Electronic Colloquium on Computational Complexity, Report No. 52 (2020)

# On One-way Functions and Kolmogorov Complexity

Yanyi Liu
Cornell University
yl2866@cornell.edu

Rafael Pass*
Cornell Tech
rafael@cornell.edu

April 14, 2020

**Abstract**

We prove the equivalence of two fundamental problems in the theory of computation:

- **Existence of one-way functions:** the existence of one-way functions (which in turn are equivalent to pseudorandom generators, pseudorandom functions, private-key encryption schemes, digital signatures, commitment schemes, and more).
- **Mild average-case hardness of $K^{\mathsf{poly}}$-complexity:** the existence of polynomials $t, p$ such that no $\mathsf{PPT}$ algorithm can determine the $t$-time bounded Kolmogorov Complexity, $K^t$, 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 characterizing "non-trivial" complexity-based Cryptography: *"Non-trivial" complexity-based Cryptography is possible iff $K^{\mathsf{poly}}$ is mildly hard-on average.*

*Cornell Tech. Supported in part by NSF Award SATC-1704788, NSF Award RI-1703846, and AFOSR Award FA9550-18-1-0267. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2019-19-020700006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

ISSN 1433-8092

# 1 Introduction

We prove the equivalence of two fundamental problems in the theory of computation: (a) the existence of one-way functions, and (b) mild 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 [BM88, HILL99], pseudorandom functions [GGM84], private-key encryption [GM84], digital signatures [Rom90], commitment schemes [Nao91], and more). Additionally, as observed by Impagliazzo [Gur89, Imp95], the existence of a OWF is 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). 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 characterizes the hardness of OWFs (and thus the feasibility of "non-trivial" complexity-based cryptography) has been a long-standing open problem.[1] This problem is particularly pressing given recent advances in quantum computing [AAB+19] and the fact 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 $K^{\mathsf{poly}}$-Complexity:** What makes the string 121212121212121 less random than 604848506683403574924? The notion of *Kolmogorov complexity* ($K$-complexity), introduced by Solomonoff [Sol64], Kolmogorov [Kol68] and Chaitin [Cha69], 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 for $t(n) = \mathsf{poly}(n)$ predates the theory of NP-completeness and 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], Ko [Ko86] and Hartmanis [Har83]. 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 *mildly hard-on-average (mildly HoA)* if there exists some polynomial $p(\cdot) > 0$ such that every PPT fails in computing $K^t(\cdot)$ for at least a $\frac{1}{p(\cdot)}$ fraction of $n$-bit strings $x$ for all sufficiently

[1] 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.*

large $n$, and that $K^{\mathsf{poly}}$ is mildly HoA if there exists some polynomial $t(n) \geq 2n$ such that $K^t$ is mildly HoA.

Our main result shows that the existence of OWFs is equivalent to mild average-case hardness of $K^{\mathsf{poly}}$. In doing so, we present the first natural (and well-studied) computational problem that characterizes the feasibility of "non-trivial" complexity-based cryptography.

**Theorem 1.1.** *The following are equivalent:*

- *One-way functions exists;*
- $K^{\mathsf{poly}}$ *is mildly hard-on-average.*

In other words,

> *"Non-trivial" Complexity-based Cryptography is feasible iff* $K^{\mathsf{poly}}$*-complexity is mildly hard-on-average.*

**On the Hardness of Approximating $K^{\mathsf{poly}}$-complexity** Our connection between OWFs and $K^t$-complexity has direct implications to the theory of $K^t$-complexity. Trakhtenbrot [Tra84] also discusses average-case hardness of the *approximate* $K^t$-complexity problem: the problem of, given a random $x$, outputting an "approximation" $y$ that is $\beta(|x|)$-close to $K^t(x)$ (i.e., $|K^t(x) - y| \leq \beta(|x|)$). He observes that there is a trivial heuristic approximation algorithm that succeeds with probability approaching 1 (for large enough $n$): Given $x$, simply output $|x|$. In fact, this trivial algorithm produces a $(d \log n)$-approximation with probability $\geq 1 - \frac{1}{n^d}$ over random $n$-bits string.[2] We note that our proof that OWFs imply mild average-case hardness of $K^{\mathsf{poly}}$ actually directly extends to show that $K^{\mathsf{poly}}$ is mildly-HoA also to $(d \log n)$-approximate. We thus directly get:

**Theorem 1.2.** *If* $K^{\mathsf{poly}}$ *is mildly hard-on-average, then for every constant* $d$*,* $K^{\mathsf{poly}}$ *is mildly hard-on-average to* $(d \log n)$*-approximate.*

In other words, the success probability of the "trivial" approximation algorithm cannot be significantly beaten unless $K^{\mathsf{poly}}$ can be *exactly* computed with overwhelming probability.

## 1.1 Related Work

We refer the reader to Goldreich's textbook [Gol01] for more context and applications of OWFs (and complexity-based cryptography in general); we highly recommend Barak's survey on candidate constructions of one-way functions [Bar17]. We refer the reader to the textbook of Li and Vitanyi [LV08] for more context and applications of Kolmogorov complexity; we highly recommend Allender's surveys on the history, and recent applications, of notions of time-bounded Kolmogorov complexity [All20a, All20b, All17].

**On Connections between $K^{\mathsf{poly}}$-complexity and OWFs** We note that some (partial) connections between $K^t$-complexity and OWFs already existed in the literature:

- Results by Kabanets and Cai [KC00] and Allender et al [ABK+06] show that the existence of OWFs implies that $K^{\mathsf{poly}}$ must be *worst-case* hard to compute; their results will be the starting point for our result that OWFs also imply *average-case hardness* of $K^{\mathsf{poly}}$.

[2] At most $2^{n-d\log n}$ out of $2^n$ strings have $K^t$-complexity that is smaller than $n - d\log n$.

- Allender and Das [AD17] show that every problem in **SZK** (the class of promise problems having statistical zero-knowledge proofs [GMR89]) can be solved in probabilistic polynomial-time using a $K^{\mathsf{poly}}$-complexity oracle. Furthermore, Ostrovsky and Wigderson [Ost91, OW93] show that if **SZK** contains a problem that is hard-on-average, then OWFs exists. In contrast, we show the existence of OWFs assuming only that $K^{\mathsf{poly}}$ is hard-on-average.

**On Worst-case to Average-case Reductions for $K^{\mathsf{poly}}$-complexity** We highlight a very elegant recent result by Hirahara [Hir18] that presents a worst-case to average-case reduction for $K^{\mathsf{poly}}$-complexity. Unfortunately, his result only gives average-case hardness w.r.t. *errorless heuristics*—namely, heuristics that always provide either the correct answer or output $\perp$ (and additionally only output $\perp$ with small probability). For our construction of a OWF, however, we require average-case hardness of $K^t$ also with respect to heuristics that may err (with small probability). Hirahara notes that it is an open problem to obtain a worst-case to average-case reductions w.r.t. heuristics that may err. Let us emphasize that average-case hardness w.r.t. errorless heuristics is a much weaker property that just "plain" average-case hardness (with respect to heuristics that may err): Consider a random 3SAT formula on $n$ variables with $1000n$ clauses. It is well-known that, with high probability, the formula is not be satisfiable. Thus, there is a trivial heuristic algorithm for solving 3SAT on such random instances: simply output "No". Yet, the question of whether there exists an efficient errorless heuristic for this problem is still open, and the non-existence of such an algorithm is implied by Feige's Random 3SAT conjecture [Fei02].

## 1.2 Proof outline

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

**OWFs from Avg-case $K^{\mathsf{poly}}$-Hardness** We show that if $K^t$ is mildly average-case hard for some $t(n) > 2n$, then a weak one-way function exists[3]; 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 $\ell$ is of length $\log(n + c)$ and $M'$ is of length $n + c$, that lets $M$ be the first $\ell$ 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 $\ell$ 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 $\ell$ (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 $\delta$, then the

[3] 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 $n$.

one-way function inverter must fail with probability at least $\frac{\delta}{O(n)}$, which concludes that $\delta$ must be small (as we assumed the inverter fails with probability significantly smaller than $\frac{1}{n}$).

**Avg-case $K^{\mathsf{poly}}$-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, for every sufficiently large polynomial $t(\cdot)$, 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 long as $t(n)$ is sufficiently greater than the running time of the PRG (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]

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 Regular 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:

$$G_f(s, h_{GL}) = f(s)||h_{GL}(s)$$

where $f$ is a one-way permutation and $h_{GL}$ is a hardcore function for $f$—by [GL89], we can select a random hardcore function $h_{GL}$ that output $O(\log n)$ 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 [GKL93, 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:

$$G_f^r(s||h_1||h_2||h_{GL}) = h_{GL}||h_1||h_2||[h_1(s)]_{r-O(\log n)}||[h_2(f(s))]_{n-r-O(\log n)}||h_{GL}(s),$$

[4] We note that, although it was not explictly pointed out, their argument actually also extends to show that $K^t$ does not have an *errorless* heuristic assuming the existence of PRGs. The point is that even on outputs of the PRG, an errorless heuristic must output either a small value or $\perp$ (and perhaps always just output $\perp$). But for random strings, the heuristic can only output $\perp$ with small probability. Dealing with heuristic that may err will be more complicated.

where $[a]_j$ means $a$ truncated to $j$ bits. As already shown in [Gol01] (see also [YLW15]), the output of the function excluding the hardcore bits is actually $1/\mathsf{poly}(n)$ -close to uniform in statistical distance (this follows directly from the Leftover Hash Lemma [HILL99, Vad12]), and this implies (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.

**Weak EP-PRGs from Any OWFs** 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 mildly HoA, we do not actually need a "full-fledged" EP-PRG: Rather, it suffices to have what we refer to as a *weak* EP-PRG $G$: a weak EP-PRG is an efficiently computable function $G$ having the property that there exists some event $E$ such that:

1. $G(\mathcal{U}_{n'} \mid E)$ has Shannon entropy $n - O(\log n)$;
2. $G(\mathcal{U}_{n'} \mid E)$ is indistinguishable from $\mathcal{U}_m$ for some $m \geq n' + O(\log n')$.

In other words, there exists some event $E$ such that conditionned on the event $E$, $G$ behaves likes an EP-PRG. We next show how to adapt the above construction to yield a weak EP-PRG from any OWF $f$. Consider $G(i||s||h_1, h_2, h_{GL}) = G_f^i(s, h_1, h_2, h_{GL})$ where $|s| = n$, $|i| = \log n$. and $i \in [n]$. We remark that for any function $f$, there exists some regularity $i^*$ such that at least a fraction $1/n$ of inputs $x$ have regularity $i^*$. Let $S_{i^*}$ denote the set of these $x$'s. Clearly, $|S| \geq 2^n/n$; thus, by the above argument, $G_f^{i^*}(\mathcal{U}_{n'} \mid S)$ 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, $G(\mathcal{U}_{\log n}||\mathcal{U}_n||\mathcal{U}_m \mid E)$ is identically distributed to $G_f^{i^*}(\mathcal{U}_{n'} \mid S)$, and thus $G$ is a weak EP-PRG from any OWF. For clarity, let us provide the full expanded description of the weak EP-PRG $G$:

$$G(i||s||h_1||h_2||h_{GL}) = h_{GL}||h_1||h_2||[h_1(s)]_{i-O(\log n)}||[h_2(f(s))]_{n-i-O(\log n)}||h_{GL}(s)$$

Note that this $G$ is *not* a PRG: if the input $i \neq i^*$ (which happens with probability $1 - \frac{1}{n}$), the output of $G$ may not be pseudorandom! But, recall that the notion of a *weak* EP-PRG only requires the output of $G$ to be pseudorandom *conditioned* on some event $E$ (while also being entropy preserving conditioned on the same event $E$).

## 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 $\mu$ 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.

**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 $\mu$ 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* [Yao82], where we only require all PPT attackers to fail with probability noticeably bounded away from 1:

**Definition 2.2.** *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. $f$ is said to be a* $\alpha$-weak one-way function ($\alpha$-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** ([Yao82])**.** *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 Universal 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 $\Pi_x$ that has $x$ hardcoded 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* $\alpha(\cdot)$ hard-on-average ($\alpha$-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 $n \in N$. We also consider what it means for a function to be average-case hard to *approximate*.

**Definition 2.5.** *We say that a function* $f : \{0,1\}^* \to \{0,1\}^*$ *is* $\alpha$ hard-on-average ($\alpha$-HoA) to $\beta(\cdot)$-approximate *if for all* PPT *heuristic* $\mathcal{H}$*, for all sufficiently large* $n \in N$*,*

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

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

Finally, we refer to a function $f$ as being *mildly* HoA (resp HoA to approximate) if there exists a polynomial $p(\cdot) > 0$ such that $f$ is $\frac{1}{p(\cdot)}$-HoA (resp. HoA to approximate).

## 2.4 Computational Indistinguishability

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

**Definition 2.6.** *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* $n \ge n_0$*:*

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

*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 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 lemma will be useful to us.

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

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

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

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

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

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

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

■

## 3 The Main Theorem

**Theorem 3.1.** *The following are equivalent:*

*(a) The existence of one-way functions.*

*(b) The existence of a polynomial $t(n) > 2n$ such that $K^t$ is mildly hard-on-average.*

*(c) For every constant $d$, the existence of a polynomial $t_0(n)$ such that for every polynomial $t(n) \geq t_0(n)$, $K^t$ is mildly hard-on-average to $(d \log n)$-approximate.*

We prove Theorem 3.1 by showing that (b) implies (a) (in Section 4) and next that (a) implies (c) (in Section 5). Finally, (c) trivially implies (b).

## 4 OWFs from Mild Avg-case $K^t$-Hardness

**Theorem 4.1.** *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 $f : \{0,1\}^{n+c+\log(n+c)} \to \{0,1\}^n$, 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 $\ell$-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 compute $K^t$ with high probability 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 a $\log(n+c)$ 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. ■

# 5 Mild 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 (even to approximate), and (2) OWFs imply weak EP-PRGs.

## 5.1 Entropy-preserving PRGs

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

**Definition 5.1.** *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 $\alpha$ (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)*.*

## 5.2 Avg-case $K^t$-Hardness from Weak EP-PRGs

**Theorem 5.2.** *Assume that for every $\gamma > 1$, there exists a weak EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+\gamma\log n}$. Then, for every constant $d$, there exists a polynomial $t_0(n)$ such that for every polynomial $t(n) \geq t_0(n)$, $K^t$ is mildly hard-on-average to $(d\log n)$-approximate.*

**Proof:** Let $\gamma \geq max(8, 8d)$, and let $g' : \{0,1\}^n \to \{0,1\}^{m'(n)}$ where $m'(n) = n + \gamma\log n$ be a 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 $t_0(n)$ be a monotonically increasing polynomial that bounds the running time of $g^c$ for every $c \leq \gamma + 1$, let $t(n) \geq t_0(n)$ and let $p(n) = 2n^{2(\alpha+\gamma+1)}$.

Assume for contradiction that there exists some $\mathsf{PPT}$ $\mathcal{H}$ that $\beta$-approximates $K^t$ with probability $1 - \frac{1}{p(m)}$ for infinitely many $m \in \mathbb{N}$, where $\beta(n) = \gamma/8\log n \geq d\log n$. Since $m'(n+1) - m'(n) \leq \gamma + 1$, there must exists some constant $c \leq \gamma + 1$ such that $\mathcal{H}$ succeeds (to $\beta$-approximate $K^t$) 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,

$$\Pr_{x \in \{0,1\}^m}[K^t(x) \geq m - \frac{\gamma}{4}\log n] \geq \frac{2^m - 2^{m-\frac{\gamma}{4}\log n}}{2^m} = 1 - \frac{1}{n^{\gamma/4}}, \tag{1}$$

since the total number of Turing machines with length smaller than $m - \frac{\gamma}{4}\log n$ is only $2^{m-\frac{\gamma}{4}\log n}$. 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,

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

since $g(s)$ can be represented by combining a seed $s$ of length $n$ with the code of $g$ (of 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 $\mathsf{PPT}$ 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 $w \geq m(n) - \frac{3}{8}\gamma\log n$ 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 at least $\frac{1}{n^2}$.

**Claim 1.** *$\mathcal{A}(1^n, \mathcal{U}_m)$ outputs 1 with probability at least* $1 - \frac{2}{n^{\gamma/4}}$.

**Proof:** Note that $\mathcal{A}(1^n, x)$ will output 1 if $x$ is a string with $K^t$-complexity larger than $m - \gamma/4\log n$ and $\mathcal{H}$ outputs a $\gamma/8\log n$-approximation to $K^t(x)$. Thus,

$$\begin{aligned}
&\Pr[\mathcal{A}(1^n, x) = 1] \\
&\geq \Pr[K^t(x) \geq m - \gamma/4\log n \wedge \mathcal{H} \text{ succeeds on } x] \\
&\geq 1 - \Pr[K^t(x) < m - \gamma/4\log n] - \Pr[\mathcal{H} \text{ fails on } x] \\
&\geq 1 - \frac{1}{n^{\gamma/4}} - \frac{1}{p(n)} \\
&\geq 1 - \frac{2}{n^{\gamma/4}}.
\end{aligned}$$

where the probability is over a random $x \leftarrow \mathcal{U}_m$ 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 $(\gamma/8\log n)$-approximate $K^t(x)$ for a random $x \in \{0,1\}^m$ with probability at most $\frac{1}{p(m)}$. By an averaging argument, for at least a $1 - \frac{1}{n^2}$ fraction of random tapes $r$ for $\mathcal{H}$, the deterministic machine $\mathcal{H}_r$ fails to approximate $K^t$ with probability at most $\frac{n^2}{p(m)}$. Fix some "good" randomness $r$ such that $\mathcal{H}_r$ approximates $K^t$ 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 $H_\infty(g(\mathcal{U}_n \mid E_n)) \leq H_\infty(\mathcal{U}_n \mid E_n) \leq H_\infty(\mathcal{U}_n) = n$. By an averaging argument, with probability at least $\frac{1}{n}$, a random $y \in g(\mathcal{U}_n \mid E_n)$ will satisfy

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

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

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

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

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

So,

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

However, for any $y \in g(\mathcal{U}_n \mid E_n)$, if $\mathcal{A}_r(1^n, y)$ outputs 1, then by Equation 2, $\mathcal{H}_r(y) > K^t(y) + \gamma/8$, so $\mathcal{H}$ fails to output a good approximation. (This follows, since by Equation 2, $K^t(y) < n - \gamma/2 \log n$ and $\mathcal{A}_r(1^n, y)$ outputs 1 only if $\mathcal{H}_r(y) \geq n - \frac{3}{8}\gamma \log n$.)

Thus, the probability that $\mathcal{H}_r$ fails (to output a good approximation) on a random $y \in \{0,1\}^m$ is at least

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

which contradicts the fact that $\mathcal{H}_r$ fails with approximate $K^t$ 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 $\leq \frac{1}{n^2}$), we have that the probability that $\mathcal{A}(g(\mathcal{U}_n \mid E_n))$ outputs 1 is at most

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

since $\gamma \geq 2$. ■

We conclude, recalling that $\gamma \geq 8$, that $\mathcal{A}$ distinguishes $\mathcal{U}_m$ and $g(\mathcal{U}_n \mid E_n)$ with probability of at least

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

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

### 5.3 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 5.3.** *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 proceeds in the following two steps given a OWF $f$ with regularity $r$.

- We "massage" $f$ into a different OWF $\hat{f}$ having the property that there exists some $\ell(n) = n - O(\log n)$ such that $\hat{f}(\mathcal{U}_n)$ is statistically close to $\mathcal{U}_{\ell(n)}$—we will refer to such a OWF as being *dense*. This is done by applying pairwise-independent hash functions (acting as strong extractors) to both the input and the output of the OWF (parametrized to match the regularity $r$) to "squeeze" out randomness from both the input and the output.

$$\hat{f}(s||\sigma_1||\sigma_1) = \sigma_1||\sigma_2||[h_{\sigma_1}(s)]_{r-O(\log n)}||[h_{\sigma_2}(f(s))]_{n-r-O(\log n)}$$

  where $[a]_j$ means $a$ truncated to $j$ bits.

- We next modify $\hat{f}$ to include additional randomness in the input (which is also revealed in the output) to make sure the function has a hardcore function:

$$f'(s||\sigma_1||\sigma_2||\sigma_{GL}) = \sigma_{GL}||\hat{f}(s||\sigma_1||\sigma_1)$$

- We finally use $f'$ to construct a PRG $G^r$ by simply adding the the Goldreich-Levin hardcore bits [GL89], $GL$, to the output of the function $f'$:

$$G^r(s||\sigma_1||\sigma_2||\sigma_{GL}) = f'(s||\sigma_1||\sigma_2||\sigma_{GL})||GL(s||\sigma_1||\sigma_2, \sigma_{GL}))$$

(We note that the above steps do not actually produce a "fully secure" PRG as the statistical distance between the output of $\hat{f}(\mathcal{U}_n)$ and uniform is 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 steps requires $f$ to be a one-way function defined on the domain $\{0,1\}^n$— all three steps still work even for one-way functions defined over domains $S$ 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 1 to account for the size of $S$). Let us start by formalizing this fact.

**Definition 5.4.** *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 $\mu$ 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 5.3 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 *family 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 5.1** (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 an $\mathcal{S}$-OWF with regularity $r(\cdot)$. Then, there exists a constant $c \geq 1$ such that for every $\alpha', \gamma' \geq 0$, there exists an efficiently computable family of functions $\{f'_i\}_{i \in \mathbb{N}}$, and an efficiently computable function $GL$, such that the following holds:*

- **density:** *For all sufficiently large $n$, the distributions*

– $\left\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL})\right\}$, *and*

– $\mathcal{U}_{s(n)+3n^c-2\alpha' \log n}$

*are* $\frac{3}{n^{\alpha'/2}}$*-close in statistical distance.*

- **pseudorandomness:** *The ensembles of distributions,*

  – $\left\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL})||GL(x, \sigma_1, \sigma_2, \sigma_{GL})\right\}_{n\in\mathbb{N}}$, *and*

  – $\left\{\mathcal{U}_{s(n)+3n^c-2\alpha' \log n+\gamma' \log n}\right\}_{n\in\mathbb{N}}$

  *are* $\frac{4}{n^{\alpha'/2}}$*-indistinguishable.*

**Proof:** Given a $r(\cdot)$-regular $\mathcal{S}$-OWF $f$, the construction of $f'$ has the form

$$f'(s||\sigma_1||\sigma_1||\sigma_{GL}) = \sigma_{GL}||\sigma_1||\sigma_2||[h_{\sigma_1}(s)]_{r-\alpha' \log n}||[h_{\sigma_2}(f(s))]_{s(n)-r-\alpha' \log n}$$

where $|x| = n, |\sigma_1| = |\sigma_2| = |\sigma_c| = n^c$, and $GL(x, \sigma_1, \sigma_2, \sigma_{GL})$ is simply the Goldreich-Levin hardcore predicate [GL89] outputting $\gamma' \log n$ inner products between $x$ and vectors in $\sigma_{GL}$. The function $f'_r$ thus maps $n' = n + 3n^c$ bits to $3n^c + s(n) - 2\alpha' \log n$ bits, and once we add output of $GL$, the total output length becomes $3n^c + s(n) - 2\alpha' \log n + \gamma' \log n$ as required. The proof in [Gol01, YLW15] directly works to show that $\{f_i\}, GL$ satisfy the requirements stated in the theorem. (For the reader's convenience, we present a simple self-contained proof of this in Appendix A.[5]) ■

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

**Lemma 5.2.** *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 $\mathsf{PPT}$ 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). ■

By combining Lemma 5.1 and Lemma 5.2, we can directly get an EP-PRG defined over a subset $\mathcal{S}$. We next turn to showing how to instead get a *weak* EP-PRG that is defined over $\{0,1\}^n$.

[5] This proof may be of independent didactic interest as an elementary proof of the existence of PRGs from regular OWFs.

**Theorem 5.5.** *Assume that there exist one way functions. Then, for every $\gamma > 1$, there exists a weak EP-PRG $g : \{0,1\}^{n'} \to \{0,1\}^{n'+\gamma \log n'}$.*

**Proof:** By Lemma 5.2, 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 function $r(\cdot)$, and an $\mathcal{S}$-OWF $f$ with regularity $r(\cdot)$. Let $s(n) = n - \log n$ (to ensure that $s(n) \leq \log |S_n|$). By Lemma 5.1, there exists a constant $c$ such that for every $\alpha', \gamma' \geq 0$, there exists an efficiently computable family of functions $\{f'_i\}_{i \in \mathbb{N}}$, and an efficiently computable function $GL$ satisfying the *density* and *pseudorandomness* properties described in Lemma 5.1. Consider some $\alpha' \geq 8c$ and any $\gamma' \geq 0$. Let $\ell(n) = s(n) + 3n^c - 2\alpha' \log n$, $\ell'(n) = \ell(n) + \gamma' \log n$ and consider the function $G : \{0,1\}^{\log n + n + 3n^c} \to \{0,1\}^{\ell'(n)}$ defined as follows:

$$G(i, x, \sigma_1, \sigma_2, \sigma_{GL}) = f'_i(x, \sigma_1, \sigma_2, \sigma_{GL}) || GL(x, \sigma_1, \sigma_2, \sigma_{GL})$$

where $|i| = \log n, i \in [n], |x| = n, |\sigma_1| = |\sigma_2| = |\sigma_{GL}| = n^c$. Let $n' = n'(n) = \log n + n + 3n^c$ denote the input length of $G$. Let $\{E_{n'(n)}\}$ be a sequence of events where

$$E_{n'(n)} = \{i, x, \sigma_1, \sigma_2, \sigma_{GL} : i = r(n), x \in S_n, \sigma_1, \sigma_2, \sigma_{GL} \in \{0,1\}^{n^c}\}$$

Note that the two distributions,

- $\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL}) || GL(x, \sigma_1, \sigma_2, \sigma_{GL})\}_{n \in \mathbb{N}}$, and
- $G(\mathcal{U}_{n'} \mid E_{n'})$

are identically distributed. It follows from Lemma 5.1 that $\{G(\mathcal{U}_{n'} \mid E_{n'})\}_{n \in \mathbb{N}}$ and $\{\mathcal{U}_{\ell'(n)}\}_{n \in \mathbb{N}}$ are $\frac{4}{n^{\alpha/2}}$-indistinguishable. Note that for $\alpha \geq 8c$, we have that $\frac{4}{n^{\alpha/2}} \leq \frac{4}{n^{4c}} \leq \frac{1}{n'(n)^2}$ for sufficiently large $n$. 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 over $S_n$. By Lemma 5.1, $f'_{r(n)}(X_n, \mathcal{U}_{3n^c})$ is $\frac{4}{n^{\alpha/2}} \leq \frac{4}{n^{4c}} \leq \frac{1}{\ell(n)^2}$ close to $\mathcal{U}_{\ell(n)}$ in statistical distance for sufficiently large $n$. By Lemma 2.2 it thus holds that

$$H(f'_{r(n)}(X_n, \mathcal{U}_{3n^c})) \geq \ell(n) - 2.$$

It thus follows that

$$H(f'_{r(n)}(X_n, \mathcal{U}_{3n^c}), GL(X_n, \mathcal{U}_{3n^c})) \geq H(f'_{r(n)}(X_n, \mathcal{U}_{3n^c})) \geq \ell(n) - 2.$$

Notice that $G(\mathcal{U}_{n'} \mid E_{n'})$ and $(f'_{r(n)}(X_n, \mathcal{U}_{3n^c}), GL(X_n, \mathcal{U}_{3n^c}))$ are identically distributed, so on inputs of length $n' = n'(n)$, the entropy loss of $G$ is $n' - (\ell(n) - 2) \leq (2\alpha' + 2) \log n + 2 \leq (2\alpha' + 4) \log n'$, thus $G$ satisfies the entropy-preserving property (by setting the entropy loss $\alpha$ in EP-PRG to be $(2\alpha' + 4)$).

The function $G$ maps $n' = \log n + n + 3n^c$ bits to $\ell'(n)$ bits, and it is thus at least $\ell'(n) - n' \geq (\gamma' - 2\alpha' - 2) \log n$ -bit expanding. Since $n' \leq n^{c+1}$ for sufficiently large $n$, if we pick $\gamma' > (c+1)\gamma + 2\alpha' + 2$, $G$ will expand its input by at least $(\gamma' - 2\alpha' - 2) \log n \geq (c+1)\gamma \log n \geq \gamma \log n'$ bits.

Finally, notice that although $G$ is only defined over some input lengths $n = n'(n)$, 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 $n' = \log n + n + 3n^c$ 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. ■

## 6 Acknowledgements

We are grateful to Naomi Ephraim, Cody Freitag, Johan Håstad, Yuval Ishai and Ilan Komargodski for helpful comments.

## References

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

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

[AD17] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Inf. Comput.*, 256:2–8, 2017.

[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.

[All17] Eric Allender. The complexity of complexity. In *Computability and Complexity - Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, pages 79–94, 2017.

[All20a] Eric Allender. Ker-i ko and the study of resource-bounded kolmogorov complexity. In *Complexity and Approximation - In Memory of Ker-I Ko*, pages 8–18, 2020.

[All20b] Eric Allender. The new complexity landscape around circuit minimization. In *Language and Automata Theory and Applications - 14th International Conference, LATA 2020, Milan, Italy, March 4-6, 2020, Proceedings*, pages 3–16, 2020.

[Bar17] Boaz Barak. The complexity of public-key cryptography. In *Tutorials on the Foundations of Cryptography*, pages 45–77. 2017.

[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.

[Cha69] Gregory J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. *J. ACM*, 16(3):407–422, 1969.

[CW79] J Lawrence Carter and Mark N Wegman. Universal classes of hash functions. *Journal of computer and system sciences*, 18(2):143–154, 1979.

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

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 534–543, 2002.

[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.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[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.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 247–258, 2018.

[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.

[LV08] Ming Li and Paul M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

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

[Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*, pages 133–138, 1991.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Theory and Computing Systems, 1993*, pages 3–17, 1993.

[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.

# A Proof of Lemma 5.1

In this section we provide a proof of Lemma 5.1. As mentionned in the main body, the proof of this lemma readily follows using the proofs in [HILL99, Gol01, YLW15], but for the convenience of the reader, we provide a simple self-contained proof of the lemma (which may be useful for didactic purposes). We start by recalling the Leftover Hash Lemma [HILL99] and the Goldreich-Levin Theorem [GL89].

**The Leftover Hash Lemma** We recall the notion of a universal hash function [CW79].

**Definition A.1.** *Let $\mathcal{H}_m^n$ be a family of functions where $m < n$ and each function $h \in \mathcal{H}_m^n$ maps $\{0,1\}^n$ to $\{0,1\}^m$. We say that $\mathcal{H}_m^n$ is a* universal hash family *if (i) the functions $h_\sigma \in \mathcal{H}_m^n$ can be described by a string $\sigma$ of $n^c$ bits where $c$ is a universal constant that does not depend on $n$; (ii) for all $x \neq x' \in \{0,1\}^n$, and for all $y, y' \in \{0,1\}^m$*

$$\Pr[h_\sigma \leftarrow \mathcal{H}_m^n : h_\sigma(x) = y \ \text{ and } \ h_\sigma(x') = y'] = 2^{-2m}$$

It is well-known that truncation preserves pairwise independence; for completeness, we recall the proof:

**Lemma A.1.** *If $\mathcal{H}_m^n$ is a universal hash family and $\ell \leq n$, then $\mathcal{H}'^n_\ell = \{h_\sigma \in \mathcal{H}_m^n : [h_\sigma]_\ell\}$ is also a universal hash family.*

**Proof:** For every $x \neq x' \in \{0,1\}^n, y, y' \in \{0,1\}^\ell$,

$$\begin{aligned}
&\Pr[h_\sigma \leftarrow \mathcal{H}_m^n; [h_\sigma(x)]_\ell = y \ \text{ and } \ [h_\sigma(x')]_\ell = y'] \\
&= \sum_{z \in \{0,1\}^n, [z]_\ell = y} \ \sum_{z' \in \{0,1\}^n, [z']_\ell = y'} Pr[h_\sigma \leftarrow \mathcal{H}_m^n; h_\sigma(x) = z \ \text{ and } \ h_\sigma(x') = z'] \\
&= 2^{-2\ell}.
\end{aligned}$$

$\blacksquare$

Carter and Wegman demonstrate the existence of efficiently computable universal hash function families.

**Lemma A.2** ([CW79]). *There exists a polynomial-time computable function* $H : \{0,1\}^n \times \{0,1\}^{n^c} \to \{0,1\}^n$ *such that for every* $n$, $\mathcal{H}_n^n = \{h_\sigma : \sigma \in \{0,1\}^{n^c}\}$ *is a universal hash family, where* $h_\sigma : \{0,1\}^n \to \{0,1\}^n$ *is defined as* $h_\sigma(x) = H(x, \sigma)$.

We finally recall the Leftover Hash Lemma.

**Lemma A.3** (Leftover Hash Lemma (LHL) [HILL99]). *For any integers* $d < k \leq n$*, let* $\mathcal{H}_{k-d}^n$ *be a universal hash family where each* $h \in \mathcal{H}_{k-d}^n$ *maps* $\{0,1\}^n$ *to* $\{0,1\}^{k-d}$*. Then, for any random variable* $X$ *over* $\{0,1\}^n$ *such that* $H_\infty(X) \geq k$*, it holds that*

$$\mathsf{SD}((H_{k-d}^n, H_{k-d}^n(X)), (H_{k-d}^n, \mathcal{U}_{k-d})) \leq 2^{-\frac{d}{2}},$$

*where* $H_{k-d}^n$ *denotes a random variable uniformly distributed over* $\mathcal{H}_{k-d}^n$.

**Hardcore functions and the Goldreich-Levin Theorem** We recall the notion of a hardcore function and the Goldreich-Levin Theorem [GL89].

**Definition A.2.** *A function* $g : \{0,1\}^n \to \{0,1\}^{v(n)}$ *is called a* hardcore function for $f : \{0,1\}^n \to \{0,1\}^*$ over $\mathcal{S} = \{S_n \subseteq \{0,1\}^n\}_{n \in N}$ *if the following ensembles are indistinguishable:*

- $\{x \leftarrow S_n : f(x)||g(x)\}_{n\in\mathbb{N}}$
- $\{x \leftarrow S_n : f(x)||\mathcal{U}_{v(n)}\}_{n\in\mathbb{N}}$

While the Goldreich-Levin theorem is typically stated for one-way functions $f$, it actually applies to any randomized function $f(x,\mathcal{U}_m)$ of $x$ that *hides* $x$. Note that hiding is a weaker property than one-wayness (where the attacker is only required to find *any* pre-image, and not necessarily the pre-image $x$ we computed the function on). Such a version of the Goldreich-Levin theorem was explicitly stated in e.g., [HHR06] (using somewhat different terminology).

**Definition A.3.** *A function* $f : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^*$ *is said to be* entropically-hiding over $\mathcal{S} = \{S_n\}_{n\in\mathbb{N}}$ ($\mathcal{S}$-hiding) *if for every* $\mathsf{PPT}$ *algorithm* $\mathcal{A}$*, there exists a negligible function* $\mu$ *such that for all* $n \in \mathbb{N}$,

$$\Pr[x \leftarrow S_n, r \leftarrow \{0,1\}^{m(n)}; A(1^n, f(x,r)) = x] \leq \mu(n)$$

**Theorem A.4** ([GL89], also see Theorem 2.12 in [HHR06]). *There exists some* $c$ *such that for every* $\gamma$*, and every* $m(\cdot)$*, there exists a polynomial-time computable function* $GL : \{0,1\}^{n+m(n)+n^c} \to \{0,1\}^{\gamma \log n}$ *such that the following holds: Let* $\mathcal{S} = \{S_n \subseteq \{0,1\}^n\}_{n\in N}$ *and let* $f : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^*$ *be* $\mathcal{S}$*-hiding. Then* $GL$ *is a hardcore function for* $f' : \{0,1\}^n \times \{0,1\}^{m(n)} \times \{0,1\}^{n^c} \to \{0,1\}^*$*, defined as* $f'(x,r,\sigma) = \sigma||f(x,r)$.

Given these preliminaries, we are ready to present the proof of Lemma 5.1.

**Proof of Lemma 5.1** 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 $s(n) \le \log |S_n|$, and let $f : S_n \to \{0,1\}^n$ be a $\mathcal{S}$-OWF with regularity $r(n)$. By Lemma A.2 and Lemma A.1, there exists some constant $c$ and a polynomial-time computable function $H : \{0,1\}^n \times \{0,1\}^{n^c} \to \{0,1\}^n$ such that for every $n, m \ge n$, $\mathcal{H}_m^n = \{h'_\sigma : \sigma \in \{0,1\}^{n^c}\}$ is a universal hash family, where $h'_\sigma = [h_\sigma]_m$ and $h_\sigma(x) = H(x,\sigma)$. We consider a "massaged" function $f_i$, obtained by hashing the input and the output of $f$: $f_i : S_n \times \{0,1\}^{n^c} \times \{0,1\}^{n^c} \to \{0,1\}^{2n^c} \times \{0,1\}^{i-\alpha' \log n} \times \{0,1\}^{s(n)-i-\alpha' \log n}$

$$f_i(x, \sigma_1, \sigma_2) = \sigma_1 || \sigma_2 || [h_{\sigma_1}(x)]_{i-\alpha' \log n} || [h_{\sigma_2}(f(x))]_{s(n)-i-\alpha' \log n}$$

where $n = |x|$ and show that the function $\hat{f}(x, (\sigma_1, \sigma_2)) = f_{r(n)}(x, \sigma_1, \sigma_2)$ is $\mathcal{S}$-hiding.

**Claim 4.** *The function $\hat{f}(\cdot,\cdot)$ is $\mathcal{S}$-hiding.*

**Proof:** Assume for contradiction that there exists a PPT $A$ and a polynomial $p(\cdot)$ such that for infinitely many $n \in \mathbb{N}$,

$$\Pr[x \leftarrow S_n, \sigma_1, \sigma_2 \leftarrow \{0,1\}^{n^c} : \mathcal{A}(1^n, f_{r(n)}(x, \sigma_1, \sigma_2)) = x] \ge \frac{1}{p(n)}$$

That is,

$$\Pr[x \leftarrow S_n, \sigma_1, \sigma_2 \leftarrow \{0,1\}^{n^c} : \mathcal{A}(1^n, \sigma_1 || \sigma_2 || [h_\sigma(x)]_{r(n)-\alpha' \log n} || [h_{\sigma_2}(f(x))]_{s(n)-r(n)-\alpha' \log n}) = x] \ge \frac{1}{p(n)}.$$

We show how to use $\mathcal{A}$ to invert $f$. Consider the PPT $\mathcal{A}'(1^n, y)$ that samples $\sigma_1, \sigma_2 \leftarrow \{0,1\}^{n^c}$ and a "guess" $z \leftarrow \{0,1\}^{r(n)-\alpha' \log n}$, and outputs $\mathcal{A}(1^n, \sigma_1 || \sigma_2 || z || [h_{\sigma_2}(y)]_{s(n)-r(n)-\alpha' \log n})$. Since the guess is correct with probability $2^{-r(n)+\alpha' \log n} \ge 2^{-r(n)}$, we have that

$$\Pr[x \leftarrow S_n : \mathcal{A}'(1^n, f(x)) = x] \ge \frac{2^{-r(n)}}{p(n)}.$$

Since the any $y \in f(S_n)$ has at least most $2^{r(n)-1}$ pre-images (since $f$ is $r(n)$-regular over $\mathcal{S}$), we have that

$$\Pr[x \leftarrow S_n : \mathcal{A}'(1^n, f(x)) = x] \ge \Pr[x \leftarrow S_n : \mathcal{A}'(1^n, f(x)) \in f^{-1}(f(x))] \times 2^{-r(n)+1}.$$

Thus,

$$\Pr[x \leftarrow S_n : \mathcal{A}'(1^n, f(x)) \in f^{-1}(f(x))] \ge 2^{-r(n)+1} \times \Pr[x \leftarrow S_n : \mathcal{A}'(1^n, f(x)) = x] \ge \frac{1}{2p(n)}$$

which contradicts that $f$ is an $\mathcal{S}$-OWF. ■

Next, consider $f'_i(s, \sigma_1, \sigma_2, \sigma_{GL}) = \sigma_{GL} || f_i(s, \sigma_1, \sigma_2)$, and the hardcore function $GL$ guaranteed to exists by Theorem A.4. Since $\hat{f}$ is $\mathcal{S}$-hiding, by Theorem A.4, the following ensembles are indistinguishable:

- $\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL}) || GL(x, (\sigma_1, \sigma_2), \sigma_{GL})\}_{n \in \mathbb{N}}$
- $\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL}) || \mathcal{U}_{\gamma' \log n}\}_{n \in \mathbb{N}}$

We finally show that $\{x \leftarrow S_n, \sigma_1, \sigma_2, \sigma_{GL} \leftarrow \{0,1\}^{n^c} : f'_{r(n)}(x, \sigma_1, \sigma_2, \sigma_{GL})\}$ is $\frac{3}{n^{\alpha'/2}}$ close to uniform for every $n$, which will conclude the proof of both the pseudorandomness and the density properties by a hybrid argument. Let $X$ be a random variable uniformly distributed over $\mathcal{S}_n$, and let $R_1, R_2, R_{GL}$ be random variables uniformly distributed over $\{0,1\}^{n^c}$. Let

$$\texttt{REAL} = f'_{r(n)}(X, R_1, R_2, R_{GL}) = R_{GL}||R_1||R_2||[h_{R_1}(X)]_{r(n)-\alpha' \log n}, [h_{R_2}(f(X))]_{s(n)-r(n)-\alpha' \log n}$$

We observe:

- For every $y \in f(S_n)$, $H_\infty(X|f(X) = y) \geq r(n) - 1$ due to the fact that $f$ is $r(n)$-regular; by the LHL (i.e., Lemma A.3), it follows that $\texttt{REAL}$ is $\frac{2}{n^{\alpha'/2}}$ close in statistical distance to

$$\texttt{HYB}_1 = R_{GL}||R_1||R_2||\mathcal{U}_{r(n)-\alpha' \log n}||[h_{R_2}(f(X))]_{s(n)-r(n)-\alpha' \log n}$$

- $H_\infty(f(X)) \geq s(n) - r(n)$ due to the fact that $f$ is $r(n)$-regular and $|S_n| \geq s(n)$; by the LHL, it follows that $\texttt{HYB}_1$ is $\frac{1}{n^{\alpha'/2}}$ close in statistical distance to

$$\texttt{HYB}_2 = R_{GL}||R_1||R_2||\mathcal{U}_{r(n)-\alpha' \log n}||\mathcal{U}_{s(n)-r(n)-\alpha' \log n} = \mathcal{U}_{s(n)+3n^c-2\alpha' \log n}$$

Thus, $\texttt{REAL}$ is $\frac{3}{n^{\alpha'/2}}$-close to uniform, which concludes the proof.



ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il