On the (non)Universality of the One-Time Pad

Yevgeniy Dodis
Department of Computer Science
New York University
Email: dodis@cs.nyu.edu

Joel Spencer
Department of Computer Science
New York University
Email: spencer@cs.nyu.edu

Abstract

Randomization is vital in cryptography: secret keys should be randomly generated and most cryptographic primitives (e.g., encryption) must *be probabilistic. As a common abstraction, it is assumed that there is a source of* truly random *bits available to all the participants of the system. While convenient, this assumption is often highly unrealistic, and cryptographic systems have to be built based on* imperfect *sources of randomness. Remarkably, this fundamental problem has received little or no attention so far, despite the fact that a related question of simulating probabilistic (*BPP*) algorithms with imperfect random sources has a long and rich history.*

In this work we initiate the quantitative study concerning feasibility of building secure cryptographic *primitives using imperfect random sources. Specifically, we concentrate on symmetric-key encryption and message authentication, where the shared secret key comes from an imperfect random source instead of being assumed truly random. In each case, we compare the class of "cryptographic" sources for the task at hand with the classes of "extractable" and "simulatable" sources, where: (1) "cryptographic" refers to sources for which the corresponding symmetric-key primitive can be build; (2) "extractable" refers to a very narrow class of sources from which one can extract nearly perfect randomness; and (3) "simulatable" refers to a very general class of* weak *random sources which are known to suffice for* BPP *simulation. For both encryption and authentication, we show that the corresponding cryptographic sources lie strictly in between extractable and simulatable sources, which implies that "cryptographic usage" of randomness is more demanding than the corresponding "algorithmic usage", but still does not require perfect randomness. Interestingly, cryptographic sources for encryption and authentication are also quite different from each other, which suggests that there might not be an elegant way to describe imperfect sources sufficient for "general cryptographic use". We believe that our initial investigation in this new area will inspire a lot of further research.*

1 Imperfect Random Sources

Randomization has proved to be extremely useful and fundamental in many areas of computer science, such as approximation algorithms, counting problems, distributed computing, primality testing, as well as cryptographic protocols (which is the topic of this paper). The common abstraction used to introduce randomness into computation is that the underlying algorithm has access to a stream of completely unbiased and independent random bits. This abstraction allows one to use randomness in a clean way, separating out the issue of actually generating such "strong" random bits. Unfortunately, in reality we do not have sources that emit perfectly uniform and independent random bits. However, there are many sources whose outputs (which need not be bits) are believed to be "somewhat random". Such sources are generally called *imperfect random sources*. We remark that the "imperfectness" of the source does not only come from the fact that it does not generate uniform random bits, but also because the exact source distribution is usually *unknown*; instead, only some *property* the distribution is known (like no string is excessively likely, etc.), and our proposed usage of a given source should work for *any* distribution satisfying this property. Thus, "imperfect source" literally means "an unknown source from a given family of probability distributions".

A large amount of research has been devoted to filling in the gap between such realistic imperfect sources and the ideal sources of randomness that are actually used in designing various algorithms and protocols. As we will argue below, the current body of knowledge nevertheless leaves a large gap in understanding the usefulness of imperfect sources for various *cryptographic* purposes. Indeed, we can roughly separate the following two major questions that have been addressed so far in studying imperfect random sources, none of which directly dealing with cryptography:

- *Simulation:* can we efficiently simulate a probabilistic (BPP) algorithm with a given source?
- *Extraction:* can we extract almost perfect randomness from a given source?

The first question addresses the problem if a given source is acceptable for universal probabilistic computation of decision or optimization problems (i.e., problems with a unique "correct" output which are potentially solved more efficiently using randomization). The second question goes for a conceptually cleaner approach in trying to provide — when possible – a "compiler" for a given imperfect source. The complier first extracts almost perfect randomness from the source, which can then be used for *any* application originally designed to work with ideal random bits. Clearly, extraction from a given source is a very desirable property to have, since it solves a much broader problem that BPP simulation. For example, "extractable" sources can be used in any *cryptographic* application (like secure encryption), but not every "simulatable" source can [19] (see below). Unfortunately, as shown below, the set of extractable sources is also dramatically smaller than the set of simulatable sources.

SIMULATABLE SOURCES. It turns out that the class of simulatable sources is extremely large. In particular, more and more imperfect (so called "weak") random sources have been shown to be simulatable [32, 30, 8, 9, 35, 2], culminating in using extremely weak sources [2]. The only thing guaranteed about a weak source is that no particular string has a very high probability of occurring. This is characterized by a parameter ℓ (called the *min-entropy* of the source) by saying that no string (of some given length) occurs with probability more than $2^{-\ell}$ (for any distribution of the source). The optimal result of [2] then says that BPP simulation is possible for any N-bit weak source of min-entropy at least N^γ, for some (arbitrarily small) $\gamma > 0$. Interestingly, we will see that weak sources are typically far too general for any randomness extraction (e.g., none of the sources [32, 30, 8, 9, 35, 2] is extractable). Instead, the works above take advantage of the fact that even though it is impossible to generate almost random bits from the corresponding weak sources, it is possible to generate random strings, a majority of which avoid falling into the negligibly small set of "bad" strings. Running the given algorithm many times on various such pseudorandom strings and computing some statistics, a correct answer is given with high probability.

Unfortunately, most of the above methods are not applicable for cryptographic use, where the randomness is needed by the application *itself*, and not mainly for the purposes of efficiency. Indeed, McInnes and Pinkas [19] have shown that *none* of the simulatable sources above can be used to securely encrypt even a single bit! (See Section 2).

EXTRACTION FROM IMPERFECT SOURCES. As we will see, extraction is much harder to achieve than simulation, even for relatively "structured" imperfect random sources. In rough terms, we can separate three types of imperfect random sources considered so far: *streaming* sources, *bit-fixing* sources, and already mentioned *weak* sources (the latter being significantly more general than the former two).

STREAMING SOURCES. Like the ideal source, a streaming source produces a stream of bits incrementally over time, but these bits are not necessarily unbiased or independent (exact details depend on the streaming source considered). The first works [34, 12, 5] considered streaming sources which generated highly *independent* (but possibly biased) random bits. As a result, elegant techniques were developed to extract many *ideal* random bits from such highly "regular" sources. Unfortunately, once the strong independence requirement was relaxed, many impossibility results were obtained. The first quite striking negative result was obtained by Sántha and Vazirani [23], who demonstrated that not even a single almost random bit can be extracted if *every* bit of the source can be *slightly* biased and depend on all the previous bits. Lichtenstein et al. [18] showed a mix of positive and (mainly) negative results when *few* bits of the source could be arbitrarily biased while the rest were *truly* random. Dodis [10] showed even more negative results for the common generalization of the above two sources.

BIT-FIXING SOURCES. A bit-fixing source produces (at once) a string of N bits, some of which (say, b) are adversarially fixed, but the other $\ell = (N - b)$ are truly random. The goal of extraction for such sources is to design a function (called a *resilient* function) whose output is "close" to random no matter which b input bits are fixed. It turns out that there is a huge difference depending on whether the b "fixed" bits get set before or after the ℓ random bits are chosen. In the first scenario (studied by [31, 7, 3, 13, 17, 11]), quite positive and by now nearly optimal results are known for extracting *many* bits (one perfect bit is trivially extracted by the parity function). In particular, close to ℓ nearly perfect bits can be extracted in this setting [11]. In the second scenario (b fixed bits are set *after* the ℓ random bits), even *one* bit is hard to extract: the optimal b for this task lies somewhere between $\Omega(N/\log^2 N)$ [1] and $O(N/\log N)$ [16].

WEAK SOURCES. Originated by Chor and Goldreich [8], much subsequent research has been dedicated to various flavors of the so called *weak* random sources. Recall, a fixed distribution has min-entropy ℓ if no element can occur with probability more that $2^{-\ell}$. Generally, a min-entropy of a probability distribution is considered the right measure for the amount of "randomness" it contains. An imperfect source has min-entropy ℓ if all of its distributions have min-entropy ℓ, even though not all such distributions might belong to the source. On the other hand, a *weak* source of min-entropy ℓ is a specific source consisting of *all* distributions of min-entropy ℓ. In other words, if an application can tolerate a weak source, we are not making any extra assumptions about our distribution except that it contains "enough randomness". Thus, weak sources are the most general sources one can consider, since they contain all natural imperfect sources as special cases. Remarkably, we already mentioned that weak sources are still sufficient to simulate BPP algorithms. On the other hand, weak sources are also too general for any kind of randomness extraction (unless we make some relaxations; see below). For example, it is trivial to show (see formal proof in [8]) that every deterministic bit extraction function from an N-bit source can be fixed to a constant by a source of (huge) min-entropy $(N-1)$, implying that one cannot even extract a single slightly random bit from such a source!

Three kinds of relaxations were recently studied to surpass the strong impossibility result above. First, Trevisan and Vadhan [29] consider the problem of extraction from *efficiently samplable* distributions with a given min-entropy. Second, we mention a series of other works [23, 31, 8, 29] which extract randomness from several *independent* imperfect sources (which is a strong assumption). Last, but not the least, we mention a large body of work on the so called *randomness extractors* [21]. Such extractors are allowed to use a small number of *truly random bits* in addition to the output of a given imperfect source. Despite having many applications (see [20, 28, 22] and the references therein), the assumption about the existence of truly random bits is not applicable in many situations.

As the summary of the above discussion, useful imperfect sources have reasonably high level of min-entropy, and weak random sources are the most general and realistic such sources. While being simulatable, weak sources (and many other less general imperfect sources) are highly non-extractable.

2 Cryptographic Sources

The main objective of this work is to initiate the study of the class of imperfect random sources applicable for various *cryptographic* use, like achieving privacy or authenticity. Let us informally call such sources "cryptographic" (w.r.t. to the application at hand). As we already mentioned, the large body of work studying simulatable and extractable sources leaves a significant gap in understanding the usefulness of imperfect sources for cryptographic purposes. We believe that the understanding above will not only tell us to what extent cryptographic applications — where randomness is crucial — can tolerate imperfect randomness, but will also shed further light on the differences between cryptography and algorithms/complexity theory. In particular, the main outcome of this work will show that cryptographic sources seem to lie strictly in between simulatable (i.e., weak) and extractable sources. Moreover, cryptographic sources for different tasks are different from each other. This suggests that different cryptographic applications use randomness differently, and there might not be an elegant way to describe sources sufficient for "general cryptographic use".

OUR CRYPTOGRAPHIC APPLICATIONS. In this work we concentrate on studying *private-key* cryptography; namely, the applications of private-key encryption and message authentication. In both applications, Alice wants to send a message m to Bob over an insecure channel, controlled by an adversary Eve. Alice and Bob originally agree on a shared secret key K, and on the publicly known encoding and decoding functions E and D. To send the message m, Alice uses K to compute the ciphertext $c = E_K(m)$ and sends c over the channel. Bob gets the ciphertext (call it c') and outputs $m' = D_K(c')$, which could be either some message, or a special symbol $\perp$ (the latter indicates that c' was an invalid ciphertext). Clearly, if $c = c'$, then we require that $m = m'$. For encryption, we also want Eve to obtain "no information" about the message m upon observing the ciphertext c. Namely, Alice and Bob want to achieve privacy. For authentication, we do not want Eve to be able to change c to some c' such that Bob outputs, with "non-trivial" probability, a valid message $m' \notin \{m, \perp\}$. Namely, Alice wants to make sure that Eve cannot meaningfully change the message transmitted (of course, Eve can always block the message, but this is inevitable).

Aside from being interesting and important in their own right, there is one more advantage to start our general investigation from these applications. Specifically, it is well-known that both of them can be solved *information-theoretically*,[1] at least if the participants share a long enough *truly random* secret key K. In our scenario, we investigate what happens if this key instead comes from some imperfect source. Considering that most work on imperfect random sources is information-theoretic as well, studying the above applications seems to be the cleanest starting point for understanding "cryptographic" sources.

PRIVATE-KEY ENCRYPTION. Recall that information-theoretic security of (one-time) private-key encryption states that the encryptions of any two messages looks statistically indistinguishable to Eve, who does not know K.[2] And the encryption is *perfect* if these encryptions are identically distributed. Assuming that the key K is a truly random N-bit string, one can easily obtain a perfect encryption of an N-bit message m using the one-time pad scheme [33]: $c = E_K(m) = m \oplus K$, $m = D_K(c) = c \oplus K$, where $\oplus$ is the "exclusive OR" operator. Notice, c is uniformly distributed irrespective of m, so this encryption is indeed perfect. (Unfortunately, it can be used to encrypt only one

[1] This means no unproven computational assumptions, like the existence of one-way functions, are needed.

[2] Where the statistical difference is negligible in the security parameter.

message of length N securely, and Shannon [24] showed that *any* secure encryption scheme must have $|K| \geq |m|$.)

We now study what happens when K is not truly random, but comes from some imperfect source of randomness. The only work so far that has studied this question is that of McInnes and Pinkas [19]. This work shows that one cannot securely encrypt even a single bit with the weak random source (in, fact, even with a more restricted source of [23])! More precisely, there is no statistically secure encryption scheme for one-bit messages tolerating a weak N-bit source of min-entropy strictly less than N (say, $N-1$). In fact, for any "encryption" (E, D) of one-bit messages, some source of huge min-entropy $(N-2)$ makes the ciphertext c *completely reveal* the encrypted bit (see Appendix A for an alternative proof). Thus, the weak sources are not only non-extractable, but also highly non-cryptographic for private-key encryption.

On the other hand, the strong negative result of [19] leaves open — and in fact *suggests* — the possibility that *every cryptographic source for encryption is extractable*. If true, this would imply that *one-time pad is a universal one-time private-key encryption*.[3] Interestingly, the conjecture above is true for *a single* (possibly non-uniform) distribution on the shared key K. Indeed, Shannon's negative result generalizes to this case saying that the *Shannon's entropy* of the key, $H(K)$, under our fixed distribution has to be at least as large as the message length N: $H(K) \geq N$. On the other hand, it is well known that the (expected) number of almost truly random bits one can extract from a *single* distribution on K is again essentially equal to $H(K)$ (up to an additive 1). This shows that the one-time pad encryption is indeed universal for a single distribution on K: whenever it is possible to securely encrypt an N-bit message m (i.e., $H(K) \geq N$), one might as well extract from K an almost uniform $H(K)$-bit random string K', and then use K' as the one-time pad for m!

The main technical contribution of this work is a precise (negative) resolution of this conjecture for general imperfect sources. To state our optimal result quantitatively, recall that the *fairness* of one random bit r is defined to be $\min[\Pr(r=0), \Pr(r=1)]$. Thus, a truly random bit is $\frac{1}{2}$-fair, while a constant bit is 0-fair. We show that

Theorem 1 *For any fairness $\varepsilon > 2^{-N/2+1}$, there is an N-bit imperfect source S of min-entropy $\ell \geq N - \log(1/\varepsilon) - O(1)$ and a one-bit encryption scheme (E, D) such that:*

1. *(E, D) is* perfectly *secure for any distribution in S;*
2. *One cannot extract an ε-fair random bit from S.*

The lower bound on ℓ is optimal up to an additive constant, but ε can be made $2^{-N/2}$ when no restriction on ℓ is made.

[3] At least for the purposes of encrypting a single bit. Of course, there is a possibility that one can encrypt more bits "directly" rather than by first extracting uniform randomness and applying the one-time pad to it.

As a corollary, for any $\ell \leq N - \Omega(1)$, there exists a source S of min-entropy ℓ which is non-extractable but cryptographic (for one-bit encryption). Moreover, the impossibility of extraction increases exponentially with "min-entropy loss" $(N-\ell)$, while the encryption scheme remains perfectly secure. The proof of this result and further discussion of encryption is given in Section 3.

To summarize, nearly perfect randomness is not inherently needed to generate *indistinguishable* distributions, while weak (i.e., simulatable) sources are too general for this task (see also Appendix A).

PRIVATE-KEY AUTHENTICATION. We also consider the question of information-theoretic private-key message authentication [14] (see also [25]). Recall, the security of such *authentication codes* is given by the parameter ε, which is the maximal probability of Eve's success (i.e., changing the ciphertext c for m into a valid ciphertext c' of some $m' \neq m$). For concreteness, we will restrict our attention to the simplest case of one-bit messages (just like we did for encryption). This will simplify our analysis, without qualitatively changing our conclusions. Indeed, to authenticate long messages one typically uses various types of *universal hash functions* [6] (see [6, 27, 4, 15] for examples). For one-bit messages, many much more trivial techniques suffice (we will see examples in Section 4).

As with the encryption, we first address the possibility of basing message authentication on weak sources. Interestingly, the result we obtain is quite different.

Theorem 2 *The optimal one-bit authentication code achieves error probability $\varepsilon = \min[2^{N/2-\ell}, 1]$ against a weak source of min-entropy ℓ. In particular, one can non-trivially tolerate weak sources of min-entropy $\ell > N/2 + \omega(1)$, but cannot go beyond this "threshold".*

Therefore, when $N^\gamma < \ell < N/2$ (for any $\gamma > 0$), we see that the weak source can simulate BPP algorithms, but cannot be used even for the most basic 2-message authentication. On the other hand, when $N/2 < \ell < N$, one can at least build secure 2-message authentication codes, but cannot extract even a single non-constant (let alone random) bit. Also, the threshold $N/2$ is quite different from the corresponding threshold N for encryption.

Finally, we show that a strong separation between the possibility of authentication and extraction continues to hold even when $\ell < N/2$. More specifically, we show

Theorem 3 *There exists an imperfect N-bit source, each of whose distributions has min-entropy* at most *ℓ (i.e., all of them have "low entropy"!), and such that:*

1. *There exists a one-bit authentication code achieving nearly optimal error probability $\varepsilon = 2^{-\ell/2+O(1)}$ against any distribution in S;*
2. *Any bit extraction function can be fixed to a constant by some source in S.*

In other words, one can potentially build a secure authentication code even for some "low-entropy" sources, but still completely fail in extracting even a single bit from this source. The proofs of the above results and further discussion of message authentication appear in Section 4.

3 Private-Key Encryption

In this section we discuss our approach for encryption in more detail (in particular, prove our main Theorem 1). We will find it convenient to slightly change our notation. Let $\mathcal{K}$ denote the universe of shared keys, and let $|\mathcal{K}| = u$ (i.e., $u = 2^N$, but we will not insist on it). Similarly, let $\mathcal{C}$ be the set of ciphertexts and $|\mathcal{C}| = n$. Also, it will be easier to replace the notion of min-entropy ℓ by an equivalent notion of *uniformity*. We will say that a distribution over the universe $\mathcal{K}$ of size u is *α-uniform*, where $\alpha \in [0, 1]$, if no element occurs with probability larger than $1/\alpha u$ (for simplicity, we will assume throughout that αu is an integer). Similarly, an imperfect source is α-uniform if all its distributions are such. Clearly, $\alpha = 2^\ell/u$ where ℓ is the corresponding min-entropy, so our change is purely syntactic. We will also call a distribution *flat* if it is uniform over some subset T of $\mathcal{K}$ (i.e., every element of T comes with probability $1/|T|$).

GRAPH REPRESENTATION. Given any candidate one-bit encryption scheme (E, D), we now give a purely graph-theoretic representation of this scheme. Consider the following directed graph $G = G(E, D)$. The n vertices of G are the n possible ciphertexts $c \in \mathcal{C}$. G will also have exactly u directed edges (call this set $\mathcal{E}$) — one for each possible shared key K. The directed edge $e_K \in \mathcal{E}$, labeled by key K, will connect vertices $E_K(0)$ (the head) and $E_K(1)$ (the tail). In this view, to encrypt 0 Alice will send to Bob the head of e_K, and to encrypt 1 she will send the tail of e_K. We let $\mathsf{IN}(c)$ denote the (multi)set of edges incoming to c (i.e., those whose tail is c), and by $\mathsf{OUT}(c)$ the (multi)set of outgoing edges. Notice, since Bob should be able to decrypt, G cannot have self-loops (i.e., $E_K(0) \neq E_K(1)$), so the sets $\mathsf{IN}(c)$ and $\mathsf{OUT}(c)$ are disjoint. Thus, an encryption scheme (E, D) is equivalent to specifying a directed (multi)graph with $|\mathcal{K}|$ edges, $|\mathcal{C}|$ vertices, and no self-loops.

Assume we are given some distribution p on $\mathcal{K}$. This distribution can be viewed as assigning a non-negative weight $p(K)$ to the edge e_K. Conversely, any non-zero weight assignment to $\mathcal{K}$ corresponds to some probability distribution p (by rescaling the weights so that they sum to 1). Therefore, we will identify these two concepts. We say that a weight assignment forms a *circulation*, if for every node $c \in \mathcal{C}$, "incoming" weight to c is equal to the "outgoing" weight from c: $w_{in}(c) \stackrel{\text{def}}{=} \sum_{e_K \in \mathsf{IN}(c)} p(K) = \sum_{e_K \in \mathsf{OUT}(c)} p(K) \stackrel{\text{def}}{=} w_{out}(c)$.

Lemma 1 *The encryption (E, D) is perfectly secure against distribution p on $\mathcal{K}$ if and only if the weight assignment above induces a circulation.*

Proof: The values $w_{in}(c)$ and $w_{out}(c)$ are respectively proportional to the conditional probabilities that that the encrypted bit was 1 or 0 given that the ciphertext was c. The encryption is perfect iff these are always equal. □

We remark that the simplest possible circulation corresponds to any simple (uniformly weighted) directed cycle in G. Additionally, it is well know that any circulation can be decomposed into a weighted sum of such uniform cycles (the converse is true as well). Finally, flat circulations decompose into a *disjoint* union of such cycles.

BIT EXTRACTION. Any deterministic bit extraction function $f : \mathcal{K} \to \{0, 1\}$ can be viewed as a *two-coloring* χ_f of the edges $\mathcal{E}$ of G. Let us call the colors "red" and "blue". Given a particular distribution p on $\mathcal{K}$, we define its weight on red edges to be $\mathsf{Red}(\chi_f, p) = \sum_{K: f(K)=0} p(K) = \Pr_{K \leftarrow p}(f(K) = 0)$, and similarly for $\mathsf{Blue}(\chi_f, p)$. The *fairness* of χ_f on p is simply the corresponding fairness of the extracted random bit $f(K)$: $\mathsf{F}(\chi_f, p) \stackrel{\text{def}}{=} \min[\mathsf{Red}(\chi_f, p), \mathsf{Blue}(\chi_f, p)]$. Given an imperfect source S, the quality of extraction given by coloring χ_f is $\mathsf{F}_S(\chi_f) = \min_{p \in S} \mathsf{F}(\chi_f, p)$. Namely, we select the source $p \in S$ that biases $f(K)$ as much as possible. Finally, the best extraction function f against S defines the quantity $\mathsf{F}_S = \max_f \mathsf{F}_S(\chi_f)$. To summarize, the quality of randomness extraction from S is given as the optimal value F_S of the following zero-sum game: (1) the first player tries to maximize the game value and chooses a two-coloring χ; (2) the second player tries to minimize the game value and chooses a distribution $p \in S$; (3) the value of this specific outcome is the fairness $\mathsf{F}(\chi, p)$.

BIT EXTRACTION VS. BIT ENCRYPTION. Having developed the terminology above, let us return to the original conjecture posed in Section 2. The question was to separate extractable sources from cryptographic sources for encryption. The approach suggested in Theorem 1 was the following. We want to see if there exists an encryption scheme (E, D) such that for a given min-entropy level ℓ one can find an imperfect source S with this min-entropy such that: (1) (E, D) is secure (in fact, perfect) one-bit encryption for any distribution $p \in S$, but (2) one cannot extract even a single "slightly" random bit from S. First, we can simplify this question as follows. Given a candidate scheme (E, D), we can without loss of generality *define* S to be the family of *all* (min-entropy ℓ) distributions against which (E, D) is perfectly secure. Recalling now our graph representation and Lemma 1, we arrive at the following question. Given a candidate directed (multi)graph G with n vertices and u edges, we let S be the family of all circulations on G which are α-uniform (recall, we will work with uniformity in place of min-entropy). Our goal is to determine F_S, which is the quality of bit extraction from this S. Let us denote this value — now dependent only on G and α — by $\mathsf{Val}(G,\alpha)$. Notice, if $\mathsf{Val}(G,\alpha) \ll 1/2$, encryption scheme (E,D) is exactly the encryption we are looking for to disprove the conjecture. On the other hand, if $\mathsf{Val}(G,\alpha) \approx 1/2$, the feasibility of perfectly encrypting a bit using (E,D) indeed implies the possibility of bit extraction.

WHEN ENCRYPTION $\iff$ EXTRACTION. Before finding graphs G disproving our conjecture, we address the following curious question. Which graphs G (i.e., encryption schemes) actually support the original conjecture? Specifically, when is $\mathsf{Val}(G,0) = 1/2$? ($\alpha = 0$ means not placing any min-entropy restrictions).

Lemma 2 $\mathsf{Val}(G,0) = \frac{1}{2}$ *if and only if G is* bipartite.

Proof: Assume G is not bipartite. Then it has some odd-length cycle C, which defines a flat circulation. Any two-coloring χ will have a different number of red and blue edges in C, which means $\mathsf{Val}(G,0) \le \mathsf{F}(\chi,C) \le \frac{1}{2} - \frac{1}{2|C|} < \frac{1}{2}$. On the other hand, if G is bipartite, then its vertex set can be partitioned into left set L and right set R, so that all the edges go between L and R. Now define χ by coloring all the edges from L to R red and those from R to L — blue. For any circulation p, the amount of outgoing flow from L to R should be equal to the amount of incoming flow from R to L, which means that the weight of red edges is the same as the weight of blue edges: $\mathsf{Red}(\chi,p) = \mathsf{Blue}(\chi,p)$, but this means that our coloring extracts a perfect coin. □

3.1 Proof of Theorem 1

We now come back to our main result. First, using the notation developed so far, we can restate Theorem 1 in the following (even stronger) form:

Theorem 4 *For any universe size u and uniformity level $\alpha \in (0, \frac{1}{16}]$,*[4] *define $\beta = \max[\alpha, 1/\sqrt{u}]$. Then, there exists a single graph G^* such that*

- $\mathsf{Val}(G^*,\alpha) = O(\beta)$. *In particular, for any $\alpha = o(1)$ we have* $\mathsf{Val}(G^*,\alpha) = o(1)$.
- *For any G,* $\mathsf{Val}(G,\alpha) = \Omega(\beta)$, *so the graph G^* above is nearly optimal.*

We remark that the result above can be viewed as a *precise* calculation to the value of the following game, given by parameters u and α. It it played by "minimization" player A and "maximization" player B:

A. Selects number of vertices n and a directed graph G with n vertices and u edges.
B. Selects a two-coloring χ of G.
A. Selects an α-uniform circulation p in G. The value of the game is $\mathsf{F}(\chi,p)$.

Theorem 4 states that this value is $\Theta(\max[\alpha, 1/\sqrt{u}])$.

[4]The choice of this constant, as well of some other constants in this section, is arbitrary and is not necessarily optimal.

UPPER BOUND. From Lemma 2, the graph G^* should be highly non-bipartite. So we let G^* be the complete directed graph on n vertices, i.e. $u = n(n-1) \approx n^2$. We show that this graph is nearly optimal in separating extractable and cryptographic sources for encryption. We start with computing $\mathsf{Val}(G^*,0)$, i.e. the optimal discrepancy when no constraints are put on the min-entropy of our circulation.

Lemma 3 $\mathsf{Val}(G^*,0) = \frac{1}{n} \approx \frac{1}{\sqrt{u}}$.

Proof: For the lower bound, consider the *lexicographic* coloring χ of G. Namely, color (i,j) red if $i < j$ and blue otherwise. Any cycle C (say, of length $s \le n$) must have at least one edge of each color, which means that $\mathsf{F}(\chi,C) \ge 1/s \ge 1/n$. On the other hand, any circulation p can be written as a convex combination of simple cycles. By linearity of $\mathsf{Red}(\chi,\cdot)$ and $\mathsf{Blue}(\chi,\cdot)$, this implies that the weight of red (resp. blue) edges in p is lower bounded by the corresponding weight in at least one of the cycles in the convex combination, and the latter we know is at least $1/n$.

For the upper bound, take *any* coloring χ of the edges of G^*. If *any* 2-cycle $i \to j \to i$ in G is monochromatic, we would get $\mathsf{F}(\chi, i \to j \to i) = 0 < 1/n$. Thus, we can assume that among each pair of edges (i,j) and (j,i), exactly one is red and one is blue. But this means that the subgraph of, say, blue edges forms a *tournament*. However, it is well known (e.g., see [26, p. 175]) that any tournament has a Hamiltonian path (the proof follows by a simple induction on the number of vertices). This means that there exists a length $(n-1)$ path consisting only of blue edges. Completing this path into a Hamiltonian cycle (by either a red or a blue edge), we get a cycle C with $\mathsf{F}(\chi,C) \le 1/n$, as needed. □

Next, we show that the bound $\mathsf{Val}(G^*,\alpha) = O(1/n)$ extends to all $\alpha \le 1/2n = \Omega(1/\sqrt{u})$ as well. Indeed, consider any two-coloring χ, as before. Let us look at all monochromatic 2-cycles in χ. If this number is at least $n/2$, this means that there are at least $n/4$ monochromatic 2-cycles of the same color. Taking the union of these 2-cycles gives a flat circulation p with $\frac{n}{2} > n(n-1) \cdot \frac{1}{2n} \ge \alpha u$ edges having $\mathsf{F}(\chi,p) = 0$. If the number of monochromatic cycles is less than $n/2$, let us remove from G^* one arbitrary vertex in each of the monochromatic 2-cycles. We get a two-coloring of a complete graph G' on at least $n/2$ vertices, where no 2-cycle is monochromatic. By the previous argument, we can find a Hamiltonian cycle C in G', which has at least $n/2 \ge \alpha n$ edges and achieves $\mathsf{F}(\chi,C) \le 2/n$.

Hence, to prove Theorem 4, i.e. $\mathsf{Val}(G^*,\alpha) = O(\max(\alpha, 1/\sqrt{u})) = O(\max(\alpha, 1/n))$, it suffices to consider the case when $\alpha \ge 1/2n$ and show $\mathsf{Val}(G^*,\alpha) = O(\alpha)$. As before, take any coloring χ, and assume wlog that it contains at least $n(n-1)/2$ blue edges. Recall, our goal is to find an α-uniform circulation p such that $\mathsf{F}(\chi,p) = O(\alpha)$. We will in fact produce a *flat* circulation satisfying this condition. Namely, our circulation will consist of $\alpha u = \alpha n(n-1)$ edges with uniform weight on them. Recall, a flat circulation can be decomposed into a disjoint union of cycles. And this is in fact the way we will build our p. We will keep adding some carefully chosen cycles C to p, each time removing C from our graph G^* (this will ensure that the cycles are disjoint), until we add a total of $\alpha n(n-1)$ edges, as required by the min-entropy requirement. Each cycle C will contain at most $O(\alpha)$ fraction of red edges, guaranteeing that $\mathsf{F}(\chi, p) = O(\alpha)$, as needed.

PICKING THE CYCLES. We now describe the procedure of choosing our cycles. First, we keep adding cycles which are entirely blue, until no such cycles are left in G^* (remember, we remove the cycle the moment we add it to p). If we already got $\alpha n(n-1)$ edges in p, we stop. Otherwise, at the end we are left with an *acyclic* "blue" subgraph G' containing at least $n(n-1)(\frac{1}{2}-\alpha)$ edges. Let us topologically order the vertices of G' so that all the edges go from left to right. As a combinatorial result of independent interest, we will show that such G' always contains a directed (blue) path of length $\Omega(1/\alpha)$ (here we use $\alpha > \Omega(1/n)$). It seems that we are done: complete the path above to a cycle C (which has $\mathsf{F}(\chi, C) = O(\alpha)$) and add it to p. However, we have to ensure that we will never reuse the "back" edge we use to complete the cycle. Thus, we prove an even stronger combinatorial result.

Lemma 4 *Let G' be an acyclic directed graph having n vertices and $u' \geq n(n-1)(\frac{1}{2}-\alpha)$ edges. Then G' contains at least αn^2 directed paths of length $\Omega(1/\alpha)$, each having a distinct pair of starting and ending vertices.*

Postponing the proof of Lemma 4 for a second, we argue that it allows us to complete the argument. Namely, since p always has at most αn^2 edges (which consequently are not present in G'), we can find a length $\Omega(1/\alpha)$ "blue" path in G' such that the "back" edge it needs to become a cycle is still present in G'. Therefore, we can keep finding almost blue cycles until the size of p becomes $\alpha n(n-1)$, as needed. The proof of Lemma 4 below then completes the first part of Theorem 4.

Proof: Let us denote by $1 \ldots n$ the n vertices of G' listed in their topological order. Let G'' denote the "complement" graph containing at most $\alpha n(n-1)$ forward edges that are not present in G'. Let $d = 8\alpha n$ (notice, $4 \leq d \leq n/2$ since $1/2n \leq \alpha \leq 1/16$), $G_0 = G'$, $n_0 = n$, $k = 0$, and repeat the following procedure until impossible. Given a vertex $i \in \{1 \ldots (n_k - d)\}$ of G_k, we call call vertices $\{i+1, \ldots, i+d\}$ of G_k the *immediate neighborhood* of i. We say that i is *lonely*, if it has at most $d/2$ outgoing edges to its immediate neighborhood (i.e., at most $d/2$ of the edges $(i, i+1), (i, i+2), \ldots, (i, i+d)$ are present in G_k). If the graph G_k has at least one lonely vertex i, we remove i (and all its adjacent edges) from G_k, thus forming a new graph G_{k+1} with $n_{k+1} = n_k - 1$ vertices. In particular, we *rename* the vertices of G_{k+1} so that they are numbered from 1 to $n_{k+1} = n_k - 1$. Finally, we increment k.

We notice that in each step we removed a vertex which did not have at least $d/2$ forward neighbors, which means that we removed at least $d/2$ new edges in the complement graph G''. Since G'' only had $\alpha n(n-1)$ edges to begin with, the number of times k we could find such a lonely vertex is at most $k \leq \alpha n^2/(d/2) = \alpha n^2/4\alpha n = n/4$. Hence, the final graph G_k has at least $3n/4$ vertices, none of which is lonely. Now, take an arbitrary starting point $i \in \{1 \ldots n/4\}$ in G_k, and greedily construct a forward path by iteratively picking any point in the immediate neighborhood of the current point (also stopping when we cross $n_k - d$). Since no points below $(n_k - d) \geq n/2$ are lonely, the length of the path is at least $(n/4)/d = \Omega(1/\alpha)$. Moreover, we have at least $n/4$ choices for the starting and $d/2 = 4\alpha n$ choices for the ending points. Therefore, the total number of distinct source/destination paths we can construct is at least αn^2, as claimed. $\square$

LOWER BOUND. Take any graph G with u edges and n vertices. To show that $\mathsf{Val}(G, \alpha) = \Omega(\beta)$, where $\beta = \max[\alpha, 1/\sqrt{u}]$, we need to show the existence of a coloring χ such that for any α-uniform circulation p we have $\mathsf{F}(\chi, p) = \Omega(\beta)$. We will show that such χ exists by *probabilistic method*. We randomly label the vertices of G by numbers from 1 to n, and color edge (i, j) of G red if $i < j$ and blue otherwise. We show that such coloring satisfies the needed property with non-zero probability, and therefore *exists*.

First, we prove the bound $\Omega(1/\sqrt{u})$. For that, we show that with high probability, G does not contain a blue (resp. red) path of length $\ell \stackrel{\text{def}}{=} 3\sqrt{u}$. Indeed, taking any path of G of length ℓ, the probability that it gets all red or all blue is exactly $2/\ell! < (e/\ell)^\ell$. Since the overall number of paths of length ℓ is certainly less than $\binom{u}{\ell} < (eu/\ell)^\ell$, the expected number of monochromatic length ℓ paths is less than $(\frac{eu}{\ell} \cdot \frac{e}{\ell})^\ell < (\frac{9u}{\ell^2})^\ell = 1$, since $\ell = 3\sqrt{u}$. Thus, some ordering with the given property exists. Now, fix any such ordering and the corresponding coloring χ, and take any circulation p. Decompose p into cycles. The property of our ordering ensures that in each cycle C, at most ℓ consecutive edges are monochromatic, so $\mathsf{F}(\chi, C) \geq 1/\ell$. Thus implies that $\mathsf{F}(\chi, p) = \Omega(1/\sqrt{u})$ as well.

Next, we show the bound $\Omega(\alpha)$. For that, call an edge (i, j) *short* in the resulting ordering if $|i - j| < d$, where $d = \alpha(n-1)/4$. Notice, the probability that a given edge of G becomes short is at most $\frac{2dn}{(n-1)n} = \frac{\alpha}{2}$, by our choice of d. Therefore, the expected number of short edges is at most $\alpha u/2$. In particular, some ordering will produce at most $\alpha u/2$ short edges. Now, fix any such ordering and the corresponding coloring χ, and take any α-uniform circulation p. Since the weight of each edge in p is at most $1/\alpha u$, the total weight of short edges in p is at most $1/2$, meaning that "long" edges must have weight at least $1/2$ too. Now decompose p into cycles and take any resulting cycle C. We claim that any consecutive sequence of blue (same argument hold for red as well) edges can contain at most $(n-1)/d = O(\alpha)$ *long* blue edges (but can contain more short blue edges). Indeed, since blue edges go "forward" by at least d steps, one cannot have more that $(n-1)/d$ blue edges without have at least one "backward" red edge. This implies that the total weight of the red edges in this cycle is at least an $\Omega(\alpha)$ fraction of the weight of *long* blue edges. Since this bound holds for every cycle C, it holds for the entire circulation p as well. Thus, the total weight of red edges (call it r) in p is at least $\Omega(\alpha)$ fraction of the weight of long blue edges (call it b_l): $r = \Omega(\alpha) \cdot b_l$. But since all red and all long blue edges include all long edges, it means $r + b_l \geq 1/2$ (remember, short edges weight at most $1/2$), which implies that $r = \Omega(\alpha)$, completing the proof.

4 Private-Key Authentication

We now address the question of building a one-time messages authentication code for one-bit messages. Our results could be viewed as the first step towards basing more general (many-time, larger message spaces) authentication codes on imperfect sources. A lot of our notation will parallel what we used for encryption in Section 3. In particular, we will also use graphs to represent an authentication code (E, D) with key space $\mathcal{K}$ of cardinality u and tagging space[5] $\mathcal{C}$ of cardinality n. However, it will be more natural to use an *undirected bipartite* (multi)graph for this purpose. Namely, this graph G has a left side L and a right side R — both being a copies of the tagging space $\mathcal{C}$. As before, there will be u edges e_K, corresponding to different secret keys $K \in \mathcal{K}$. The edge e_K will connect the "left" copy of $E_K(0)$ to the "right" copy of $E_K(1)$ (given $c \in \mathcal{C}$, we let c_ℓ and c_r denote the left and right copies of c in G). Notice, there is no restriction about not connecting c_ℓ to c_r, and also edges could be duplicated.

FLAT DISTRIBUTIONS. As before, a probability distribution p on $\mathcal{K}$ can be viewed as assigning weights to the edges of G. Given such distribution p and observing a tag c of some bit b (say, $b = 0$), the optimal strategy for producing the tag c' of $(1-b) = 1$ involves picking the vertex $c'_r \in R$ having the largest weight going from c_ℓ to c'_r. Because of that, flat distributions will play a particularly important role in our study. Recall, such distributions assign equal weight to some subset of $\mathcal{K}$. It is well known that every α-uniform distribution is a convex combination of α-uniform flat distributions. This implies that among all α-uniform sources, the best ones for the adversary are exactly the flat distribution having αu edges in their support.

[5] We find it more natural to refer to the output of $E_K(\cdot)$ as a "tag" rather than a "ciphertext" like we did for encryption.

Now, let p be an α-uniform flat distribution having support on the edge set $\mathcal{E}'$ of $u' = \alpha u$ edges. Let $\Gamma_{0\to 1}(p)$ denote the optimal probability of the adversary to produce a valid tag for 1 after seeing the tag for 0, and similarly for $\Gamma_{1\to 0}(p)$. The security of the authentication code (E, D) on distribution p is then $\varepsilon(p) = \max[\Gamma_{0\to 1}(p), \Gamma_{1\to 0}(p)]$. Let L' (R') be the set of left (right) vertices belonging to some edge in $\mathcal{E}'$, and let $n_\ell = |L'|$ and $n_r = |R'|$. We will also call our flat distribution p *simple* if no two edges in $\mathcal{E}'$ connect the same pair of vertices (i.e., all the keys are functionally distinct).

Lemma 5 *For any flat distribution p,*

$$\Gamma_{0\to 1}(p) \geq \max\left[\frac{n_\ell}{\alpha u},\ \frac{1}{n_r}\right];\ \Gamma_{1\to 0}(p) \geq \max\left[\frac{n_r}{\alpha u},\ \frac{1}{n_\ell}\right]$$

and, thus, $\varepsilon(p) \geq \frac{1}{\sqrt{\alpha u}}$. *For* simple *flat distributions,* $\Gamma_{0\to 1}(p) = \frac{n_\ell}{\alpha u}$, $\Gamma_{1\to 0}(p) = \frac{n_r}{\alpha u}$, $\varepsilon(p) = \frac{\max[n_\ell, n_r]}{\alpha u}$.

Proof: The fact that $\Gamma_{0\to 1}(p) \geq 1/n_r$ is obvious since there are only n_r possible tags for 1. Next, let $d(c)$ be the degree of the node c in $\mathcal{E}'$. Then the probability that $E_K(0) = c_\ell$ is equal to $d(c_\ell)/\alpha u$. On the other hand, conditioned on $E_K(0) = c_\ell$, there are at most $d(c_\ell)$ possibilities for $E_K(1)$, implying that the adversary can predict the value $E_K(1)$ with probability at least $1/d(c_\ell)$ (the latter becomes equality for simple flat distributions). Thus, $\Gamma_{0\to 1}(p) \geq \sum_{c_\ell \in L'} \frac{d(c_\ell)}{\alpha u} \cdot \frac{1}{d(c_\ell)} = \frac{n_\ell}{\alpha u}$. Similar proof holds for $\Gamma_{1\to 0}(p)$. Finally, $\varepsilon(p) = \max[\Gamma_{1\to 0}(p), \Gamma_{0\to 1}(p)] \geq \max[\frac{n_\ell}{\alpha u}, \frac{1}{n_\ell}, \frac{n_r}{\alpha u}, \frac{1}{n_r}] \geq \frac{1}{\sqrt{\alpha u}}$. $\square$

PROOF OF THEOREM 2. We can now examine the construction of optimal authentication codes secure against weak sources. In our notation, Theorem 2 states that the optimal authentication code for all α-uniform distributions achieves error $\min[\frac{1}{\alpha\sqrt{u}}, 1]$. For the upper bound, consider the complete bipartite graph G^* on n nodes (so that $u = n^2$). Recall, it suffices to consider only flat α-uniform distributions. Notice, each such distribution is necessarily simple. Then, for any such distribution on αu edges touching n_ℓ left and n_r right nodes of G^*, applying Lemma 5 yields that $\varepsilon = \frac{\max[n_\ell, n_r]}{\alpha u} \leq \frac{n}{\alpha u} = \frac{\sqrt{u}}{\alpha u} = \frac{1}{\alpha\sqrt{u}}$, as needed. In retrospective and coming back to our original notation, the above authentication code is extremely simple. One splits an N-bit secret key into two equal length random pads s_0 and s_1. Then, to authenticate a bit b, Alice can use the pad s_b. Intuitively, if the min-entropy of the source is above $N/2$, learning s_b still leaves some randomness in s_{1-b}, so the latter indeed cannot be predicted well.

We next show that the above graph G^* is indeed optimal for dealing with α-uniform sources. For any graph G on u nodes, we consider two possibilities. First, assume the edges of G touch at least $\sqrt{u}$ left vertices, i.e. $|L| \geq \sqrt{u}$. Take any subgraph of G with αu edges which also touches

$\sqrt{u}$ left vertices (making n_ℓ at least $\sqrt{u}$ for the corresponding flat distribution p). By Lemma 5, $\varepsilon(p) \geq n_\ell/\alpha u \geq \sqrt{u}/\alpha u = 1/\alpha\sqrt{u}$. On the other hand, assume the edges of G do not touch $\sqrt{u}$ vertices the left side, i.e. $|L| \leq \sqrt{u}$. Take an α fraction of left vertices having the largest degree in G. They form the set L' of size $\alpha|L| \leq \alpha\sqrt{u}$. Clearly, the vertices of L' have at least αu edges of G adjacent to them. We make these edges form our flat distribution p. Then, Lemma 5 again implies that $\varepsilon(p) \geq 1/n_\ell \geq 1/\alpha\sqrt{u}$.

PROOF OF THEOREM 3. Finally, we show that the separation between extractable and cryptographic sources holds for "low" levels of min-entropy as well. In our notation, Theorem 3 states that there exists a graph G^* on u edges and a family S of *at most* α-uniform distributions on the edges of G, so that: (1) $\varepsilon(p) \leq O(1/\sqrt{\alpha u})$, for all $p \in S$; but (2) for every two-coloring χ of the edges of G, the support set of some $p \in S$ is monochromatic.

As earlier, we let G^* be the complete bipartite graph on n vertices (so that $u = n^2$). The family S will consist of the flat distributions corresponding to the following sets of αn^2 edges. Take any left and right subsets $L' \subset L$ and $R' \subset R$ of cardinality $\sqrt{2\alpha} \cdot n = \sqrt{2\alpha u}$. Then take any subgraph of size αu of the complete bipartite subgraph $L' \times R'$. Since all our flat distributions are simple, we get by Lemma 5 that $\varepsilon(p) \leq \frac{\sqrt{2\alpha u}}{\alpha u} = O(\frac{1}{\sqrt{\alpha u}})$, as desired. Notice, this is nearly the best possible by Lemma 5 too, since for any flat α-uniform p we have $\varepsilon(p) \geq \frac{1}{\sqrt{\alpha u}}$.

It remains to show that no bit extraction is possible from S. For that, take any two-coloring χ of the edges of G^*, and assume wlog that at least $n^2/2$ edges are colored blue. Let us look at the subgraph G' formed by these blue edges. We need to show that G' contains at least one distribution in S, i.e. that there exists L' and R' of cardinality $\sqrt{2\alpha u}$ such that the complete subgraph $L' \times R'$ contains at least αn^2 blue edges. We show the existence of such L' and R' by probabilistic method. Namely, pick L' and R' of size $\sqrt{2\alpha u}$ completely at random. Each blue edge will get inside $L' \times R'$ with probability 2α, so the expected number of blue edges inside $L' \times R'$ is at least $2\alpha \cdot \frac{n^2}{2} = \alpha n^2$. This shows that some L' and R' matching the above expectation exist, completing the proof.

5 Conclusions and Further Research

In this work we investigated the extent to which conventional cryptographic primitives such as encryption and authentication can be build based on imperfect sources of randomness. In particular, we compared the class of such "cryptographic" sources for the applications above with the well studied classes of weak (i.e., simulatable) and extractable random sources. Our results illustrate that the set of sources sufficient for various cryptographic applications seems to be quite different from the above well studied classes, and also strongly depends on the cryptographic task at hand. Thus, cryptographic primitives do not inherently rely on ideal randomness, but cannot tolerate very general weak sources of randomness.

We believe that our initial investigation of the possibility of basing cryptography on imperfect random sources will inspire a lot of further research. In particular, many questions remain open. For example, it is interesting to extend our quantitative results for private-key encryption and especially authentication to larger than one-bit message spaces. It is also interesting to consider other information-theoretic primitives like authenticated encryption and secret sharing schemes. Finally, many new questions appear when we look at *computationally* secure primitives (like one-way functions or public-key encryption and signature schemes). In particular, we still have to rely on (possibly stronger!) computational assumptions in order to build computational primitives which are secure against various imperfect sources. Investigating which such sources can still be tolerated in this setting is a very interesting research direction.

Acknowledgments: We would like to thank Petar Maymounkov, Amit Sahai, Luca Trevisan and Salil Vadhan for useful discussions.

References

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

[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] C. Bennett, G. Brassard, and J. Robert. Privacy Amplification by public discussion. In *SIAM J. on Computing*, pp. 17(2):210–229, 1988.

[4] J. Bierbrauer, T. Johansson, G. Kabatianskii, B. Smeets. On Families of Hash Functions via Geometric Codes and Concatenation. In *Proc. of CRYPTO*, pp. 331–342, 1993.

[5] M. Blum. Independent unbiased coin-flipsfrom a correclated biased source — a finite state Markov chain. *Combinatorica*, 6(2):97–108, 1986.

[6] L. Carter and M. Wegman. Universal classes of hash functions. In *JCSS*, vol. 18, pp. 143–154, 1979.

[7] B. Chor, J. Friedman, O. Goldreich, J. Håstad, S. Rudich, R. Smolensky. The Bit Extraction Problem or t-resilient Functions. In *Proc. of FOCS*, pp. 396–407, 1985.

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

[9] A. Cohen, A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proc. of FOCS*, pp. 14–19, 1989.

[10] Y. Dodis. New Imperfect Random Source with Applications to Coin-Flipping. In *Proc. of ICALP*, pp. 297–309, 2001.

[11] Y. Dodis, A. Sahai and A. Smith. On Perfect and Adaptive Security in Exposure-Resilient Cryptography. In *Proc. of EuroCrypt*, pp. 301–324, 2001.

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

[13] J. Friedman. On the Bit Extraction Problem. In *Proc. of FOCS*, pp. 314–319, 1992.

[14] E. Gilbert, F. MacWilliams and N. Sloane. Codes which detect deception. In *Bell Systems Technical J.*, 53:405–424, 1949.

[15] T. Helleseth, T. Johansson. Universal Hash Functions from Exponential Sums over Finite Fields and Galois Rings. In *proc. of CRYPTO*, pp. 31–44, 1996.

[16] J. Kahn, G. Kalai, N. Linial. The Influence of Variables on Boolean Functions. In *Proc. of* 30*th FOCS*, pp. 68–80, 1989.

[17] K. Kurosawa, T. Johansson and D. Stinson. Almost k-wise independent sample spaces and their cryptologic applications. In *J. of Cryptology*, 14:231–253, 2001.

[18] D. Lichtenstein, N. Linial, M. Saks. Some Extremal Problems Arising from Discrete Control Processes. *Combinatorica*, 9:269–287, 1989.

[19] J. McInnes, B. Pinkas. On the Impossibility of Private Key Cryptography with Weakly Random Keys. In *Proc. of CRYPTO*, pp. 421–435, 1990.

[20] N. Nisan, A. Ta-Shma. Extracting Randomness: a survey and new constructions. In *JCSS*, 58(1):148–173, 1999.

[21] N. Nisan, D. Zuckerman. Randomness is Linear in Space. In *JCSS*, 52(1):43–52, 1996.

[22] O. Reingold, R. Shaltiel, A. Wigderson. Extracting randomness via repeated condensing. In *Proc. of FOCS*, 2000.

[23] M. Sántha, U. Vazirani. Generating Quasi-Random Sequences from Semi-Random Sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[24] C. Shannon. Communication Theory of Secrecy systems. In *Bell Systems Technical J.*, 28:656–715, 1949.

[25] G. J. Simmons. An introduction to shared secret and/or shared control schemes and their application. In *"Contemporary Cryptology, The Science of Information Integrity"*, G. J. Simmons, ed., IEEE Press, pp. 441-497, 1992.

[26] S. Skiena. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Addison-Wesley, 1990.

[27] D. Stinson. Universal Hashing and Authentication Codes. In *Designs, Codes and Cryptography*, 4(4):369–380, 1994.

[28] L. Trevisan. Construction of Extractors Using PseudoRandom Generators. In Proc. of STOC, pp. 141–148, 1999.

[29] L. Trevisan, S. Vadhan. Extracting Randomness from Samplable Distributions. In *Proc. of FOCS*, 2000.

[30] U. Vazirani. Randomness, Adversaries and Computation. *PhD Thesis*, University of California, Berkeley, 1986.

[31] U. Vazirani. Strong Communication Complexity or Generating Quasi-Random Sequences from Two Communicating Semi-Random Sources. *Combinatorica*, 7(4):375–392, 1987.

[32] U. Vazirani, V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proc. of* 26*th FOCS*, pp. 417–428, 1985.

[33] G. Vernam. Secret Signaling Systems. US Patent, 1919.

[34] J. von Newman. Various techniques used with connection with random digits. In *National Bureau of Standards, Applied Math. Series*, 12:36–38, 1951.

[35] D. Zuckerman. Simulating BPP Using a General Weak Random Source. *Algorithmica*, 16(4/5):367-391, 1996.

A Simple Proof for Impossibility of Basing Encryption on Weak Random Sources

For completeness, we reprove the result of [19] in our graph notation. This result states that for any encryption scheme (E, D) with an N-bit key (i.e, directed graph G on $u = 2^N$ edges) there exists a min-entropy $(N-2)$ (i.e., $\frac{1}{4}$-uniform) distribution p on the edges $\mathcal{E}$ of G such that the ciphertext (i.e., the vertex of G) completely reveals the encrypted bit. First, we notice that the encryption (E, D) is completely insecure against some distribution p if and only if the nodes of G can be decomposed into disjoint parts L and R, such that all positive weight edges in p go from L to R. Indeed, in this case no node has both an incoming and an outgoing edge of positive weight, so there is no uncertainty to the adversary. Thus, it suffices to show that there exist sets L and R such that at least $u/4$ edges of $\mathcal{E}$ go from L to R, since a uniform distribution on these edges will then define the source we need. Let's place each vertex of G into L or R with probability $1/2$ each. Then, each edge e_K of $\mathcal{E}$ with go from L to R (as needed) with probability $1/4$. Thus means that, *on average*, $u/4$ edges will go from L to R, irrespective of what graph G we started from. But this means that the needed L and R *exist*, completing the proof.