One-way Functions Exist iff K^t-Complexity is Hard-on-Average

Yanyi Liu[*] Rafael Pass[†]

Abstract

We prove that the following are equivalent:

- **Existence of one-way functions:** the existence of one-way functions (which in turn are equivalent to PRGs, pseudo-random functions, secure encryptions, digital signatures, commitment schemes, and more).
- **Average-case hardness of K^t-complexity:** the existence of polynomials t, p such that no PPT algorithm can determine the t-time bounded Kolmogorov Complexity for more than a $1 - \frac{1}{p(n)}$ fraction of n-bit strings.

In doing so, we present the first natural, and well-studied, computational problem (i.e., K^t-complexity) that captures the feasibility of non-trivial cryptography.

[*]Cornell
[†]Cornell Tech

1 Introduction

We prove the equivalence of two fundamental problems in the theory of computation: (a) the existence of one-way functions, and (b) average-case hardness of the time-bounded Kolmogorov Complexity problem.

Existence of One-way Functions: A *one-way function* [DH76] (OWF) is a function f that can be efficiently computed (in polynomial time), yet no probabilistic polynomial-time (PPT) algorithm can invert f with inverse polynomial probability for infinitely many input lengths n. Whether one-way functions exist is unequivocally the most important open problem in Cryptography (and arguably the most importantly open problem in the theory of computation, see e.g., [Lev03]): OWFs are both necessary [IL89] and sufficient for many of the most central cryptographic tasks (e.g., pseudorandom generators [HILL99], pseudorandom functions [GGM84], private-key encryption [GM84, BM88], digital signatures [Rom90], commitment schemes [Nao91], and more). Additionally, as observed by Impagliazzo [Gur89, Imp95], the existence of a OWF is also equivalent to the existence of polynomial-time method for sampling hard *solved* instances for an NP language (i.e., hard instances together with their witnesses).[1]

While many candidate constructions of OWFs are known—most notably based on factoring [RSA83], the discrete logarithm problem [DH76], or the hardness of lattice problems [Ajt96]—the question of whether there exists some *natural* computational problem that captures the hardness of OWFs (and thus the feasibility of "non-trivial" cryptography) has been a long standing open problem.[2] This problem is particularly pressing given that many classic OWF candidates (e.g., based on factoring and discrete log) can be broken by a quantum computer [Sho97].

Average-case Hardness of K^t-Complexity: What makes the string 12121212121212121 less random than 60484850668340357492? The notion of *Kolmogorov complexity* (K-complexity), introduced by Solomonoff [Sol64] and Kolmogorov [Kol68], provides an elegant method for measuring the amount of "randomness" in individual strings: The K-complexity of a string is the length of the shortest program (to be run on some fixed universal Turing machine U) that outputs the string x. From a computational point of view, however, this notion is unappealing as there is no efficiency requirement on the program. The notion of $t(\cdot)$*-time-bounded Kolmogorov Complexity* (*K^t-complexity*) overcomes this issue: $K^t(x)$ is defined as the length of the shortest program that outputs the string x within time $t(|x|)$. As surveyed by Trakhtenbrot [Tra84], the problem of efficiently determining the K^t-complexity of strings was studied in the Soviet Union since the 60s as a candidate for a problem that requires "brute-force search" (see Task 5 on page 392 in [Tra84]). The modern complexity-theoretic study of this problem goes back to Sipser [Sip83], Hartmanis [Har83][3] and Ko [Ko86]. Intriguingly, Trakhtenbrot also notes that a "frequential" version of this problem was considered in the Soviet Union in the 60s: the problem of finding an algorithm that succeeds for a "high" fraction of strings x—in more modern terms from the theory of average-case complexity [Lev86], whether K^t can be computed by a heuristic algorithm with inverse polynomial error, over random inputs x. We say that K^t is $\frac{1}{p(\cdot)}$*-hard-on-average*, if no PPT algorithm succeeds in computing $K^t(\cdot)$ for more than an $1 - \frac{1}{p(n)}$ fraction of n-bit strings x, for infinitely many n.

[1] A OWF f directly yields the desired sampling method: pick a random string r and let $x = f(r)$ be the instance and r the witness. Conversely, to see why the existence of such a sampling method implies a one-way function, consider the function f that takes the random coins used by the sampling method and outputs the instance generated by it.

[2] Note that Levin [Lev85] presents an ingenious construction of a *universal one-way function*—a function that is one-way if one-way functions exists. But his construction (which relies on an enumeration argument) is artificial. Levin [Lev03] takes a step towards making it less artificial by constructing a universal one-way function based on a new specially-tailored *Tiling Expansion problem*.

[3] Hartmanis's paper considered a somewhat different notion of K^t complexity.

Our man result shows that the existence of OWFs is equivalent to the average-case hardness of the K^t-complexity problem. In doing so, we present the first natural (and well-studied) computational problem that captures the feasibility of "non-trivial" cryptography.

Theorem 1.1. *The following are equivalent:*

- *The existence of one-way functions.*
- *The existence of polynomials $t(n) > 2n, p(n) > 0$ such that K^t is $\frac{1}{p(\cdot)}$-hard-on-average.*

1.1 Proof outline

We provide a brief outline for the proof of Theorem 1.1.

OWFs from Avg-case K^t-Hardness We show that if K^t is average-case hard for some $t(n) > 2n$, then a weak one-way function exists[4]; the existence of (strong) one-way functions then follows by Yao's hardness amplification theorem [Yao82]. Let c be a constant such that every string x can be output by a program of length $|x| + c$ (running on the fixed Universal Turing machine U). Consider the function $f(\ell||M')$, where ℓ is of length $\log(n + c)$ and M' is of length $n + c$, that lets M be the first ℓ bits of M', and outputs $\ell||y$ where y is the output of M after $t(n)$ steps. We aim to show that if f can be inverted with high probability—significantly higher than $1 - 1/n$—then K^t-complexity of random strings $z \in \{0, 1\}^n$ can be computed with high probability. Our heuristic $\mathcal{H}$, given a string z, simply tries to invert f on $\ell||z$ for all $\ell \in [n + c]$, and outputs the smallest ℓ for which inversion succeeds. First, note that since every length $\ell \in [n + c]$ is selected with probability $1/(n+c)$, the inverter must still succeed with high probability even if we condition the output of the one-way function on any particular length ℓ (as we assume that the one-way function inverter fails with probability significantly smaller than $\frac{1}{n}$). This, however, does not suffice to prove that the heuristic works with high probability, as the string y output by the one-way function is not uniformly distributed (whereas we need to compute the K^t-complexity for uniformly chosen strings). But, we show using a simple counting argument that y is not too "far" from uniform in relative distance. The key idea is that for every string z with K^t-complexity w, there exists some program M_z of length w that outputs it; furthermore, by our assumption on c, $w \leq n + c$. We thus have that $f(\mathcal{U}_{n+c+\log(n+c)})$ will output $w||z$ with probability at least $\frac{1}{n+c} \cdot 2^{-w} \geq \frac{1}{n+c} \cdot 2^{-(n+c)} = O(\frac{2^{-n}}{n})$ (we need to pick the right length, and next the right program). So, if the heuristic fails with probability δ, then the one-way function inverter must fail with probability at least $\frac{\delta}{O(n)}$, which concludes that δ must be small (as we assumed the inverter fails with probability significantly smaller than $\frac{1}{n}$).

Avg-case K^t-Hardness from EP-PRGs To show the converse direction, our starting point is the earlier result by Kabanets and Cai [KC00] and Allender et al [ABK+06] which shows that the existence of OWFs implies that K^t-complexity must be *worst-case* hard to compute. In more detail, they show that if K^t-complexity can be computed in polynomial-time for *every* input x, then pseudo-random generators (PRGs) cannot exists. This follows from the observations that (1) random strings have high K^t-complexity with overwhelming probability, and (2) outputs of a PRG always have small K^t-complexity (as the seed plus the constant-sized description of the PRG suffice to compute the output). Thus, using an algorithm that computes K^t, we can easily distinguish outputs of the PRG from random strings—simply output 1 if the K^t-complexity is high, and 0 otherwise. This method, however, relies on the algorithm working for every input. If we only have access to a heuristic $\mathcal{H}$ for K^t, we have no guarantees that $\mathcal{H}$ will output a correct value when we feed it a pseudorandom string, as those strings are *sparse* in the universe of all strings.

[4]Recall that an efficiently computable function f is a weak OWF if there exists some polynomial $q > 0$ such that f cannot be efficiently inverted with probability better than $1 - \frac{1}{q(n)}$ for sufficiently large n.

To overcome this issue, we introduce the concept of an *entropy-preserving PRG (EP-PRG)*. This is a PRG that expands the seed by $O(\log n)$ bits, while ensuring that the output of the PRG looses at most $O(\log n)$ bits of *Shannon entropy*—it will be important for the sequel that we rely on Shannon entropy as opposed to min-entropy. In essence, the PRG preserves (up to an additive term of $O(\log n)$) the entropy in the seed s. We next show that any good heuristic $\mathcal{H}$ for K^t can break such an EP-PRG. The key point is that since the output of the PRG is entropy preserving, by an averaging argument, there exists an $1/n$ fraction of "good" seeds S such that conditioned on the seed belonging to S, the output of the PRG has *min-entropy* $n - O(\log n)$. This means that the probability that $\mathcal{H}$ fails to compute K^t on outputs of the PRG, conditioned on picking a "good" seed, can increase at most by a factor $poly(n)$. We conclude that $\mathcal{H}$ can be used to determine (with sufficiently high probability) the K^t-complexity for both random strings and for outputs of the PRG.

EP-PRGs from OWFs We start by noting that the standard Blum-Micali-Goldreich-Levin [BM84, GL89] PRG construction from one-way *permutations* is entropy preserving. To see this, recall the construction: $G_f(s) = f(s)||GL(s)$ where f is a one-way permutation and $GL(s)$ is a hardcore function for f—by [GL89], every one-way permutation can be modified into a one-way permutation that has a hardcore function that outputs $O(\log n)$ bits. Since f is a permutation, the output of the PRG fully determines the input and thus there is actually no entropy loss. We next show that the PRG construction of [HILL99, Gol01, YLW15] from *regular* OWFs also is an EP-PRG. We refer to a function f as being r-regular if for every $x \in \{0,1\}^*$, $f(x)$ has between $2^{r(n)-1}$ and $2^{r(n)}$ many preimages. Roughly speaking, the construction applies pairwise independent hash functions (that act as strong extractors) H_1, H_2 to both the input and output of the OWF (parametrized to match the regularity r) to "squeeze" out randomness from both the input and the output, and finally also applies a hardcore function that outputs $O(\log n)$ bits: $G^r_f(s||H_1||H_2) = H_1||H_2||H_1(s)||H_2(f(s))||GL(s)$. As already shown in [Gol01] (see also [YLW15]), the output of the function excluding the hardcore bits is actually $1/n^2$-close to uniform in statistical distance (this follows directly from the Leftover Hash Lemma [HILL99, Vad12]), and this implies (again using an averaging argument) that the Shannon entropy of the output is at least $n - O(\log n)$, thus the construction is an EP-PRG. We finally note that this construction remains both secure and entropy preserving even if the input domain of the function f is not $\{0,1\}^n$, but rather *any* set S of size $2^n/n$; this will be useful to us shortly.

Unfortunately, constructions of PRGs from OWFs [HILL99, Hol06, HHR06, HRV10] are not entropy preserving as far as we can tell. We, however, remark that to prove that K^t is HoA, we do not actually need a "full-fledged" EP-PRG: Rather, it suffices to have a "weak" EP-PRG G, where there exists some event E such that (1) conditioned on E, $G(\mathcal{U}_n)$ has Shannon entropy $n - O(\log n)$, and (2) conditioned on E, $G(\mathcal{U}_n)$ is pseudorandom. We next show how to adapt the above construction to yield a weak EP-PRG from any OWF. Consider $G(i||s) = G^i_f(s)$ where $|s| = n$ and $|i| = \log n$. We remark that for any function f, there exists some regularity i^* such that at least a fraction $1/n$ of inputs x have (approximate) regularity i^*. Let S_{i^*} denote the set of these x's. Clearly, $|S| \geq 2^n/n$; thus, by the above argument, $G^{i^*}_f(\mathcal{U}_N \mid S)$ is both pseudorandom and has entropy $n - O(\log n)$. Finally, consider the event E that $i = i^*$ and $s \in S_{i^*}$. By definition, $G(\mathcal{U}_{\log n}||\mathcal{U}_n \mid E)$ is identically distributed to $G^{i^*}_f(\mathcal{U}_N|S)$, and thus G is a weak EP-PRG from any OWF.

2 Preliminaries

We assume familiarity with basic concepts such as Turing machines, polynomial-time algorithms, probabilistic polynomial-time algorithms (PPT), non-uniform polynomial-time and non-uniform PPT algorithms. A function μ is said to be *negligible* if for every polynomial $p(\cdot)$ there exists some n_0 such that for all $n > n_0$, $\mu(n) \leq \frac{1}{p(n)}$. A *probability ensemble* is a sequence of random variables $A = \{A_n\}_{n\in\mathbb{N}}$. We let $\mathcal{U}_n$ the uniform distribution over $\{0,1\}^n$.

2.1 One-way functions

We recall the definition of one-way functions [DH76]. Roughly speaking, a function f is one-way if it is polynomial-time computable, but hard to invert for PPT attackers. The standard (cryptographic) definition of a one-way function (see e.g., [Gol01]) requires every PPT attacker to fail (with high probability) on all sufficiently large input lengths.

Definition 2.1. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a* one-way function (OWF) *if for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow \{0,1\}^n; y = f(x) : A(1^n, y) \in f^{-1}(f(x))] \leq \mu(n)$$

We may also consider a weaker notion of a "weak one-way function", where we only require all PPT attackers to fail with inverse polynomial probability:

Definition 2.2. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a α*-weak one-way function (α-weak OWF) *if for every* PPT *algorithm $\mathcal{A}$, for all sufficiently large $n \in N$,*

$$\Pr[x \leftarrow \{0,1\}^n; y = f(x) : A(1^n, y) \in f^{-1}(f(x))] < 1 - \alpha(n)$$

We say that f is simply a weak one-way function (weak OWF) *if there exists some polynomial $q > 0$ such that f is a $\frac{1}{q(\cdot)}$-weak OWF.*

Yao's hardness amplification theorem [Yao82] shows that any weak OWF can be turned into a (strong) OWF.

Theorem 2.3. *Assume there exists a weak one-way function. Then there exists a one-way function.*

2.2 K^t-Complexity

Let U be some fixed Turing machine, and let $U(M, 1^t)$ be the output of the Turing machine M when M is simulated on U for t steps. The t-time bounded Kolmogorov Complexity (K^t-Complexity) [Sip83, Tra84, Ko86] of a string x, $K^t(x)$ is defined as the length of the shortest machine M that outputs x (when running on the universal turing machine U) within $t(|x|)$ steps. More formally,

$$K^t(x) = \min_M \{|M| : U(M, 1^{t(|x|)}) = x\}.$$

A trivial observation about K^t-complexity is that the length of a string x essentially (up to an additive constant) bounds the K^t-complexity of the string; this follows by considering the program Π_x that has x hard-coded and simply outputs it.

Fact 2.1. *There exists a constant c such that for every function $t(n) > 2n$, for every $x \in \{0,1\}$ it holds that $K^t(x) \leq |x| + c$.*

2.3 Average-case Hard Functions

We turn to defining what it means for a function to be average-case hard (for PPT algorithms).

Definition 2.4. *We say that a function $f : \{0,1\}^* \to \{0,1\}^*$ is α hard-on-average (α-HoA) if for all* PPT *heuristic $\mathcal{H}$, for all sufficiently large $n \in N$,*

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = f(x)] < 1 - \alpha(|n|)$$

In other words, there does not exists a PPT "heuristic" $\mathcal{H}$ that computes f with probability $1 - \alpha(n)$ for infinitely many $n \in N$.

2.4 Computational Indistinguishability

We recall the definition of (computational) indistinguishability [GM84].

Definition 2.5 (Indistinguishability). *Two ensembles $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ are said to be $\mu(\cdot)$-*indistinguishable*, if for every probabilistic machine D (the "distinguisher") whose running time is polynomial in the length of its first input, there exist some $n_0 \in \mathbb{N}$ so that for every $n \geq n_0$:*

$$|\Pr[D(1^n, A_n) = 1] - \Pr[D(1^n, B_n) = 1]| < \mu(n)$$

We say that $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ simply indistinguishable *if they are $\frac{1}{p(\cdot)}$-indistinguishable for every polynomial $p(\cdot)$.*

2.5 Statistical Distance and Shannon Entropy

For any two random variables X and Y defined over some set $\mathcal{V}$, we let $\mathsf{SD}(X,Y) = \frac{1}{2}\sum_{v\in\mathcal{V}} |\Pr[X = v] - \Pr[Y = v]|$ denote the *statistical distance* between X and Y. For a random variable X, let $H(X) = \mathsf{E}[\log \frac{1}{\Pr[X=x]}]$ denote the (Shannon) entropy of X, and let $H_\infty(X) = \min_{x\in Supp(X)} \log \frac{1}{\Pr[X=x]}$ denote the *min entropy* of X. The following simple lemma will be useful to us.

Lemma 2.2. *For every $n \geq 4$, the following holds. Let X be a random variable over $\{0,1\}^n$ such that $\mathsf{SD}(X, \mathcal{U}_n) \leq \frac{1}{n^2}$. Then $H(X_n) \geq n - 2$.*

Proof: Let $S = \{x \in \{0,1\}^n : \Pr[X = x] \leq 2^{-(n-1)}\}$. Note that for every $x \notin S$, x will contribute at least

$$\frac{1}{2}\left(\Pr[X = x] - \Pr[U_n = x]\right) \geq \frac{1}{2}\left(\Pr[X = x] - \frac{\Pr[X = x]}{2}\right) = \frac{\Pr[X = x]}{4}$$

to $SD(X, \mathcal{U}_n)$. Thus,

$$\Pr[X \notin S] \leq 4 \cdot \frac{1}{n^2}.$$

Since for every $x \in S$, $\log \frac{1}{\Pr[X=x]} \geq n - 1$ and the probability that $X \in S$ is at least $1 - 4/n^2$, it follows that

$$H(X) \geq \Pr[X \in S](n-1) \geq (1 - \frac{4}{n^2})(n-1) \geq n - \frac{4}{n} - 1 \geq n - 2.$$

■

3 OWFs from Avg-case K^t-Hardness

Theorem 3.1. *Assume there exists polynomials $t(n) > 2n, p(n) > 0$ such that K^t is $\frac{1}{p(\cdot)}$-HoA. Then there exists a weak OWF f (and thus also a OWF).*

Proof: Let c be the constant from Fact 2.1. Consider the function $f : \{0,1\}^{n+c+log(n+c)} \rightarrow \{0,1\}^n$, which given an input $\ell||M'$ where $|\ell| = \log(n+c)$ and $|M'| = n+c$, outputs $\ell||U(M, 1^{t(n)})$ where M is the ℓ-bit prefix of M'. This function is only defined over some inputs lengths, but by an easy padding trick, it can be transformed into a function f' defined over all input lengths, such that if f is (weakly) one-way (over the restricted input lengths), then f' will be (weakly) one-way (over all input lengths): $f'(x')$ simply truncates its input x' (as little as possible) so that the (truncated) input x now becomes of length $m = n + c + log(n + c)$ for some n and output $f(x)$.

We now show that if K^t is $\frac{1}{p(\cdot)}$-HoA, then f is a $\frac{1}{q(\cdot)}$-weak OWF, where $q(n) = 2^{2c+3}np(n)^2$, which concludes the proof of the theorem. Assume for contradiction that f is not a $\frac{1}{q(\cdot)}$-weak OWF. That is, there exists some PPT attacker $\mathcal{A}$ that inverts f with probability at least $1 - \frac{1}{q(n)} \leq 1 - \frac{1}{q(m)}$ for infinitely many $m = n + c + log(n + c)$. Fix some such $m, n > 2$. By an averaging argument, except for a fraction $\frac{1}{2p(n)}$ of random tapes r for $\mathcal{A}$, the *deterministic* machine $\mathcal{A}_r$ (i.e., machine $\mathcal{A}$ with randomness fixed to r) fails to invert f with probability at most $\frac{2p(n)}{q(n)}$. Fix some such "good" randomness r for which $\mathcal{A}_r$ succeeds to invert f with probability $1 - \frac{2p(n)}{q(n)}$.

We next show how to use $\mathcal{A}_r$ to approximate K^t over random inputs $z \in \{0,1\}^n$. Our heuristic $\mathcal{H}_r(z)$ runs $\mathcal{A}_r(i||z)$ for all $i \in [n + c]$ where i is represented as an $\log(n + c)$ bit string, and outputs the length of the smallest program M output by $\mathcal{A}_r$ that produces the string z within $t(n)$ steps. Let S be the set of strings $z \in \{0,1\}^n$ for which $\mathcal{H}_r(z)$ fails to compute $K^t(z)$. Note that $\mathcal{H}_r$ thus fails with probability

$$fail_r = \frac{|S|}{2^n}.$$

Consider any string $z \in S$ and let $w = K^t(z)$ be its K^t-complexity. By Fact 2.1, we have that $w \leq n + c$. Since $\mathcal{H}_r(z)$ fails to compute $K^t(z)$, $\mathcal{A}_r$ must fail to invert $(w||z)$. But, since $w \leq n + c$, the output $(w||z)$ is sampled with probability

$$\frac{1}{n+c} \cdot \frac{1}{2^{|w|}} \geq \frac{1}{(n+c)} \frac{1}{2^{n+c}} \geq \frac{1}{n2^{2c+1}} \cdot \frac{1}{2^n}$$

in the one-way function experiment, so $\mathcal{A}_r$ must fail with probability at least

$$|S| \cdot \frac{1}{n2^{2c+1}} \cdot \frac{1}{2^n} = \frac{1}{n2^{2c+1}} \cdot \frac{|S|}{2^n} = \frac{fail_r}{n2^{2c+1}}$$

which by assumption (that $\mathcal{A}_r$ is a good inverter) is at most that $\frac{2p(n)}{q(n)}$. We thus conclude that

$$fail_r \leq \frac{2^{2c+2}np(n)}{q(n)}$$

Finally, by a Union Bound, we have that $\mathcal{H}$ (using a uniform random tape r) fails in computing K^t with probability at most

$$\frac{1}{2p(n)} + \frac{2^{2c+2}np(n)}{q(n)} = \frac{1}{2p(n)} + \frac{2^{2c+2}np(n)}{2^{c+3}np(n)^2} = \frac{1}{p(n)}.$$

Thus, $\mathcal{H}$ computes K^t with probability $1 - \frac{1}{p(n)}$ for infinitely many $n \in \mathbb{N}$, which contradicts the assumption that K^t is $\frac{1}{p(\cdot)}$-HoA. ■

4 Avg-case K^t-Hardness from OWFs

We introduce the notion of a (weak) *entropy-preserving* pseudo-random generator (EP-PRG) and next show (1) the existence of a weak EP-PRG implies that K^t is hard-on-average, and (2) OWFs imply weak EP-PRGs.

4.1 Entropy-preserving PRGs

We start by defining the notion of a weak Entropy-preserving PRG.

Definition 4.1. *An efficiently computable function* $g : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$ *is a* weak entropy-preserving pseudorandom generator (weak EP-PRG) *if there exists a sequence of events* $= \{E_n\}_{n\in\mathbb{N}}$ *and a constant* α *(referred to as the* entropy-loss constant*) such that the following conditions hold:*

- **(pseudorandomness):** $\{g(\mathcal{U}_n|E_n)\}_{n\in\mathbb{N}}$ *and* $\{\mathcal{U}_{n+\gamma \log n}\}_{n\in\mathbb{N}}$ *are* $(1/n^2)$*-indistinguishable;*

- **(entropy-preserving):** *For all sufficiently large $n \in \mathbb{N}$, $H(g(\mathcal{U}_n|E_n)) \geq n - \alpha \log n$.*

If for all n, $E_n = \{0,1\}^n$ (i.e., there is no conditioning), we say that g is an entropy-preserving pseudorandom generator (EP-PRG).

4.2 Avg-case K^t-Hardness from Weak EP-PRGs

Theorem 4.2. *Assume that for every $\gamma > 1$, there exists a weak EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$. Then there exists a polynomials $t(n) > 2n, p(n) > 0$ such that K^t is $\frac{1}{p(\cdot)}$-HoA.*

Proof: Let $\gamma = 4$, and let $g' : \{0,1\}^n \to \{0,1\}^{m'(n)}$ where $m'(n) = n + \gamma \log n$ be an weak EP-PRG. For any constant c, let $g^c(x)$ be a function that computes $g'(x)$ and truncates the last c bits. It directly follows that g^c is also a weak EP-PRG (since g' is so). Let $t(n) > 2n$ be a monotonically increasing polynomial that bounds the running time of g^c for every $c \leq \gamma + 1$, and let $p(n) = 2n^{2(\alpha+\gamma+1)}$.

Assume for contradiction that there exists some PPT $\mathcal{H}$ that computes K^t with probability $1 + \frac{1}{p(m)}$ for infinitely many $m \in \mathbb{N}$. Since $m'(n+1) - m'(n) \leq \gamma + 1$, there must exists some constant $c \leq \gamma + 1$ such that $\mathcal{H}$ succeeds with probability $1 + \frac{1}{p(m)}$ for infinitely many m of the form $m = m(n) = n + \gamma \log n - c$. Let $g(x) = g^c(x)$; recall that g is a weak EP-PRG (trivially, since g^c is so), and let $\alpha, \{E_n\}$, respectively, be the entropy loss constant and sequence of events, associated with it.

We next show that $\mathcal{H}$ can be used to break the weak EP-PRG g. Towards this, recall that a random string has high K^t-complexity with high probability: for $m = m(n)$, we have,

$$\Pr_{x \in \{0,1\}^m}[K^t(x) \geq m - \frac{\gamma}{2}\log n] \geq \frac{2^m - 2^{m-\frac{\gamma}{2}\log n}}{2^m} = 1 - \frac{1}{n^{\gamma/2}},$$

since the total number of Turing machines with length smaller than $m - \frac{\gamma}{2}\log n$ is only $2^{m-\frac{\gamma}{2}\log n}$. However, any string output by the EP-PRG, must have "low" K^t complexity: For every sufficiently large $n, m = m(n)$, we have that,

$$\Pr_{s \in \{0,1\}^n}[K^t(g(s)) \geq m - \frac{\gamma}{2}\log n] = 0,$$

since $g(s)$ can be represented by combining a seed s of length n with the code of g (of a constant length), and the running time of $g(s)$ is bounded by $t(|s|) = t(n) \leq t(m)$, so $K^t(g(s)) = n + O(1) = (m - \gamma \log n + c) + O(1) \leq m - \gamma/2 \log n$ for sufficiently large n.

Based on these observations, we now construct a PPT distinguisher $\mathcal{A}$ breaking g. On input $1^n, x$, where $x \in \{0,1\}^{m(n)}$, $\mathcal{A}(1^n, x)$ lets $w \leftarrow \mathcal{H}(x)$ and outputs 1 if $w \geq m(n) - \frac{\gamma}{2}\log n$ and 0 otherwise. Fix some n and $m = m(n)$ for which $\mathcal{H}$ succeeds with probability $\frac{1}{p(m)}$. The following two claims conclude that $\mathcal{A}$ distinguishes $\mathcal{U}_{m(n)}$ and $g(\mathcal{U}_n \mid E_n)$ with probability $\frac{1}{n^2}$.

Claim 1. *$\mathcal{A}(1^n, \mathcal{U}_m)$ outputs 1 with probability at least $1 - \frac{2}{n^{\gamma/2}}$.*

Proof: Recall that $\mathcal{A}(1^n, x)$ will output 1 if x is a string with K^t-complexity larger than $m - \gamma/2 \log n$ and $\mathcal{H}$ outputs a correct K^t-complexity for x. Thus,

$$\begin{aligned}
&\Pr[\mathcal{A}(1^n, x) = 1] \\
&\geq \Pr[K^t(x) \geq m - \gamma/2 \log n \wedge \mathcal{H} \text{ succeeds on } x] \\
&\geq 1 - \Pr[K^t(x) < m - \gamma/2 \log n] - \Pr[\mathcal{H} \text{ fails on } x] \\
&\geq 1 - \frac{1}{n^{\gamma/2}} - \frac{1}{p(n)} \\
&\geq 1 - \frac{2}{n^{\gamma/2}}.
\end{aligned}$$

where the probability is over a random $x \leftarrow \mathcal{U}_n$ and the randomness of $\mathcal{A}$ and $\mathcal{H}$. ■

Claim 2. *$\mathcal{A}(1^n, g(\mathcal{U}_n \mid E_n))$ outputs 1 with probability at most $1 - \frac{1}{n} + \frac{2}{n^{\alpha+\gamma}}$*

Proof: Recall that by assumption, $\mathcal{H}$ fails to computes $K^t(x)$ for random $x \in \{0,1\}^m$ with probability at most $\frac{1}{p(m)}$. By an averaging argument, for at least an $1 - \frac{1}{n^2}$ fraction of random tapes r for $\mathcal{H}$, the deterministic machine $\mathcal{H}_r$ fails to correctly compute K^t with probability at most $\frac{n^2}{p(m)}$. Fix some "good" randomness r such that $\mathcal{H}_r$ computes K^t with probability at least $1 - \frac{n^2}{p(m)}$. We next analyze the success probability of $\mathcal{A}_r$. Assume for contradiction that A_r outputs 1 with probability at least $1 - \frac{1}{n} + \frac{1}{n^{\alpha+\gamma}}$ on input $g(\mathcal{U}_n \mid E_n)$. Recall that (1) the entropy of $g(\mathcal{U}_n \mid E_n)$ is at least $n - \alpha \log n$ and (2) the quantity $-\log \Pr[g(\mathcal{U}_n \mid E_n) = y]$ is upper bounded by n for all $y \in g(\mathcal{U}_n \mid E_n)$ since $H_\infty(g(\mathcal{U}_n \mid E_n)) \le H_\infty(\mathcal{U}_n \mid E_n) \le H_\infty(\mathcal{U}_n) = n$. By an averaging argument, with probability at least $\frac{1}{n}$, a random $y \in g(\mathcal{U}_n \mid E_n)$ will satisfy

$$-\log \Pr[g(\mathcal{U}_n \mid E_n) = y] \ge (n - \alpha \log n) - 1.$$

We refer to an output y satisfying the above condition as being "good" and other y's as being "bad". Let $S = \{y \in g(\mathcal{U}_n \mid E_n) : \mathcal{A}_r(1^n, y) = 1 \wedge y \text{ is good}\}$, and let $S' = \{y \in g(\mathcal{U}_n \mid E_n) : \mathcal{A}_r(1^n, y) = 1 \wedge y \text{ is bad}\}$. Since

$$\Pr[\mathcal{A}_r(1^n, g(\mathcal{U}_n \mid E_n)) = 1] = \Pr[g(\mathcal{U}_n \mid E_n) \in S] + \Pr[g(\mathcal{U}_n \mid E_n) \in S'],$$

and $\Pr[g(\mathcal{U}_n \mid E_n) \in S']$ is at most the probability that $g(\mathcal{U}_n)$ is "bad" (which as argued above is at most $1 - \frac{1}{n}$), we have that

$$\Pr[g(\mathcal{U}_n \mid E_n) \in S] \ge \left(1 - \frac{1}{n} + \frac{1}{n^{\alpha+\gamma}}\right) - \left(1 - \frac{1}{n}\right) = \frac{1}{n^{\alpha+\gamma}}.$$

Furthermore, since for every $y \in S$, $\Pr[g(\mathcal{U}_n \mid E_n) = y] \le 2^{-n+\alpha \log n + 1}$, we also have,

$$\Pr[g(\mathcal{U}_n \mid E_n) \in S] \le |S| 2^{-n+\alpha \log n + 1}$$

So,

$$|S| \ge \frac{2^{n-\alpha \log n - 1}}{n^{\alpha+\gamma}} = 2^{n-(2\alpha+\gamma)\log n - 1}$$

However, for any $y \in g(\mathcal{U}_n \mid E_n)$, if $\mathcal{A}_r(1^n, y)$ outputs 1, then $\mathcal{H}_r(y) \neq K^t(y)$. Thus, the probability that $\mathcal{H}_r$ fails on a random $y \in \{0,1\}^m$ is at least

$$|S|/2^m = 2^{-(2\alpha+2\gamma)\log n - 1 + c} \ge 2^{-2(\alpha+\gamma)\log n - 1} = \frac{1}{2n^{2(\alpha+\gamma)}}$$

which contradicts the fact that $\mathcal{H}_r$ fails with probability at most $\frac{n^2}{p(m)} < \frac{1}{2n^{2(\alpha+\gamma)}}$ (since $n < m$).

We conclude that for every good randomness r, $\mathcal{A}_r$ outputs 1 with probability at most $1 - \frac{1}{n} + \frac{1}{n^{\alpha+\gamma}}$. Finally, by Union Bound (and since a random tape is bad with probability $\le \frac{1}{n^2}$), we have that the probability that $\mathcal{A}(g(\mathcal{U}_n \mid E_n))$ outputs 1 is at most

$$\frac{1}{n^2} + \left(1 - \frac{1}{n} + \frac{1}{n^{\alpha+\gamma}}\right) \le 1 - \frac{1}{n} + \frac{2}{n^2},$$

since $\gamma \ge 2$. ■

We conclude, recalling that $\gamma \ge 4$, that $\mathcal{A}$ distinguishes $\mathcal{U}_m$ and $g(\mathcal{U}_n \mid E_n)$ with probability of at least

$$\left(1 - \frac{2}{n^{\gamma/2}}\right) - \left(1 - \frac{1}{n} + \frac{2}{n^2}\right) \ge \left(1 - \frac{2}{n^2}\right) - \left(1 - \frac{1}{n} + \frac{2}{n^2}\right) = \frac{1}{n} - \frac{4}{n^2} \ge \frac{1}{n^2}$$

for infinitely many $n \in \mathbb{N}$. ■

4.3 Weak EP-PRGs from OWFs

In this section, we show how to construct a weak EP-PRG from any OWF. Towards this, we first recall the construction of [HILL99, Gol01, YLW15] of a PRG from a *regular* one-way function [GKL93].

Definition 4.3. *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called* regular *if there exists a function $r : \mathbb{N} \to \mathbb{N}$ such that for all sufficiently long $x \in \{0,1\}^*$,*

$$2^{r(|x|)-1} \leq |f^{-1} f(x)| \leq 2^{r(|x|)}.$$

We refer to r as the regularity of f.

As mentioned in the introduction, the construction, roughly speaking, proceeds in the following two steps given a OWF f with regularity r.

- By the Goldreich-Levin Theorem [GL89], for every $\gamma \geq 0$, f can be modified into a different regular OWF f' that has $\gamma \log n$-bit hard-core function GL.
- We next "massage" f' into a different OWF f'' having the property that there exists some $\ell(n) = n - O(\log n)$ such that $f''(\mathcal{U}_n)$ is statistically close to $\mathcal{U}_{\ell(n)}$—we will refer to such a OWF as being *dense*. This is done by applying a pairwise-independent hash functions to both the input and the output of f': $f''(x, h_1, h_2) = h_1||h_2||h_1(x)||h_2(f'(x))$, where h_1 and h_2 are appropriately parametrized to based on the regularity $r(|x|)$; more precisely h_1 outputs $r(|x|) - O(\log |x|)$ bits, and h_2 outputs $|x| - r(|x|) - O(\log |x|)$ bits. (Note that knowing the regularity is crucial so we know how many bits to "extract" from the input and the outputs.) This steps also ensures that $GL(x)$ is still hardcore.
- The final PRG is then $G(x, h_1, h_2) = f''(x, h_1, h_2)||GL(x)$.

(We note that the above two steps do not actually produce a "full-fledged" PRG as the statistical distance between the output of $f'(\mathcal{U}_n)$ and uniform is actually only $\frac{1}{\mathsf{poly}(n)}$ as opposed to being negligible. [Gol01] thus present a final amplification step to deal with this issue—for our purposes it will suffice to get a $\frac{1}{\mathsf{poly}(n)}$ indistinguishability gap so we will not be concerned about the amplification step.)

We remark that nothing in the above two steps requires f to be a one-way function defined on the domain $\{0,1\}^n$—both steps still work even for one-way functions defined over domain S that are different than $\{0,1\}^n$ as long as a lower bound on the size of the domain is efficiently computable (by a minor modification of the construction in Step 2 to account for the size of S).

Definition 4.4. *Let $\mathcal{S} = \{S_n\}$ be a sequence of sets such that $S_n \subseteq \{0,1\}^n$ and let $f : S_n \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a* one-way function over $\mathcal{S}$ ($\mathcal{S}$-OWF) *if for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow S_n; y = f(x) : A(1^n, y) \in f^{-1}(f(x))] \leq \mu(n)$$

We refer to f as being regular if it satisfies Definition 4.3 with the exception that we only quantify over all $n \in N$ and all $x \in S_n$ (as opposed to all $x \in \{0,1\}^n$).

We say that a *sequence of functions* $\{f_i\}_{i \in I}$ *is efficiently computable* if there exists a polynomial-time algorithm M such that $M(i, x) = f_i(x)$.

Lemma 4.1 (implicit in [Gol01, YLW15])**.** *Let $\mathcal{S} = \{S_n\}$ be a sequence of sets such that $S_n \subseteq \{0,1\}^n$, let s be an efficiently computable function such that $s(n) \leq \log |S_n|$, and let f be a $\mathcal{S}$-OWF with regularity $r(\cdot)$.*

Then, there exists some $\alpha' \geq 0$, some $c \geq 0$, an efficiently computable sequence of functions $\{f'_i\}_{i \in \mathbb{N}}$ such that for every $\gamma' \geq 0$, there exists an efficiently computable function $GL(\cdot)$ such that:

- **pseudorandomness:** *The ensembles of distributions* $\{x \leftarrow S_n, h \leftarrow \{0,1\}^{2n^c} : f'_{r(n)}(x,h)||GL(x)\}_{n\in\mathbb{N}}$ *and* $\{\mathcal{U}_{\ell'(n)}\}_{n\in\mathbb{N}}$ *are* $\frac{1}{\ell'(n)^2}$*-indistinguishable where* $\ell'(n) = s(n) + 2n^c - \alpha' \log n + \gamma' \log n$.

- $\ell(\cdot)$**-density:** *For all sufficiently large* n*, the distributions* $\{x \leftarrow S_n, h \leftarrow \{0,1\}^{2n^c} : f'_{r(n)}(x,h)\}$ *and* $\mathcal{U}_{\ell(n)}$ *are* $\frac{1}{\ell(n)^2}$*-close in statistical distance where* $\ell(n) = s(n) + 2n^c - \alpha' \log n$.

Proof: Recall that given a $\mathcal{S}$-OWF f which is regular over $\mathcal{S}$ with a $\gamma' \log n$-bit hardcore function GL[5], the construction has the form $f'_r(x, h_1, h_2) = h_1||h_2||h_1(x)||h_2(f(x))$ where $|x| = n, |h_1| = |h_2| = n^c$, and $h_1 : \{0,1\}^n \rightarrow \{0,1\}^{\ell_1(n)}$, $h_2 : \{0,1\}^n \rightarrow \{0,1\}^{\ell_2(n)}$, where c is a constant that does not depend on ℓ_1 and ℓ_2 (as long $\ell_1(n), \ell_2(n) < n$).

The proof in [Gol01, YLW15] does not rely on the input range being $\{0,1\}^n$—rather, the only thing needed to make the proof go through is that $\ell_1(n) \leq r(n) - d \log n$, and $\ell_2(n) \leq s(n) - r(n) - d \log n$ for some sufficiently large d—this makes sure that there is enough min-entropy in both the input and the output to ensure that the extractors h_1, h_2 work properly.

The function f'_r thus maps $n' = n + 2n^c$ bits to $2n^c + s(n) - 2d \log n$ bits. ■

We start by observing that every OWF actually is a regular $\mathcal{S}$-OWFs for a sufficiently large $\mathcal{S}$.

Lemma 4.2. *Let* f *be an one way function. There exists an integer function* $r(\cdot)$ *and a sequence of sets* $\mathcal{S} = \{S_n\}$ *such that* $S_n \subseteq \{0,1\}^n$, $|S_n| \geq \frac{2^n}{n}$*, and* f *is a* $\mathcal{S}$*-OWF with regularity* r.

Proof: The following simple claim is the crux of the proof:

Claim 3. *For every* $n \in \mathbb{N}$*, there exists an integer* $r_n \in [n]$ *such that*

$$\Pr[x \leftarrow \{0,1\}^n : 2^{r_n - 1} \leq |f^{-1}f(x)| \leq 2^{r_n}] \geq \frac{1}{n}.$$

Proof: For all $i \in [n]$, let

$$w(i) = \Pr[x \leftarrow \{0,1\}^n, 2^{i-1} \leq |f^{-1}f(x)| \leq 2^i].$$

Since for all x, the number of pre-images that map to $f(x)$ must be in the range of $[1, 2^n]$, we know that $\sum_{i=1}^n w(i) = 1$. By an averaging argument, there must exists such r_n that $w(r_n) \geq \frac{1}{n}$. ■

Let $r(n) = r_n$ for every $n \in N$, $S_n = \{x \in \{0,1\}^n : 2^{r(n)-1} \leq |f^{-1}f(x)| \leq 2^{r(n)}]\}$; regularity of f when the input domain is restricted to $\mathcal{S}$ follows directly. It only remains to show that f is a $\mathcal{S}$-OWF; this follows directly from the fact that the set S_n are dense in $\{0,1\}$. More formally, assume for contradiction that there exists a PPT algorithm $\mathcal{A}$ that inverts f with probability $\varepsilon(n)$ when the input is sampled in S_n. Since $|S_n| \geq \frac{2^n}{n}$, it follows that $\mathcal{A}$ can invert f with probability at least $\varepsilon(n)/n$ over uniform distribution, which is a contradiction (as f is a OWF). ■

We now show how to construct a weak EP-PRG from OWFs.

Theorem 4.5. *Assume that there exist one way functions. Then, for every* $\gamma > 1$*, there exists a weak EP-PRG* $g : \{0,1\}^{n'} \rightarrow \{0,1\}^{n' + \gamma \log n'}$.

Proof: By Lemma 4.1 and Lemma 4.2, there exists a sequence of sets $\mathcal{S} = \{S_n\}$ such that $S_n \subseteq \{0,1\}^n, |S_n| \geq \frac{2^n}{n}$, a $\mathcal{S}$-OWF f with regularity $r(\cdot)$, some $\alpha' \geq 0$, some $c \geq 0$, and an efficiently computable sequence of functions $\{f'_i\}_{i\in\mathbb{N}}$. Additionally, for every $\gamma' \geq 0$, there exists an efficiently computable function $GL(\cdot)$. Let $s(n) = n - \log n$ (to ensure that $s(n) \leq \log |S_n|$), and $\ell'(n) = s(n) + 2n^c - \alpha' \log n + \gamma' \log n$ be

[5] By the Goldreich-Levin Theorem [GL89], we can assume without loss of generality that any (regular) function has such a hardcore function.

as in Lemma 4.1. Consider the construction $g : \{0,1\}^{\log n+n+2n^c} \to \{0,1\}^{\ell'(n)}$ that takes an input (i,x,h) where $|i| = \log n, |x| = n, |h| = 2n^c$ and outputs $f'_i(x,h)||GL(x)$. Let $n' = \log n + n + 2n^c$ denote the input length of g. Let $\{E_{n'}\}$ be a sequence of events such that $E_{n'} = \{(r(n), x, h) : x \in S_n, h \in \{0,1\}^{2n^c}\}$.

Note that the two distributions, $g(\mathcal{U}_{n'} \mid E_{n'})$ and $\{x \leftarrow S_n, h \leftarrow \{0,1\}^{2n^c} : f'_{r(n)}(x,h)||GL(x)\}_{n\in\mathbb{N}}$, are identically distributed. It follows from Lemma 4.1 that $\{g(\mathcal{U}_{n'} \mid E_{n'})\}_{n\in\mathbb{N}}$ and $\{\mathcal{U}_{\ell'(n)}\}_{n\in\mathbb{N}}$ are $\frac{1}{\ell'(n)^2}$-indistinguishable. Thus, g satisfies the pseudorandomness property of a weak EP-PRG.

We further show that the output of g preserves entropy. Let X_n be a random variable uniformly distributed in S_n. By Lemma 4.1, $f'_{r(n)}(X_n, \mathcal{U}_{2n^c})$ is $\frac{1}{\ell(n)^2}$-close to $\mathcal{U}_{\ell(n)}$ in statistical distance where $\ell(n) = s(n) + 2n^c - \alpha' \log n$. We apply Lemma 2.2 and obtain

$$H(f'_{r(n)}(X_n, \mathcal{U}_{2n^c})) \geq \ell(n) - 2.$$

Then it follows that

$$H(f'_{r(n)}(X_n, \mathcal{U}_{2n^c}), GL(X_n)) \geq H(f'_{r(n)}(X_n, \mathcal{U}_{2n^c})) \geq \ell(n) - 2.$$

Notice that $g(\mathcal{U}_{n'} \mid E_{n'})$ and $(f'_{r(n)}(X_n, \mathcal{U}_{2n^c}), GL(X_n))$ are identical distributions, so on inputs of length n', the entropy loss of g is $n' - (\ell(n) - 2) \leq (\alpha' + 3)\log n + 2 \leq (\alpha' + 4)\log n'$, thus g satisfies the entropy preserving property.

The function g maps $n' = \log n + n + 2n^c$ bits to $\ell'(n)$ bits, and it is thus at least $\ell'(n) - n' \geq (\gamma' - \alpha' - 2)\log n$ -bit expanding. Since $n' \leq n^{c+1}$ for sufficiently large n, if we pick $\gamma' > (c+1)\gamma + \alpha' + 2$, g will expand its input by at least $(\gamma' - \alpha' - 2)\log n \geq (c+1)\gamma \log n \geq \gamma \log n'$ bits.

Finally, notice that although g is only defined over some input lengths, by taking "extra" bits in the input and appending them to the output, g can be transformed to a weak EP-PRG g' defined over all input lengths: $g'(x')$ finds a prefix x of x' as long as possible such that $|x|$ is of the form $n' = \log n + n + 2n^c$ for some n, rewrites $x' = x||y$, and outputs $g(x)||y$. The entropy preserving and the psuedorandomness property of g' follows directly; finally, note that if $|x'|$ is sufficiently large, it holds that $n^{c+1} \geq |x'|$, and thus by the same argument as above, g' will also expand its input by at least $\gamma \log |x'|$ bits.

■

References

[ABK+06] Eric Allender, Harry Buhrman, Michal Kouckỳ, Dieter Van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108, 1996.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. On the cryptographic applications of random functions. In *CRYPTO*, pages 276–288, 1984.

[GKL93] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. *SIAM Journal on Computing*, 22(6):1163–1175, 1993.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol01] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[Gur89] Yuri Gurevich. The challenger-solver game: variations on the theme of p=np. In *Logic in Computer Science Column, The Bulletin of EATCS*. 1989.

[Har83] J. Hartmanis. Generalized kolmogorov complexity and the structure of feasible computations. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 439–445, Nov 1983.

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold. On the power of the randomized iterate. In *CRYPTO*, pages 22–40, 2006.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[HRV10] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:89, 2010.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 230–235, 1989.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory '95*, pages 134–147, 1995.

[KC00] Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79, 2000.

[Ko86] Ker-I Ko. On the notion of infinite pseudorandom sequences. *Theor. Comput. Sci.*, 48(3):9–33, 1986.

[Kol68] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2(1-4):157–168, 1968.

[Lev85] Leonid A. Levin. One-way functions and pseudorandom generators. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 363–365, 1985.

[Lev86] Leonid A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

[Lev03] L. A. Levin. The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[RSA83] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems (reprint). *Commun. ACM*, 26(1):96–99, 1983.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sip83] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 330–335. ACM, 1983.

[Sol64] R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1 – 22, 1964.

[Tra84] Boris A Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.

[Vad12] Salil P Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91, 1982.

[YLW15] Yu Yu, Xiangxue Li, and Jian Weng. Pseudorandom generators from regular one-way functions: New constructions with improved parameters. *Theor. Comput. Sci.*, 569:58–69, 2015.