tilde{A}$. Therefore,

$$rank(\tilde{A}) = n - r + 1 \geq n - k + 1$$

hence $\tilde{A}$ is a 2^{r-1}-to-1 mapping over $GF(2)^n$. We conclude that $H_\infty(\tilde{A}Y) \geq H_\infty(Y) - k + 1$ for every random variable Y over $\{0,1\}^n$.

Formally, we define the extraction function $RS_k : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ by

$$RS_k(x,y) \stackrel{\text{def}}{=} \left(\langle x, A^0 y \rangle_2 \ , \ \langle x, A^1 y \rangle_2 \ , \ \ldots \ , \ \langle x, A^{k-1} y \rangle_2 \right)$$

where A is the right-shift matrix.
If $H_\infty(X) + H_\infty(Y) = b_X + b_Y$, then $H_\infty(X) + H_\infty(\sum_{i \in S} A^{i-1} Y) \geq b_X + b_Y - k$ for every S. By corollary 7

$$\text{Bias}(RS_k(X,Y)_S) \leq 2^{-\frac{b_X + b_Y - n - k}{2} - 1}$$

We now use the xor-lemma to get the following corollary:

Corollary 8. *If $H_\infty(X) + H_\infty(Y) = b_X + b_Y$, then the bias of $RS_k(X,Y)$ is at most $2^{-\frac{b_X + b_Y - n - 2k}{2} - 1}$.*

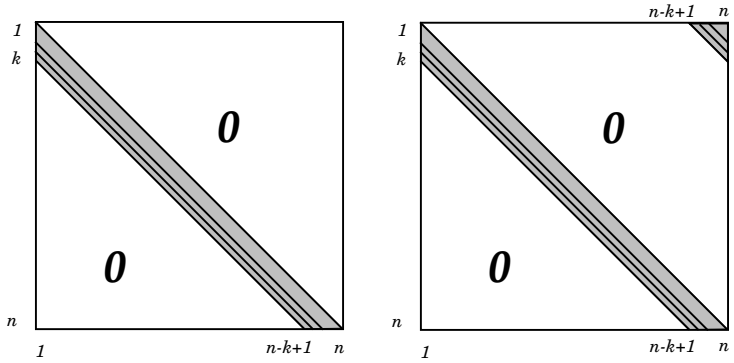

Figure 2: A general structure of a matrix $\tilde{A} = \tilde{A}(S) = \sum_{i \in S} A^{i-1}$, for S a non-empty subset of $[k]$, where A is the (non-cyclic) right-shift matrix (left), and when A is the cyclic-right-shift matrix (right).

4.5.2 An Extraction Function based on Cyclic Right-Shift

The function RS_k presented above allows for a sub optimal bias, since the rank of the matrix $\tilde{A}$ can be as low as $n - k + 1$.

A similar linear function, which was used in [Vaz87b], has $rank(\tilde{A}) \geq n-1$. The higher rank allows us to extract twice as many bits using this function.

Let A be the matrix of cyclic right-shift by 1. A^i is the matrix of cyclic right shift by i. The matrix $\tilde{A} = \tilde{A}(S) = \sum_{i \in S} \tilde{A}^{i-1}$, for non-empty $S \subset [k]$, may have $|S| \leq k$ diagonals (see figure 2).

Formally, we define an extraction function $CS : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ as

$$CS(x,y) \stackrel{\text{def}}{=} \left(\langle x, A^0 y \rangle_2 \ , \ \langle x, A^1 y \rangle_2 \ , \ \ldots \ , \ \langle x, A^{k-1} y \rangle_2 \right)$$

where A is the cyclic right shift matrix. Note that $CS_k(X,Y)$ is the same as taking the first k bits from the convolution of (X,Y).

Similarly to the previous section, we show that $\tilde{A}$ has very high rank. The proof here is not as straight forward as the proof of 4.5.1, and it applies only when n is prime with 2 as a primitive root modulo n.

Claim 9 ([Vaz87b]). *Let n be a prime with 2 a primitive root modulo n (i.e. 2 is a generator of Z_n^*). Let $\vec{u} \in \{0,1\}^n \setminus \{0^n, 1^n\}$ be a vector which is not the all 0's or the all 1's vector. Let B be an $n \times n$ matrix over $GF(2)$, such that the rows of B are the n right-cyclic-shifts of $\vec{u}$. Then $rank(B) \geq n - 1$.*

Proof. Let $f \stackrel{\text{def}}{=} 1 + \cdots + x^{n-1}$. It can be shown that f is irreducible in $GF(2)[x]$. Let K be the field $GF(2)[x]/f$. We represent each element $v \in K$ by a polynomial of degree at most $n-2$ over $GF(2)$ (and also as a vector of coefficients $\vec{v} \in GF(2)^{n-1}$). $|K| = 2^{n-1}$, thus K is isomorphic to $GF(2^{n-1})$.

Let $g \stackrel{\text{def}}{=} f \cdot (x-1) = x^n - 1$, and let R be the ring $GF(2)[x]/g$. We represent each element $v \in R$ by a polynomial of degree at most $n-1$ over $GF(2)$ (and also as a vector of coefficients $\vec{v} \in GF(2)^n$). Note that $x^n =_R 1$; In R, multiplication by x causes a cyclic shift in the coefficients. That is, for every $v \in R$, the vector $(\overrightarrow{x \cdot v}) = \Pi(\vec{v})$, where Π is the right-cyclic-shift.

Also, since $g = f \cdot (x-1)$ we get that $GF(2)[x]/g \equiv GF(2)[x]/f \times GF(2)[x]/(x-1)$, i.e. $R = K \times GF(2)$.

Consider the isomorphism $\alpha : R \to K$, given by $\alpha(v) \stackrel{\text{def}}{=} v \ (mod\ f) \in K$ for every $v \in R$. Note that α is a 2-to-1 mapping, with $\alpha(v) = \alpha(f - v)$ for every $v \in R$.

Let $u^{-1} = \sum_{i=0}^{n-2} a_i x^i \in K$ be the inverse of u in K, i.e. $u^{-1} \cdot u =_K 1$.

Denote by $\vec{b}_i$ the ith row of B. Then $\vec{b}_1 = \vec{u}$, and for every $i \in \{2, \ldots, n\}$, $\vec{b}_i = \Pi^{i-1}\vec{u}$; That is, the rows of B are the elements $\{u\ ,\ x \cdot u\ ,\ x^2 \cdot u\ , \ldots ,\ x^{n-1} \cdot u\}$ of R.

We say that $x^t \in K$ lies in the span of the rows of B, if there exist $S \subset [n]$ s.t. $\alpha(\sum_{i \in S} \vec{b}_i) = x^t$. We note that 1 lies in the span of the rows of B, since

$$\alpha(\sum_{i\ :\ a_i = 1} \vec{b}_{i+1}) = \alpha(\sum_{i=0}^{n-2} a_i x^i \ \cdot u) = \alpha(u^{-1} \cdot u) = 1 \in K$$

Similarly, $x, x^2, \ldots, x^{n-2}$ all lie in the span of the rows of B. Therefore, we can say that K is in the span of the rows of B.

Assume, for the sake of contradiction, that $rank(B) = r' \leq n-2$, then the rows of B can be expressed as the linear combinations of r' vectors in $GF(2)$, and thus the span of the rows of B would be of size at most $2^{r'} \leq 2^{n-2}$. □

Corollary 10. *Let A be the cyclic right shift matrix. The linear transform given by the matrix $\tilde{A} = \sum_{i \in S} A^{i-1}$ (for $S \neq [n], \emptyset$) is either an injective map or a 2-to-1 map, where for every $y \in \{0,1\}^n$, both y and $(1^n - y)$ are mapped to the same element.*

Note that since $S \subset [k]$, for $k < n$ we get $S \subsetneqq [n]$.

Similar to corollary 8, we get

Corollary 11. *If $H_\infty(X) + H_\infty(Y) = b_X + b_Y$, the bias of $CS_k(X, Y)$ is at most $2^{-\frac{b_X+b_Y-n-k-1}{2}}$*

4.6 Summary

The two functions RS_k and CS_k, presented above, allow for the extraction of $\Omega(b_X + b_Y - n)$ bits with bias of $2^{-\Omega(b_X+b_Y-n)}$. The parameters of CS_k are better than the parameters of RS_k by a constant factor: With CS_k, we can extract twice as much bits with the same bias.

Note that both functions are far from extracting all the randomness that is present in the sources; While the sum of min-entropies of the sources is $b_X + b_Y$, we extract at most $b_X + b_Y - n$ random bits.

To extract all the randomness from X, we use the bits extracted so far as a seed for an extractor. The seed length for an extractor must be polylogarithmic (in n), and an improvement by a constant factor in the seed length does not make a significant difference.

Also, the proof technique we use for proving the properties of CS_k works only when n is a prime with 2 as a primitive root modulo n. The proof for RS_k is simpler and works for every n.

To summarize, both functions have the properties we seek, and RS_k is more general than CS_k. In out two-steps algorithm for extracting all the randomness from X, we use RS_k in the first step.

5 Extracting Bits that are independent of X

We show that with very high probability (on X), $f(X,Y)$ is quasi-random even to someone who sees X.

Let $f_x(y) \stackrel{\text{def}}{=} f(x,y)$. We show that $f_x(Y)$ has small bias for almost every x, and conclude that $f(X,Y)$ has small bias *even given* X. [3]

5.1 Extracting a Single Bit

We extract a single bit from X,Y by taking the inner product modulo 2. We then show that if X,Y are flat, for almost all the values x of X the bias of $\langle x,Y\rangle_2$ is small; Thus, the bias of $\langle x,Y\rangle_2$ is almost independent of X.

Claim 12. *If X,Y are flat, and $H_\infty(X)+H_\infty(Y) = b_X+b_Y$, then for every $\xi > 0$*

$$\Pr_X\left[\text{Bias}\left(\langle x,Y\rangle_2\right) \geq \xi\right] \leq \frac{2^{-(b_X+b_Y-n)}}{4\xi^2}$$

Proof. Let F be the $\{\pm 1\}$ matrix describing the inner product function, i.e. $F_{x,y} = (-1)^{\langle x,y\rangle_2}$ for every $x,y \in \{0,1\}^n$. Denote by F' the the submatrix $(\mathbb{X}\times\mathbb{Y})$ of F. When X,Y are flat, the bias of F' is the bias of $\langle X,Y\rangle_2$.

When looking at F', we claim that the probability of any row of F' to have bias of ξ or more, is at most $\frac{2^{-(b_X+b_Y-n)}}{4\xi^2}$.

We use the following claim on Hadamard matrices:

Claim 13. *Let H be a $2^n\times 2^n$ Hadamard matrix, and let $\vec{h}_y$ denote the y-th column of H. Let $\mathbb{Y}\subset 2^n$ be a subset of the columns of H, $|\mathbb{Y}| = r$. Let $\tilde{H}$ be the $2^n\times r$ submatrix of H achieved by taking only the columns $\{\vec{h}_y\}_{y\in\mathbb{Y}}$.*

Then the number of rows *of $\tilde{H}$ that have (as submatrices) bias greater than ξ is at most $\frac{2^n}{4r\xi^2}$.*

[3] A weaker claim appears in the work of Chor and Goldreich ([CG88], lemma 27) and applies for every function $f:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}$. If $\text{Bias}(f(X,Y)\leq\varepsilon)$, when $H_\infty(X) = n-1$ and $H_\infty(Y) = b$, then only a $\sqrt{\varepsilon}$-fraction of the x's have $\text{Bias}(f_x(Y)) \geq 4\sqrt{\varepsilon}$.

Proof. Let $\overrightarrow{D}$ sum all the columns in $\tilde{H}$, i.e. $\overrightarrow{D} \stackrel{\text{def}}{=} \sum_{y\in\mathbb{Y}} \vec{h}_y$. For every $x \in \{0,1\}^n$, $\vec{D}_x$ equals the sum $\sum_{y\in\mathbb{Y}} H_{x,y}$; That is, $\vec{D}_x$ is the sum of elements in row x of $\tilde{H}$. If row x of $\tilde{H}$ has bias ξ or more, then $|\vec{D}_x| \geq 2r\xi$, and $\vec{D}_x^2 \geq 4r^2\xi^2$.

The square of the L_2-norm of $\overrightarrow{D}$ is

$$\sum_{x=1}^{2^n} \vec{D}_x^2 = ||\overrightarrow{D}||_2^2 = \langle \overrightarrow{D}^T , \overrightarrow{D} \rangle = \langle (\sum_{y\in\mathbb{Y}} \vec{h}_y)^T , \sum_{y\in\mathbb{Y}} \vec{h}_y \rangle = r \cdot 2^n$$

Since every ξ-biased row of $\tilde{H}$ contributes at least $4r^2\xi^2$ to the left-hand term, at most $\frac{2^n}{4r\xi^2}$ such rows exist. □

Since F is an Hadamard matrix, there are at most $\frac{2^n}{4|\mathbb{Y}|\xi^2}$ rows in F' that are ξ-biased. It follows that the probability of hitting a ξ-biased row, when selecting a row uniformly at random from the rows of F', is at most

$$\frac{\frac{2^n}{4|\mathbb{Y}|\xi^2}}{|\mathbb{X}|} = \frac{2^{(n-b_X-b_Y)}}{4\xi^2}$$

□

5.2 Extracting $\Omega(\mathbf{b_X + b_Y - n})$ Bits

We next show that RS_k (defined in 4.5), produce output that has small bias for almost all the values of X, and we show that this means the output is independent of X.

Lemma 14 (main lemma). *Let f be the function $RS_k : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$, defined in section 4.5.1. Define the function f_x, s.t. $f_x(y) \stackrel{\text{def}}{=} f(x,y)$, for every $x, y \in \{0,1\}^n$. If X, Y are flat and $H_\infty(X) + H_\infty(Y) = b_X + b_Y$, then for every $0 < \beta < 1$,*

$$\Pr_X[\text{Bias}(f_x(Y)) \geq \beta] \leq \frac{2^{-(b_X+b_Y-n-3k)}}{4\beta^2}$$

Proof. By the xor-lemma ,

$$\text{Bias}(f_x(Y)) \leq \sqrt{\sum_{\emptyset\neq S\subset[k]} \text{Bias}^2(f_x(Y)_S)} \leq 2^{k/2} \cdot \max_S \left\{ \text{Bias}(f_x(Y)_S) \right\}$$

where $f_x(Y)_S$ is the xor of the S bits of $f_x(Y)$.

We say that $x \in \{0,1\}^n$ is "good" if for every non-empty $S \subset [k]$,

$$\mathrm{Bias}(f(x,Y)_S) \leq \xi$$

That is, for every "good" x, the bias of $f(x,Y)$ is at most $2^{k/2} \cdot \xi$.

For every non-empty S, we know that $\tilde{A}(S)$ is a 2^r-to-1 mapping, for some $r < k$. Therefore, the min-entropy of $\tilde{A}(S)Y$ is at least $H_\infty(Y) - k$. Recall that $f(x,Y)_S = \langle x, \tilde{A}(S)Y \rangle_2$.

From section 5.1 we know that

$$\Pr_X \left[\mathrm{Bias}(f(x,Y)_S) \geq \xi \right] \leq \frac{2^{-(b_X+(b_Y-k)-n)}}{4\xi^2}$$

We use the union bound, to get the probability of getting a "bad" x

$$\Pr_X \left[\exists S \neq \emptyset \; s.t. \; \mathrm{Bias}(f(x,Y)_S) \geq \xi \right] \leq 2^k \cdot \frac{2^{-(b_X+(b_Y-k)-n)}}{4\xi^2} = \frac{2^{-(b_X+b_Y-n-2k)}}{4\xi^2}$$

Denote $\beta \stackrel{\text{def}}{=} 2^{k/2}\xi$ and use the xor-lemma to get

$$\Pr_X[\mathrm{Bias}(f_x(Y)) \geq \beta] \leq \frac{2^{-(b_X+b_Y-n-3k)}}{4\beta^2}$$

□

We next show that the main lemma guarantees that the distribution of $RS_k(X,Y)$ is close to the distribution U_k, even given X.

Claim 15. *Let X, Y be random variables over $\{0,1\}^n$, and let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ be such that*

$$\Pr_X \left[f_x(Y) \textit{ is } \beta\textit{-close to } U_k \right] = 1 - \mu$$

where $f_x(y) \stackrel{\text{def}}{=} f(x,y)$ for every $x, y \in \{0,1\}^n$.

Then the distribution of $(X, f(X,Y))$ is $(\beta+\mu)$-close to the distribution of (X,U), where U is distributed uniformly on $\{0,1\}^k$ independent of X.

Proof. Recall that the distance $\Delta\big((X, f(X,Y)), (X,U)\big)$ can be written as

$$\begin{aligned}
& \frac{1}{2} \cdot \sum_{(x,z)\in\{0,1\}^{n+k}} \left| \Pr\left[X = x, f(X,Y) = z\right] - \Pr\left[X = x, U = z\right] \right| \\
= \; & \frac{1}{2} \cdot \sum_{x\in\{0,1\}^n, z\in\{0,1\}^k} \left| \Pr[X = x] \cdot \Pr\left[f(X,Y) = z | X = x\right] \; - \; \Pr[X = x] \cdot 2^{-k} \right| \\
= \; & \sum_{x\in\{0,1\}^n} \Pr[X = x] \cdot \frac{1}{2} \sum_{z\in\{0,1\}^k} \left| \Pr\left[f(X,Y) = z | X = x\right] - 2^{-k} \right| \\
= \; & \sum_{x\in\{0,1\}^n} \Pr[X = x] \cdot \mathrm{Bias}(f_x(Y))
\end{aligned}$$

The probability of "bad" x's is at most μ, and for every "good" x, the bias of $(f(X,Y)|X = x)$ is at most β. Separating for "bad" and "good" x's, we get

$$\sum_{\text{"bad"}x} \Pr[X = x] \cdot \mathrm{Bias}(f(x,Y)) + \sum_{\text{"good"}x} \Pr[X = x] \cdot \mathrm{Bias}(f(x,Y)) \leq \mu \cdot 1 + 1 \cdot \beta$$

□

Corollary 16. *If X, Y are random sources with $H_\infty(X) \geq b_X, H_\infty(Y) \geq b_Y$, and $b_X + b_Y > n$, then for any $k < \frac{b_X + b_Y - n}{3}$ (i.e., k is $\Omega(b_X + b_Y - n)$), there are $\beta, \mu = 2^{-\Omega(b_X + b_Y - n)}$ such that $RS_k(X,Y)$ is $(\beta+\mu)$-close to U_k, even conditioned on X.*

Proof. For proving the corollary, it is enough to take k, β in the main lemma to be $k = c_1 \cdot (b_X + b_Y - n)$, for $0 < c_1 < 1/3$, and $\beta = 2^{-c_2 \cdot (b_X + b_Y - n)}$, for $0 < c_2 < \frac{1-3c_1}{2}$. The corollary follows from claim 15. □

Two cases of particular interest are

- $b_X + b_Y = n + \Omega(polylog(n))$: in this case, we extract $k = polylog(n)$ bits that are $(2^{-polylog(n)})$-close to U_k. That is, we extract $polylog(n)$ bits which are quasi-random.

- $b_X + b_Y = n \cdot (1 + c)$ for some $0 < c < 1$: in this case, we extract $k = \Omega(n)$ bits, that are $(2^{-\Omega(n)})$-close to U_k. Here we extract a linear number (in n) of bits with exponentially small bias.

We should note that the parameters of corollary 16 can be improved, both by using the function CS_k (which has better parameters than RS_k, as noted in section 4.6) and by using a bound more tight than the union bound we use in the proof for the main lemma. When f is the function CS_k, and using the tighter proof, we can write (similarly to the main lemma)

$$\Pr_X[\mathrm{Bias}(f_x(Y)) \geq \beta] \leq \frac{2^{-(b_X+b_Y-n-k)}}{4\beta^2}$$

using this bound, we can extract $k' = c_1 \cdot (b_X + b_Y - n)$ for $c_1 < 1$, with bias of $\beta' = 2^{-c_2 \cdot (b_X+b_Y-n)}$, for $c_2 \leq \frac{1-c_1}{2}$, when $b_X + b_Y > n$.

5.3 Extracting All the Randomness From X

We showed how to extract $k = \Omega(b_X + b_Y - n)$ bits, with bias $\delta \leq 2^{-\Omega(b_X+b_Y-n)}$, independent of X. Our next step is to use these bits to extract all the randomness from X.

Claim 17 (main result). *For every b_X, b_Y such that $b_X + b_Y = n + \Omega(polylog(n))$, there exists an explicit construction of a function $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$, with $m = b_X + c \cdot (b_X + b_Y - n) - O(1)$, for any constant $0 < c < 1/3$, such that for every pair of independent random variables X, Y over $\{0,1\}^n$, with $H_\infty(X) \geq b_X$ and $H_\infty(Y) \geq b_Y$, the output of $g(X,Y)$ is $(2^{-\Omega(b_X+b_Y-n)})$-close to uniform.*

Proof. Our main tool for this claim will be the use of extractors. [4]

Definition 3 ((l,ϵ)-Extractor). *A function $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^m$ is an (l,ϵ)-extractor if for every distribution X over $\{0,1\}^n$ with $H_\infty(X) \geq l$, $E(X, U_k)$ is ϵ-close to uniform. (Note that k, m, l, ϵ can be functions of n.)*

An explicit construction of an (l,ϵ)-extractor is an algorithm that is an (l,ϵ)-extractor for every input length n.

[4] For a discussion on extractors, see [Sha02].

Explicit construction of extractors has received huge attention in the last years. Ta-Shma ([TS96]) was the first to present explicit construction of extractors with seed length that is polylogarithmic in n. Recent constructions of explicit extractors improve over his results, with the current best parameters achieved by the construction of [LRVW03].

[RRV02] presented an (l, ϵ)-extractor for any l, ϵ, with seed length $k = polylog(\frac{n}{\epsilon})$, and $m = l + k - 2\log(\frac{1}{\epsilon}) - O(1)$ output bits. That is, if the input has min-entropy of at least l, and the seed is truly random, the extractor guarantees that the m output bits are ϵ-close to U_m.

Let $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be an (l, ϵ)-extractor, with $b_X \geq l$, and $m = b_X + c \cdot (b_X + b_Y - n)$ for $c < 1/3$. Let X, Y be the weak random sources, and let $Z = RS_k(X, Y)$ be the bits extracted in the first step of our algorithm. Using corollary 16, we get $|Z| = c \cdot (b_X + b_Y - n) = \Omega(polylog(n))$ and $\mathrm{Bias}(Z) = \delta = 2^{-\Omega(b_X + b_Y - n)}$, even conditioned on the distribution of X. Then $E(X, Z)$ is $(\epsilon + \delta)$-close to U_m, and $(\epsilon + \delta) = 2^{-\Omega(b_X + b_Y - n)}$.

□

6 Open Problems

When X, Y are random variables over $\{0,1\}^n$, we showed how to extract at least half the randomness from the input sources, when $H_\infty(X) + H_\infty(Y) = n + \Omega(polylog(n))$. The following open problems come to mind:

- Constructing an explicit function that extracts almost $b_X + b_Y$ random bits, when $H_\infty(X) + H_\infty(Y) \gg n$.
- Constructing an explicit function that can extract *even a single random bit* when $H_\infty(X) + H_\infty(Y) < n$.
- Constructing an explicit function that outputs even a single bit that is not constant, for all X, Y such that $H_\infty(X) = H_\infty(Y) = \varepsilon \cdot n$ for $\varepsilon < \frac{1}{2}$.

  This problem is analogues to a combinatorial open problem on bipartite Ramsey graphs; Denote $N = 2^n$, and consider a bipartite graph $G = (U, V, E)$, where $|U| = |V| = N$, and let F be the adjacency matrix of G, i.e. $F_{i,j} = -1$ iff $(i,j) \in E$.

  Such two sources X, Y induce a submatrix of size $N^\varepsilon \times N^\varepsilon$ of F. F contains a mono-chromatic submatrix of size $N^\varepsilon \times N^\varepsilon$ iff G or it's complement contain a copy of $K_{N^\varepsilon \times N^\varepsilon}$.

  Thus this problem is equivalent to finding an explicit construction of a bipartite graph G with N vertices on each side, such that both G and $\vec{G}$ do not contain a clique of size $N^\varepsilon \times N^\varepsilon$. This would imply that the $(K_{N^\varepsilon \times N^\varepsilon}, K_{N^\varepsilon \times N^\varepsilon})$-Ramsey number is greater than N.

References

[AS91] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley, New York, first edition, 1991.

[Blu84] M. Blum. Independent Unbiased Coin Flips From a Correlated Biased Source: a Finite State Markov Chain. In *IEEE Symposium on the Foundations of Computer Science*, number 25, 1984.

[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.

[CW89] A. Cohen and A. Wigderson. Dispersers, Deterministic Amplification and Weak random Sources. In *IEEE Symposium on Foundations of Computer Science*, number 30, pages 14–19, 1989.

[DO] Y. Dodis and R. Oliveira. On Extracting Private Randomness over a Public Channel.

[Eli72] P. Elias. The Efficient Construction of an Unbiased Random Sequence. *Ann. Math. Statist.*, 43(3):865–870, 1972.

[Gol02] O. Goldreich. Randomized Methods in Computation. Lecture Notes, 2002.

[LRVW03] C. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: optimal up to constant factors. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 602–611. ACM Press, 2003.

[Mur70] H.F. Murry. A General Approach for Generating Natural Random Variables. In *IEEE Transactions on Computers*, volume C-19, pages 1210–1213, 1970.

[Nis96] N. Nisan. Extracting randomness: How and why: A survey. In *In Proceedings, Eleventh Annual IEEE Conference on Computational Complexity*, pages 44–58. IEEE Computer Society Press., IEEE Computer Society Press., 24-27 May 1996.

[NN90] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. In *ACM Symposium on Theory of Computing*, pages 213–223, 1990.

[NTS99] N. Nisan and A. Ta-Shma. Extracting Randomness: A Survey and New Constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

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

[Per92] Y. Peres. Iterating Von Neumann's Procedude for Extracting Random Bits. *Ann. Math. Statist.*, 20(1):590–597, March 1992.

[Rit] T. Ritter. Random Number Machines: A Literature Survey. In *http://www.ciphersbyritter.com/RES/RNGMACH.HTM*.

[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.

[Sha02] R. Shaltiel. Recent developments in explicit constructions of extractors. BEATCS Computational Complexity Column, June 2002.

[SV86] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources (extended abstract). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 276–285. ACM Press, 1996.

[Vaz85] U. V. Vazirani. Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 366–378. ACM Press, 1985.

[Vaz87a] U. V. Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7(4):375–392, 1987.

[Vaz87b] U.V. Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 160–168. ACM Press, 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 Semi-Random Polynomial Time. In *IEEE Symposium on the Foundations of Computer Science*, number 26, 1985.