2-Source Extractors Under Computational Assumptions and Cryptography with Defective Randomness*

Yael Tauman Kalai† Xin Li‡ Anup Rao §

May 8, 2009

1 Context

Randomness is a useful resource for solving many problems in computer science. The goal of the broad area of *derandomization* is to weaken the assumptions that are placed on the randomness used. One way to do this is to design randomness extractors: these are algorithms that take randomness that comes from defective sources and extract truly random bits from them. A 2-Source extractor is an algorithm that extracts random bits from two independent sources, each giving bits with some entropy. The best known extractor algorithms for this situation require that at least one of the sources has 0.499 entropy rate, even though the probabilistic method shows that a random bit can be extracted from two independent sources each of which gives n bits with only $\log n + O(1)$ bits of entropy. Past work in derandomization has left us with the dramatic result that any randomized algorithm can be simulated by an algorithm that only has access to a *single* source of randomness with some entropy. However, even though randomness is used in an essential way in cryptography and distributed computing, there was no analogous result for these domains. Moreover, there is a strong negative result showing that many cryptographic tasks are impossible given only a single weak random source. Nevertheless, it was later shown how to use a variant of the DDH assumption to do secure multiparty computation, where each party has only access to an *independent* weak source of randomness. Unfortunately, this result requires that the number of participants in the protocol is super constant, and depends on the security parameter.

2 Our Contribution

In this work, we assume that there exists a *one way permutation for weak sources*, f. This means that for any distribution $X \in \{0,1\}^n$ with $\Omega(n)$ min-entropy, a circuit of size $\text{poly}(n^{\log n})$ cannot invert $f(X)$, except with negligible probability. A concrete candidate may be the exponentiation function, over an appropriate group (i.e. we might hope that computing discrete log is hard, even on weak sources). Using any such permutation f, we show how to build a 2-source extractor that only requires the sources to have entropy $\Omega(n)$.[1] Our conclusion is an information theoretic object, even though our assumption is a computational one.

We use this extractor (and f again) to design a *computational network extractor protocol*. This is a protocol that allows a collection of processors, each of which has access to a *single* independent source with

*In this note, we omit most references, which can be found in the introduction of the paper.

†Microsoft Research, yael@microsoft.com.

‡University of Texas at Austin, lixints@cs.utexas.edu. Supported in part by NSF Grant CCF-0634811 and THECB ARP Grant 003658-0113-2007.

§Institute for Advanced Study, arao@ias.edu. Supported in part by NSF Grants CCF-0634811 and CCR-0324906. Part of this work was done while the author was visiting Microsoft Research New England.

[1]Our extractor can also extract randomness if only one of the sources has entropy $\Omega(n)$ and the other has only entropy k. However, then we need to assume that it is hard to invert $f(X)$, where X is any distribution with entropy k.

$\Omega(n)$ entropy, to extract bits which are computationally indistinguishable from being uniform and private. Our protocol is guaranteed to succeed as long as at least 2 of the participants are honest. We also construct a computational network extractor protocol for the case that each processor has access to an independent source with only n^{ϵ} entropy, which succeeds as long as there are at least u honest participants, where u is a constant that depends on ϵ.

The key observation used in our work is an idea that was implicitly used in the hardcore predicate construction of Goldreich-Levin — a one way function can be used to generate distributions which are computationally indistinguishable from being independent, even though they are very correlated in reality. If R, X are uniform, Goldreich-Levin showed that $(\langle X, R\rangle, f(X), R)$ look like three independent distributions to any small circuit. This idea generalizes[2]: if X is a weak source with some entropy, R is independent and uniform, and Ext is a *reconstructive extractor*, then the following two distributions are computationally indistinguishable:

$$(\mathsf{Ext}(X, R), f(X), R) \approx (\mathsf{Uniform}, \mathsf{f(X)}, \mathsf{R}),$$

where here the three strings on the right are independent. Reconstructive extractors are a well studied object in extractor theory. Today we know of several explicit constructions, many of which only require $\mathsf{polylog}(n)$ bits in R.

Next we give an informal description of the ideas that are involved in building our 2-source extractor.

2.1 A Toy Version of our 2-Source Extractor

Let X, Y be the two independent sources, each which has entropy δn, from which we would like to extract random bits. In order to describe our key ideas, we shall make a simplifying assumption: we assume that for every δ, there is a constant $t(\delta)$ such that if $X = X_1, X_2, \ldots, X_t$ is broken into t blocks, there is a constant g such that for every fixing of $x_1, \ldots, x_{g-1}$, we have that $X_g|x_1, \ldots, x_{g-1}$ is uniform. Thus, the g'th block is uniform even conditioned on previous blocks. Although our assumption seems unreasonable at first, it turns out that there is a way to use past work on extractors (and a careful analysis) to convert two independent sources X, Y into sources that are close to having properties similar to what we assume.

Since one of the blocks in X is uniform, we might hope that the bitwise xor $X_1 \oplus \cdots \oplus X_t$ would be uniform. Of course this is not at all true, since the blocks following the good block g can depend on X_g. Still, we show how to use a one way permutation to make something like this work. Let Ext and f be as in our discussion above. We define

$$\mathsf{TExt}(x, y) \stackrel{def}{=} \mathsf{Ext}(f(y), x_1) \oplus \mathsf{Ext}(f^2(y), x_2) \oplus \cdots \oplus \mathsf{Ext}(f^t(y), x_t)$$

Here f^i denotes the permutation obtained by composing f with itself i times.

We shall argue that $\mathsf{TExt}(X, Y)$ is computationally indistinguishable from being uniform. Since computational indistinguishability is the same as statistical indistinguishability when the input is of size $O(\log n)$, we conclude that the first $O(\log n)$ bits of $\mathsf{TExt}(X, Y)$ are actually statistically indistinguishable from uniform.

Let $R_1(x, y)$ denote the string $\mathsf{Ext}(f(y), x_1) \oplus \mathsf{Ext}(f^2(y), x_2) \oplus \cdots \oplus \mathsf{Ext}(f^{g-1}(y), x_{g-1})$. Note that since $X_1, \ldots, X_{g-1}$ are fixed, $R_1(X, Y)$ is actually only a function of Y. We denote this string by $R_1(Y)$. Then it suffices to show that

$$(\mathsf{Ext}(f^g(Y), X_g), X, R_1(Y), f^{g+1}(Y)) \approx (\mathsf{Uniform}, X, R_1(Y), f^{g+1}(Y)), \tag{1}$$

since $\mathsf{TExt}(\mathsf{X}, \mathsf{Y})$ can be efficiently computed from $\mathsf{Ext}(f^g(Y), X_g), X, R_1(Y), f^{g+1}(Y)$. We take Ext with output length that is significantly shorter than the entropy of Y, and thus we can argue that for almost all $R_1(Y)$, Y has linear entropy even conditioned on the value of $R_1(Y)$. Let Y' denote the source obtained after such a fixing. Then we can use the properties of reconstructive extractors to argue that

$$(\mathsf{Ext}(f^g(Y'), X_g), X, f^{g+1}(Y')) \approx (\mathsf{Uniform}, X, f^{g+1}(Y')),$$

Since the above statement is true for Y' obtained by all but a negligible fraction of the fixings of $R_1(Y)$, this actually proves Equation (1).

[2]This observation was made in an earlier work of TaShma and Zuckerman.