Improved Randomness Extraction from Two Independent Sources

Yevgeniy Dodis[1] ⋆, Ariel Elbaz[2] ⋆⋆, Roberto Oliveira[3] ⋆⋆⋆, and Ran Raz[4] †

[1] Department of Computer Science
New York University
dodis@cs.nyu.edu

[2] Department of Computer Science
Columbia University
arielbaz@cs.columbia.edu

[3] Department of Mathematics
New York University
oliveira@cims.nyu.edu

[4] Department of Computer Science
Weizmann Institute
ran.raz@weizmann.ac.il

Abstract. Given two independent weak random sources X, Y, with the same length ℓ and min-entropies b_X, b_Y whose sum is greater than $\ell + \Omega(\mathsf{polylog}(\ell/\varepsilon))$, we construct a deterministic two-source extractor (aka "blender") that extracts $\max(b_X, b_Y) + (b_X + b_Y - \ell - 4\log(1/\varepsilon))$ bits which are ε-close to uniform. In contrast, best previously published construction [4] extracted at most $\frac{1}{2}(b_X + b_Y - \ell - 2\log(1/\varepsilon))$ bits. Our main technical tool is a construction of a strong two-source extractor that extracts $(b_X + b_Y - \ell) - 2\log(1/\varepsilon)$ bits which are ε-close to being uniform and *independent* of one of the sources (aka "strong blender"), so that they can later be reused as a seed to a seeded extractor. Our strong two-source extractor construction improves the best previously published construction of such strong blenders [7] by a factor of 2, applies to more sources X and Y, and is considerably simpler than the latter. Our methodology also unifies several of the previous two-source extractor constructions from the literature.

1 Introduction

Imperfect Randomness. Randomization has proved to be extremely useful and fundamental in many areas of computer science. Unfortunately, in many situations one does not have ideal sources of randomness, and therefore has to base a given application on *imperfect sources of randomness*.

⋆ Partially supported by the NSF CAREER and Trusted Computing Awards.
⋆⋆ Work supported by ISF grant.
⋆⋆⋆ Funded by a doctoral fellowship from CNPq, Brazil.
† Work supported by ISF grant.

Among many imperfect sources considered so far, perhaps the most general and realistic source is the *weak source* [29, 4]. The only thing guaranteed about a weak source is that no string (of some given length ℓ) occurs with probability more than 2^{-b}, where b is the so-called *min-entropy* of the source. We will call this source (ℓ, b)-weak. Unfortunately, handling such weak sources is often necessary in many applications, as it is typically hard to assume much structure on the source beside the fact that it contains some randomness. Thus, by now a universal goal in basing some application on imperfect sources is to make it work with weak sources. The most direct way of utilizing weak sources would be to extract nearly perfect randomness from such a source. Unfortunately, it is trivial to see [4] that no deterministic function can extract even one random bit from a weak source, as long as $b \leq \ell - 1$ (i.e., the source is not almost random to begin with). This observation leaves two possible options. First, one can try to use weak sources for a given application without an intermediate step of extracting randomness from it. Second, one can try designing probabilistic extractors, and later justify where and how one can obtain the additional randomness needed for extraction.

USING A SINGLE WEAK SOURCE. A big and successful line of research [26, 24, 4, 5, 29, 2] following the first approach showed that a single weak source is sufficient to simulate any probabilistic computation of decision or optimization problems (i.e., problems with a "correct" output which are potentially solved more efficiently using randomization; That is, all problems in BPP). Unfortunately, most of the methods in this area are not applicable for applications of randomness, where the randomness is needed by the application itself, and not mainly for the purposes of efficiency. One prime example of this is cryptography. For example, secret keys have to be random, and many cryptographic primitives (such as public-key encryption) *must* be probabilistic. The problem of basing a cryptographic protocol on a single weak random source has only been studied in the setting of information-theoretic symmetric-key cryptography. In this scenario, the shared secret key between the sender and the recipient is no longer random, but comes from a weak source. As a very negative result, McInnes and Pinkas [12] proved that one cannot securely encrypt even a single bit, even when using an "almost random" $(\ell, \ell - 1)$-weak source. Thus, one cannot base symmetric-key encryption on weak sources. Dodis and Spencer [6] also consider the question of message authentication and show that one cannot (non-interactively) authenticate even one bit using $(\ell, \ell/2)$-weak source (this bound is tight as Maurer and Wolf [11] showed how to authenticate up to $\ell/2$ bits when $b > \ell/2$).

USING SEVERAL WEAK SOURCES. Instead, we will assume that we have several weak sources, each one independent from all the other weak sources. Specifically, we will try to extract nearly ideal randomness from two weak sources.

The question of extracting randomness from two or more independent random sources originated in the works of Sántha and Vazirani [20, 25, 27]. Chor and Goldreich [4] were the first to consider general weak sources of equal block length; let us say that the sources X and Y are (ℓ_X, b_X)-weak and (ℓ_Y, b_Y)-weak, and for simplicity we also assume $\ell_X = \ell_Y = \ell$. It is known [7] that in this setting it is possible to extract nearly all the entropy (i.e., nearly $b_X + b_Y - 2\log(1/\varepsilon)$

bits) from the two sources, where ε is the required distance from the uniform distribution. However, best known explicit constructions achieve much weaker parameters. Specifically, for one bit extractors Chor and Goldreich showed that the inner product function works provided $b_X + b_Y > \ell + 2\log(1/\varepsilon)$. The best multi-bit two-source extractor ([4], thm 14) is capable of extracting almost $\frac{1}{2}(b_X + b_Y - \ell - 2\log(1/\varepsilon))$ random bits, but [4] only show how to *efficiently* implement this extractor when $b_X + b_Y > 1.75\ell$. It was also known how to extract a non-constant bit (or a few bits) from two sources when $b_X = \ell(\frac{1}{2} + o(1)),\ b_Y = O(\log(\ell))$ [9, 1], and this method can also be used to extract unbiased bits from such sources. In a setting of more than two weak sources, Barak et al. [3] recently show how to extract randomness from $(1/\delta)^{O(1)}$ independent $(\ell, \delta\ell)$-weak sources (for any $\delta > 0$), but this improvement does not apply to the case of two sources considered in this work.

Motivated by cryptographic applications, Dodis and Oliveira [7] recently considered the problem of extracting random bits from two independent sources which are (essentially) *independent* from one of the two sources (called the "seed"). They termed this two-source extractor a *strong blender*, since it is a common generalization of a two-source extractor (which they simply termed "blender") and a *strong extractor* [16], where the seed source is assumed to be public and truly uniform (and the objective of the latter is to make the seed as short as possible). They showed the existence of such strong blenders, whenever $b_X, b_Y > \log \ell + 2\log(1/\varepsilon)$, which are capable of extracting $(b_X - 2\log(1/\varepsilon))$ bits from X which are independent from the (ℓ, b_Y)-weak seed Y. On a constructive side, they showed that the constructions from [4] for two-source extractors (i.e., "blenders") and the related work of Vazirani [27] for the so-called SV sources [20] do extend to give strong blenders. The best such construction in [7] (based on a similar construction from [4]) is capable of extracting $\frac{1}{2}(b_X + b_Y - \ell - 2\log(1/\varepsilon))$ bits independent of one of the sources, but this is again efficient only when $b_X + b_Y > 1.75\ell$.

To summarize, while randomness extraction from two independent sources is possible, the known constructions and parameters seem far from optimal. In contrast, there are many efficient constructions of one-source randomness extractors (with truly random seeds) with nearly optimal parameters (see [10, 17, 21] and the references therein). Our work gives an improvement in the number of extracted bits, for the case $b_X + b_Y > \ell$, allowing for the first time the extraction of more than half the total randomness that is present in the sources.

Our Results. We give a simple two-source extractor capable of extracting $\max(b_X, b_Y) + (b_X + b_Y - \ell - 4\log(1/\varepsilon))$ bits whenever the sum of min-entropies is slightly greater than ℓ: $b_X + b_Y \geq \ell + \Omega(\mathsf{polylog}(\ell/\varepsilon))$. Our construction is based on two observations. First, a strong blender with good parameters can give a two-source extractor (i.e., a "regular blender") with even better parameters, when combined with a seeded extractor (with good parameters). Specifically, if a strong blender extracts $k \geq \Omega(\mathsf{polylog}(\ell/\varepsilon))$ nearly random bits Z which are independent from the weak seed Y, we can now apply a seeded extractor to Y, using the just extracted Z as the seed (since Z is "long enough" to be used with state-of-the-art extractors such as [17]). This allows us to extract a total of $(k+b_Y-2\log(1/\varepsilon))$ bits, which could be very high. While this obvious observation would already improve the state-of-the-art in the problem of two-source extraction, even when combined with the known strong blender constructions from [7], our second observation is a much simpler (and better) construction of strong blenders. Specifically, we show how to extract $(b_X+b_Y-\ell-2\log(1/\varepsilon))$ bits which are independent from one of the sources. Thus our strong blender construction: (1) improves by a factor of two the best previously published construction of [7]; (2) is very simple and efficient without any limitations on b_X and b_Y (as long as $b_X+b_Y \geq \ell+2\log(1/\varepsilon)$); and (3) uses only trivial properties from linear algebra in its analysis. By choosing the initial "seed" source to be the one containing more min-entropy, we get our final two-source extractor.

Our Strong Blender. In fact, we give the following general technique for constructing our strong blender. Assume we wish to extract k nearly random bits from X using Y as a seed. Let $A_1, \ldots, A_k$ be some $\ell \times \ell$ matrices over $GF[2]$, specified later. View X and Y as ℓ-bit vectors, and output bits $((A_1X) \cdot Y, \ldots, (A_kX) \cdot Y)$, where $\cdot$ denotes the inner product modulo 2, and A_iX is the matrix-vector multiplication over $GF[2]$. Of course, the main question is which condition on matrices $A_1 \ldots A_k$ would allow us to maximize the value of k. As we show, all we need is to have every non-empty sum $A_S = \sum_{i\in S} A_i$ (where $S \neq \emptyset$) of these matrices to have "high" rank. In particular, by using a known linear algebra construction [14, 13, 19]) of such matrices, we can ensure that all the needed matrices A_S will have full rank, which will in turn give us the desired value $k = (b_X+b_Y-\ell-2\log(1/\varepsilon))$. However, we also notice that by using several other simple, but sub-optimal choices of matrices $A_1 \ldots A_k$, we will obtain two of the three constructions from [7] as special cases, as well as several other constructions which appeared in the literature for related problems (i.e., [27]). Thus, our methodology elegantly unifies several of the previous approaches into one elementary framework.

2 Preliminaries

2.1 Basic notation

We mostly employ standard notation. The symbol log is reserved for the base 2 logarithm. For a positive integer t, U_t denotes a random variable that is uniform over $\{0,1\}^t$ and independent of all other random variables under consideration. We also write $[t] \equiv \{1, 2, \ldots t\}$. For two random variables A, B taking values in the finite set $\mathcal{A}$, their *statistical distance* is $\|A - B\|_s \equiv \frac{1}{2}\sum_{a\in\mathcal{A}} \left|\Pr[A=a] - \Pr[B=a]\right|$, and the min-entropy of A is $H_\infty(A) \equiv \min_{a\in\mathcal{A}}\big(-\log(\Pr[A=a])\big)$. For a random variable A taking values in $\{0,1\}^t$, the *bias* of A is its statistical distance from U_t, $\|A - U_t\|_s$. The L_2 norm of a vector $v \in \mathcal{R}^t$ is $\|v\|_2 \equiv \sqrt{\sum_{i=1}^t (v_i)^2}$. For two strings s, t, their concatenation is denoted $\big(s\ ,\ t\big)$. For a vector $a \in \{0,1\}^t$ and $1 \leq i \leq t$, we say that $i \in a$ if $a_i = 1$. Also recall that an (ℓ, b)-source denotes some random variable X over $\{0,1\}^\ell$ with min-entropy $H_\infty(X) \geq b$.

2.2 Strong Blenders and Extractors

We start by defining the notion of a randomness extractor [16].

Definition 1 ([16]). *Let $b \geq 0$, $\varepsilon > 0$. A (b, ε)-extractor* $\textsc{Ext} : \{0,1\}^\ell \times \{0,1\}^k \to \{0,1\}^m$ *is an efficient function such that for all (ℓ, b)-sources Y, we have*

$$\| \ \textsc{Ext}(Y,\ U_k) \ - \ U_m \ \|_s \leq \varepsilon$$

Clearly, one of the main objectives in the area of building explicit extractors is to minimize the seed length k while still extracting nearly all the randomness from the source. Many nearly optimal constructions exist by now (see [15, 23, 10, 17, 22, 21] and the references therein). While none of them clearly beats the rest in all parameters, it is known (e.g., [17]) how to extract $m = k + b - 2\log(1/\varepsilon)$ nearly random bits (which is optimal) using a seed of length $k = O(\mathsf{polylog}(\ell/\varepsilon))$.

In this work, however, we are interested in (strong) randomness extraction from two weak sources. Correspondingly, we define the notion of two-source extractors (aka "blenders") and strong blenders. For simplicity, we assume both sources have the same length ℓ throughout the paper.

Definition 2. *A (b_X, b_Y, ε)-strong blender (*SB*) is an efficient function* $\textsc{Ble} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^k$ *such that for all (ℓ, b_X)-weak sources X and all (ℓ, b_Y)-weak sources Y, we have*

$$\left\| \ \big(Y\ ,\ \textsc{Ble}(X,Y)\big) \ - \ \big(Y\ ,\ U_k\big) \ \right\|_s \leq \varepsilon$$

A (b_X, b_Y, ε)-two-source extractor is an efficient function $\textsc{Two-Ext} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^m$ *such that for all (ℓ, b_X)-weak sources X and all (ℓ, b_Y)-weak sources Y, we have*

$$\|\textsc{Two-Ext}(X,Y) \ - \ U_m\|_s \ \leq \ \varepsilon$$

We now observe the following relation among these three primitives which follows immediately from the triangle inequality: namely, given X, Y, the output of a strong blender $\textsc{Ble}(X,Y)$ can be used as a random seed for an extractor $\textsc{Ext}$, to extract the randomness from Y.

Lemma 1. *Let* $\textsc{Ble} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^k$ *be a (b_X, b_Y, ε)-strong blender and* $\textsc{Ext} : \{0,1\}^\ell \times \{0,1\}^k \to \{0,1\}^m$ *be a (b_Y, ε)-extractor. Then the following function* $\textsc{Two-Ext} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^m$ *is a $(b_X, b_Y, 2\varepsilon)$-two-source extractor:*

$$\textsc{Two-Ext}(X,Y) \ = \ \textsc{Ext}(\ Y,\ \textsc{Ble}(X,\ Y\ )\ )$$

Our final two-source extractor will follow by applying the Lemma above to the strong blender which we construct in the next section (using any good seeded extractor, such as the one in [17]). Thus, from now on we concentrate on improved constructions of strong blenders.

3 Efficient Strong Blender Constructions

In this section we show a construction of an efficient strong blender. In 3.1 we give a general technique for constructing a strong blender, that requires a set of $\ell \times \ell$ matrices over $GF[2]$ with the property that the sum of every subset of matrices has high rank. In section 3.2 we give an explicit set of matrices (that appeared in [14, 13, 19]), with the property that the sum of any subset of matrices has full rank. This gives the best possible result for our technique. In the same section we also review some previous constructions for two-source randomness extraction as special cases of our technique.

3.1 General Technique for Constructing Strong Blenders

We now show how to extract many random bits from two weak sources X, Y of the same length ℓ, such that the bits are random even if one sees Y.
Let $A_1, \ldots, A_k$ be $\ell \times \ell$ matrices over $GF[2]$, such that for every nonempty subset $S \subseteq [k]$, the rank of $A_S \stackrel{\text{def}}{=} \sum_{i \in S} A_i$ is at least $\ell - r$, for $0 \leq r < \ell$ (we will want to have high rank and keep r as small as possible).

The proposed strong blender is

$$\begin{aligned} \text{BLE}_A : \{0,1\}^\ell \times \{0,1\}^\ell &\rightarrow \{0,1\}^k \\ (x, y) &\longmapsto \Big((A_1 x) \cdot y, \ldots, (A_k x) \cdot y\Big) \end{aligned} \tag{1}$$

where $\cdot$ is the inner product mod 2 and $A_i x$ is a matrix-vector multiplication over $GF[2]$.

Theorem 1. *The function* BLE_A *is a* (b_X, b_Y, ϵ)-SB *with* $\log \frac{1}{\epsilon} = \frac{b_X + b_Y + 2 - (\ell + r + k)}{2}$, *that is*

$$\left\| \big(Y \ , \ \text{BLE}_A(X, Y)\big) \ - \ \big(Y \ , \ U_k\big) \right\|_s \leq 2^{-\frac{b_X + b_Y + 2 - (\ell + r + k)}{2}}$$

Our main tool for proving theorem 1 is the Parity lemma (also known as the XOR lemma, see [8, 27]), which relates the bias of a k-bit random variable T with the sum of squared biases of every subset of the bits in T.

Lemma 2 (Parity lemma [8]). *For any k-bit random variable T,* $\|T \ - \ U_k\|_s$ *is upper bounded by*

$$\sqrt{\sum_{0^k \neq a \in \{0,1\}^k} \|T \cdot a \ - \ U_1\|_s^2}$$

where $T \cdot a$ is the inner product (mod 2) of T and a.

We also use a relation between the rank of a matrix L and the min-entropy of LX.

Lemma 3. *Let L be an $\ell \times \ell$ matrix over $GF[2]$. If the rank of L is $\ell - r$, then for any random variable X over* $\{0,1\}^\ell$, $\quad H_\infty(LX) \geq H_\infty(X) - r$.

Proof. Note that for any $z \in \{0,1\}^\ell$, there are at most 2^r different values $x \in \{0,1\}^\ell$ such that $Lx = z$. Thus, $\max_{z\in\{0,1\}^\ell} \Pr[LX = z] \leq 2^r \cdot \max_{x\in\{0,1\}^\ell} \Pr[X = x]$.

Proof. [of Theorem 1] Following [4, 7], it is sufficient to consider the case when Y is uniformly distributed on some set S_Y, $|S_Y| = 2^{b_Y}$, and X is uniformly distributed on some set S_X, $|S_X| = 2^{b_X}$.

$$\begin{aligned}
\|(Y\ ,\ \mathrm{BLE}_A(X,Y)) - (Y\ ,\ U_k)\| &= \frac{1}{|S_Y|}\sum_{y\in S_Y} \|\mathrm{BLE}_A(X,y) - U_k\|_s \\
\text{(by parity lemma)} \qquad &\leq \sum_{y\in S_Y} \frac{1}{|S_Y|}\sqrt{\sum_{0^k\neq a\in\{0,1\}^k} (\|\mathrm{BLE}_A(X,y)\cdot a - U_1\|_s)^2} \\
\text{(by concavity)} \qquad &\leq \sqrt{\sum_{0^k\neq a\in\{0,1\}^k}\left[\frac{1}{|S_Y|}\sum_{y\in S_Y}(\|\mathrm{BLE}_A(X,y)\cdot a - U_1\|_s)^2\right]}
\end{aligned}$$

We now proceed to upper bound each of the (above) bracketed terms by $2^{-(b_X+b_Y+2-\ell-r)}$; the final result then follows from simple addition. Fixing a value of $a = a_1a_2\ldots a_k \in \{0,1\}^k \backslash \{0^k\}$, and identifying a with the set of all $i \in [k]$ for which $a_i = 1$, we have

$$\mathrm{BLE}_A(X,y)\cdot a = \sum_{i\in a}\mathrm{BLE}_A(X,y)_i = \sum_{i\in a}(A_iX)\cdot y = \left(\Big(\sum_{i\in a}A_i\Big)X\right)\cdot y$$

where all sums are taken modulo 2.

Now define $A_a \stackrel{\text{def}}{=} \sum_{i\in a} A_i \pmod 2$, and recall that the rank of A_a is at least $\ell - r$. As a proof tool, we introduce the $2^\ell \times 2^\ell$ matrix M whose rows and columns are labeled by the elements of $\{0,1\}^\ell$, and whose (x,y)th entry is $M_{x,y} = (-1)^{(A_ax)\cdot y}$ (for $x, y \in \{0,1\}^\ell$). We also let $\overrightarrow{M}_x$ be the xth row of M. The following two properties are the key to what follows.

1. *Every row of M is equal to at most 2^r other rows of M.* Indeed, if $x, x' \in \{0,1\}^\ell$ satisfy $\overrightarrow{M}_x = \overrightarrow{M}_{x'}$, then it must be that $A_ax = A_ax' \pmod 2$, i.e. $x + x' \pmod 2$ is in the kernel of A_a. Since the rank of A_a is at least $\ell - r$, the kernel of A_a has at most 2^r elements.
2. *Rows of M that are distinct are orthogonal (in the Euclidean inner product sense).* For if $x \neq x' \in \{0,1\}^\ell$ are such that $\overrightarrow{M}_x \neq \overrightarrow{M}_{x'}$, then $z \stackrel{\text{def}}{=} A_a(x + x') \neq 0^\ell \pmod 2$, and the $\mathcal{R}$-valued Euclidean inner product of $\overrightarrow{M}_x$ and $\overrightarrow{M}_{x'} = (\overrightarrow{M}_x)^T(\overrightarrow{M}_{x'}) = \sum_{y\in\{0,1\}^\ell}(-1)^{z\cdot y \pmod 2}$ is 0.

Now let $\overrightarrow{D} \in \mathcal{R}^{2^\ell}$ be the vector defined by $\overrightarrow{D} \stackrel{\text{def}}{=} \frac{1}{2|S_X|}\sum_{x\in S_X}\overrightarrow{M}_x$. It is not hard to see that the yth coordinate D_y of $\overrightarrow{D}$ is equal to $\frac{1}{2}(\Pr_{x\in S_X}[M_{x,y} =$

$1] - \Pr_{x \in S_X}[M_{x,y} = -1])$, so that $|D_y| = \|\text{BLE}_A(X, y) \cdot a - U_1\|_s$. The quantity we wish to bound is thus

$$\frac{1}{|S_Y|} \sum_{y \in S_Y} (\|\text{BLE}_A(X, y) \cdot a - U_1\|_s)^2 = \frac{1}{|S_Y|} \sum_{y \in S_Y} |D_y|^2 \le \frac{1}{|S_Y|} \left\|\overrightarrow{D}\right\|_2^2$$

and $\left\|\overrightarrow{D}\right\|_2^2$ equals the Euclidean inner product $\overrightarrow{D}^T \overrightarrow{D}$ over $\mathcal{R}$

$$\left\|\overrightarrow{D}\right\|_2^2 = \frac{1}{4|S_X|^2} \sum_{x, x' \in S_X} (\overrightarrow{M}_x)^T (\overrightarrow{M}_{x'})$$

But properties 1. and 2. above show that for any $x \in S_X$

$$\begin{aligned} \sum_{x' \in S_X} (\overrightarrow{M}_x)^T (\overrightarrow{M}_{x'}) &= \sum_{x' : \overrightarrow{M}_{x'} = \overrightarrow{M}_x} (\overrightarrow{M}_x)^T (\overrightarrow{M}_{x'}) + \sum_{x' : \overrightarrow{M}_{x'} \neq \overrightarrow{M}_x} (\overrightarrow{M}_x)^T (\overrightarrow{M}_{x'}) \\ &\le 2^r \left\|\overrightarrow{M}_x\right\|_2^2 \le 2^{r+\ell} \end{aligned}$$

Hence $\left\|\overrightarrow{D}\right\|_2^2 \le 2^{r+\ell}/4|S_X|$, and plugging this bound into (3.1) yields

$$\frac{1}{|S_Y|} \sum_{y \in S_Y} (\|\text{BLE}_A(X, y) \cdot a - U_1\|_s)^2 \le \frac{2^{r+\ell}}{4|S_X||S_Y|} = 2^{-(b_X + b_Y + 2 - (\ell + r))}$$

which, as noted above, implies the theorem.

We emphasize that the above technique is entirely general: *any* set of matrices $A_1, \ldots, A_k$ such that the sum of any non-empty subset of matrices has rank of at least $\ell - r$, can be used to extract $k < b_X + b_Y + 2 - \ell - r$ bits from ℓ-bit weak sources X, Y with respective min-entropies b_X, b_Y.

The following two cases are of special interest (of course, they apply only when r is "small enough"; that is, when $r + k < b_X + b_Y + 2 - \ell$):

1. If the min-entropies $b_X + b_Y$ sum up to $\ell + \mathsf{polylog}(\ell)$, we can extract polylogarithmic (in ℓ) number of bits with bias $2^{-\Omega(\mathsf{polylog}(\ell))}$.
2. If the min-entropies $b_X + b_Y$ sum up to $(1 + c)\ell$, for any constant $0 < c \le 1$, we can extract linear (in ℓ) number of bits with exponentially small bias.

3.2 Explicit Strong Blender Instantiations

Theorem 1 subsumes constructions introduced (sometimes implicitly) in previous works. These constructions are presented below, followed by our more randomness-efficient construction.

Identity Matrix. This is the simplest case when $k = 1$ and A_1 is the identity matrix. This gives $r = 0$ and implies that the inner product function is a strong blender with bias $\varepsilon = 2^{-\frac{(b_X+b_Y+1-\ell)}{2}}$, reproving the result from [7] (adapting the result from [4]).

CYCLIC SHIFT MATRICES. Vazirani [27] used cyclic shift matrices, to extract randomness from SV sources. When ℓ is a prime with 2 as a primitive root, the sum of any subset of matrices has rank at least $\ell-1$ (that is $r=1$, in our notation above). Formally, let A_i be a linear transformation matrix corresponding to a cyclic shift by $i-1$ bits; That is, the jth row of A_i has a 1 in the $j-i+1 \pmod{\ell}$ column and zero elsewhere. The required property of these matrices is given below.

Lemma 4 ([27]). *Let ℓ be a prime with 2 a primitive root modulo ℓ (i.e. 2 is a generator of Z_ℓ^*). Let $\overrightarrow{u} \in \{0,1\}^\ell \setminus \{0^\ell, 1^\ell\}$ be a vector (which is not the all 0's or the all 1's vector). Let A be an $\ell \times \ell$ matrix over $GF(2)$, such that the rows of A are the ℓ right-cyclic-shifts of $\overrightarrow{u}$. Then $rank(A) \geq \ell - 1$.*

By Theorem 1, a corresponding strong blender can extract k bits with bias $2^{-\frac{(b_X+b_Y+1-\ell-k)}{2}}$, or, equivalently, extract $k = (b_X + b_Y + 1 - \ell - 2\log(1/\varepsilon))$ bits with bias ε.

(NON-CYCLIC) RIGHT SHIFT MATRICES. Let $A_1, \ldots, A_k$ be linear transformation matrices such that A_i corresponds to the right shift by $i-1$ bits. For any non-empty subset $S \subseteq [k]$, it is easy to see that the rank of $A_S = \sum_{i\in S} A_i$ is at least $\ell - k + 1$. By theorem 1 one can build a strong blender from those matrices that extracts k bits with bias $2^{-\frac{(b_X+b_Y-\ell-2k)}{2}}$, or, equivalently, extract $k = (b_X + b_Y - \ell - 2\log(1/\varepsilon))/2$ bits with bias ε.

THE MATRICES FROM A GENERAL ERROR-CORRECTING CODE. The construction of [7] (adapted from [25]) also pertains to the present framework. Let C be an $[\ell, k, d]$ linear error correcting code (i.e., having dimension k and minimal distance d in $\{0,1\}^\ell$). Let $A_1, \ldots, A_k$ be diagonal matrices, the diagonal of A_i containing the coordinates of a codeword c_i encoding the i-th unit vector in $\{0,1\}^k$. By linearity, the diagonal of $A_S = \sum_{i\in S} A_i$ is also a codeword in C, which is non-zero when $S \neq \emptyset$, and thus has at least d ones, so that $rank(A_S) \geq d$. Then Theorem 1 applies with $r = \ell - d$, and the corresponding strong blender extracts k bits with bias $2^{-\frac{(b_X+b_Y+d-2\ell-k)}{2}}$, or, equivalently, extract $k = (b_X + b_Y + d - 2\ell - 2\log(1/\varepsilon))$ bits with bias ε (notice, sometimes this k may not be achievable due to the coding constraints).

OUR CONSTRUCTION. We now present our new construction of strong blenders via Theorem 1. We show that there is a set of matrices $A_1, \ldots A_\ell$ whose non-empty sums have *full rank*, thus achieving the best possible result for using the technique in section 3.1. This latter fact follows from [14, 13, 19], and we reproduce its simple proof for completeness. We use the isomorphism $\{0,1\}^\ell \approx GF[2^\ell]$. Take any basis $x_1, x_2, \ldots, x_\ell$ of $GF[2^\ell]$. (There are many such bases for $GF[2^\ell]$, since it is isomorphic to the ℓ-dimensional vector space over $GF[2]$. One such basis, which is easy to use, is $1, x, x^2, \ldots, x^{\ell-1}$, where x is any primitive element of $GF[2^\ell]$; that is, an element that is not the root of any polynomial of degree $< \ell$ over $GF[2^\ell]$.) The matrices $\{A_i\}_{i=1}^{\ell}$ are defined such that matrix A_i corresponds to left-multiplication by the basis-element x_i:

$$A_i : y \in GF[2^\ell] \mapsto x_i y$$

Now for each non-empty set $S \subseteq [\ell]$, $x_S = \sum_{i \in S} x_i$ is a non-zero element in $GF[2^\ell]$, and therefore has an inverse x_S^{-1} in $GF[2^\ell]$. Let U_S be the $\ell \times \ell$ matrix over $GF[2]$ that corresponds to left-multiplication by x_S^{-1}:

$$U_S : y \in GF[2^\ell] \mapsto x_S^{-1} y$$

Now, it is easy to see that U_S is the matrix-inverse of $A_S = \sum_{i \in S} A_i$, since left-multiplication by matrix $U_S \sum_{i \in S} A_i$ corresponds to multiplication by the field identity element $x_S^{-1} \sum_{i \in S} x_i = 1$.

This immediately implies the desired conclusion:

Lemma 5. *Let $A_1, \ldots, A_\ell$ be matrices presented above. Then for all $\emptyset \neq S \subseteq [\ell]$, the rank of $A_S \stackrel{\text{def}}{=} \sum_{i \in S} A_i$ is ℓ.*

Plugging the first k of the matrices $A_1, \ldots, A_\ell$ into theorem 1 we deduce the existence of an *explicit* strong blender that can extract k bits with bias $2^{-\frac{(b_X + b_Y + 2 - \ell - k)}{2}}$, or, equivalently, extract $k = (b_X + b_Y + 2 - \ell - 2\log(1/\varepsilon))$ bits.

References

1. N. Alon. Tools from higher algebra. In *Handbook of Combinatorics*, R.L. Graham, M. Grtschel and L. Lovsz, eds, North Holland (1995), Chapter 32, pp. 1749-1783.
2. A. Andreev, A. Clementi, J. Rolim, L. Trevisan. Dispersers, deterministic amplification, and weak random sources. In *SIAM J. on Comput.*, 28(6):2103–2116, 1999.
3. B. Barak, R. Impagliazzo and A. Wigderson. Extracting Randomness from Few Independent Sources. *Manuscript*, 2004.
4. B. Chor, O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.
5. A. Cohen, A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proc. of FOCS*, pp. 14–19, 1989.
6. Y. Dodis, J. Spencer. On the (Non-)Universality of the One-Time Pad. In *Proc. of FOCS*, 2002.
7. Y. Dodis, R. Oliveira. On Extracting Private Randomness over a Public Channel. In *RANDOM-APPROX 252-263*, 2003.
8. O. Goldreich. Three XOR-Lemmas - An Exposition. *Electronic Colloquium on Computational Complexity (ECCC)*, 1995.
9. R.L. Graham, J.H. Spencer. A constructive solution to a tournament problem. In *Canad. Math. Bull.* 14, 45-48.
10. C. Lu, O. Reingold, S. Vadhan, A. Wigderson. Extractors: Optimal Up to Constant Factors. In *Proc. of STOC*, 2003.
11. U. Maurer, S. Wolf. Privacy Amplification Secure Against Active Adversaries. In *Proc. of CRYPTO*, Lecture Notes in Computer Science, Springer-Verlag, vol. 1294, pp. 307–321, 1997.

12. J. McInnes, B. Pinkas. On the Impossibility of Private Key Cryptography with Weakly Random Keys. In *Proc. of CRYPTO*, pp. 421–435, 1990.
13. R. Meshulam. Spaces of Hankel matrices over finite fields. *Linear Algebra and its Applications*, 218, 1995.
14. E. Mossel, A. Shpilka, L. Trevisan. On ϵ-biased Generators in NC^0. In *Proc. of FOCS*, 2003.
15. N. Nisan, A. Ta-Shma. Extracting Randomness: a survey and new constructions. In *JCSS*, 58(1):148–173, 1999.
16. N. Nisan, D. Zuckerman. Randomness is Linear in Space. In *JCSS*, 52(1):43–52, 1996.
17. R. Raz, O. Reingold, S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *Journal of Computer and System Sciences*, 2002.
18. L. Rónyai, L. Babai, M. Ganapathy. On the number of zero-patterns in a sequence of polynomials *Journal of the AMS*, 2002.
19. R. Roth. Maximum rank array codes and their application to crisscross error correction. *IEEE transactions on Information Theory*, 37, 1991.
20. M. Sántha, U. Vazirani. Generating Quasi-Random Sequences from Semi-Random Sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.
21. R. Shaltiel. Recent developments in Explicit Constructions of Extractors. *Bulletin of the EATCS*, 77:67–95, 2002.
22. R. Shaltiel, C. Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *Proceedings of FOCS 2001*, pp.648-657, IEEE Computer Society, 2001.
23. L. Trevisan. Construction of Extractors Using PseudoRandom Generators. In Proc. of STOC, pp. 141–148, 1999.
24. U. Vazirani. Randomness, Adversaries and Computation. *PhD Thesis*, University of California, Berkeley, 1986.
25. U. Vazirani. Strong Communication Complexity or Generating Quasi-Random Sequences from Two Communicating Semi-Random Sources. *Combinatorica*, 7(4):375–392, 1987.
26. U. Vazirani, V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proc. of 26th FOCS*, pp. 417–428, 1985.
27. U. Vazirani. Efficiency Considerations in using semi-random sources. In *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pp. 160–168, 1987.
28. A. Wigderson. Open problems. Notes from *DIMACS Workshop on Pseudorandomness and Explicit Combinatorial Constructions*, 1999
29. D. Zuckerman. Simulating BPP Using a General Weak Random Source. *Algorithmica*, 16(4/5):367-391, 1996.