On the Possibility of Basing Cryptography on $\mathsf{EXP} \neq \mathsf{BPP}$

Yanyi Liu[1] and Rafael Pass[2⋆]

[1] Cornell University, yl2866@cornell.edu
[2] Cornell Tech, rafael@cs.cornell.edu

Abstract. Liu and Pass (FOCS'20) recently demonstrated an equivalence between the existence of one-way functions (OWFs) and mild average-case hardness of the time-bounded Kolmogorov complexity problem. In this work, we establish a similar equivalence but to a different form of time-bounded Kolmogorov Complexity—namely, Levin's notion of Kolmogorov Complexity—whose hardness is closely related to the problem of whether $\mathsf{EXP} \neq \mathsf{BPP}$. In more detail, let $Kt(x)$ denote the Levin-Kolmogorov Complexity of the string x; that is, $Kt(x) = \min_{\Pi \in \{0,1\}^*, t \in \mathbb{N}} \{|\Pi| + \lceil \log t \rceil : U(\Pi, 1^t) = x\}$, where U is a universal Turing machine, and $U(\Pi, 1^t)$ denotes the output of the program Π after t steps, and let MKtP denote the language of pairs (x, k) having the property that $Kt(x) \leq k$. We demonstrate that:

- $\mathsf{MKtP} \notin \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$ (i.e., MKtP is infinitely-often *two-sided error* mildly average-case hard) iff infinititely-often OWFs exist.
- $\mathsf{MKtP} \notin \mathsf{Avg}_{\mathsf{neg}}\mathsf{BPP}$ (i.e., MKtP is infinitely-often *errorless* mildly average-case hard) iff $\mathsf{EXP} \neq \mathsf{BPP}$.

Thus, the only "gap" towards getting (infinitely-often) OWFs from the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$ is the seemingly "minor" technical gap between two-sided error and errorless average-case hardness of the MKtP problem. As a corollary of this result, we additionally demonstrate that any reduction from errorless to two-sided error average-case hardness for MKtP implies (unconditionally) that $\mathsf{NP} \neq \mathsf{P}$.

We finally consider other alternative notions of Kolmogorov complexity—including space-bounded Kolmogorov complexity and conditional Kolmogorov complexity—and show how average-case hardness of problems related to them characterize log-space computable OWFs, or OWFs in NC^0.

1 Introduction

A *one-way function* [DH76] (OWF) is a function f that can be efficiently computed (in polynomial time), yet no probabilistic polynomial-time (PPT) algo-

⋆ Supported in part by NSF Award SATC-1704788, NSF Award RI-1703846, AFOSR Award FA9550-18-1-0267, and a JP Morgan Faculty Award. This material is based upon work supported by DARPA under Agreement No. HR00110C0086. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

rithm 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 important 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 primitives and protocols (e.g., pseudorandom generators [BM88,HILL99], pseudorandom functions [GGM84], private-key encryption [GM84], digital signatures [Rom90], commitment schemes [Nao91], identification protocols [FS90], coin-flipping protocols [Blu82], and more). These primitives and protocols are often referred to as *private-key primitives*, or "Minicrypt" primitives [Imp95] as they exclude the notable task of public-key encryption [DH76,RSA83]. 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 OWFs can be based on some "standard" complexity-theoretic assumption is mostly wide open. Indeed, a central open problem, originating in the seminal work of Diffie and Hellman [DH76] is whether the existence of OWFs can be based on the assumptions that $\mathsf{NP} \neq \mathsf{P}$ or $\mathsf{NP} \neq \mathsf{BPP}$. Arguably, this is the most important open problem in the foundations of Cryptography. So far, however, most results in the literature have been negative. Notably, starting with the work by Brassard [Bra83] in 1983, a long sequence of works have shown various types of black-box separations between *restricted* types of OWF (e.g., one-way permutations) and NP-hardness (see e.g., [Bra83,BT03,AGGM06,GWXY10,Liv10,HMX10,BB15]). We emphasize, however, that these results only show limited separations: they either consider restricted types of one-way functions, or restricted classes of black-box reductions.[3]

In this work, our goal is to address an even more basic (and ambitious) problem: can we base Cryptography on the "super-weak" assumption that $\mathsf{EXP} \neq \mathsf{BPP}$:

Can the existence of OWFs be based on the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$*?*

While we (obviously) are not able to provide a full positive answer to this problem (which as we shall see later on, would imply that $\mathsf{NP} \neq \mathsf{P}$), we are able to show that the task of basing OWFs on the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$ boils down to (more precisely, is equivalent to) a seemingly minor technical problem regarding different notions of average-case w.r.t. Levin's notion of Kolmogorov Complexity [Lev73]. Towards explaining our main result, let us first review some recent connections between Cryptography and Kolmogorov Complexity.

[3] We highlight that a recent result by Pass and Venkitasubramaniam [PV20] takes a step towards a positive results, showing that to prove the existence of OWFs from average-case hardness of NP, it suffices to prove that average-case hardness of TFNP (rather than NP) implies the existence of OWFs.

1.1 Connections Between OWFs and Kolmogorov Complexity

What makes the string 12121212121212121 less random than 60484850668340357492? 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". The modern complexity-theoretic study of this problem goes back to Sipser [Sip83], Ko [Ko86] and Hartmanis [Har83].

A very recent result by Liu and Pass [LP20] shows that "mild" average-case hardness[4] of the time-bounded Kolmogorov complexity problem (when the time-bound is some polynomial) is *equivalent* to the existence of OWFs. While the time-bounded Kolmogorov complexity problem is in NP (when the time-bound is a polynomial), it is not known whether this problem is average-case complete for NP, thus their result falls short of basing OWFs on the assumption that NP is average-case hard (i.e., that there exists some problem in NP that is average-case hard w.r.t. some sampleable distribution over instances).

In this work, we will extend their work to consider other variants of the notion of "resource-bounded" Kolmogorov complexity [Kol68]. The central advantage of doing so will be that we will be able to base OWFs on the average-case hardness of some problem that is average-case complete for EXP! The only reason that this result falls short of basing OWF on $\mathsf{EXP} \neq \mathsf{BPP}$ is that the notion of average-case hardness in the EXP-completeness result is slightly different from the notion of average-case hardness for the "OWF-completeness" result. However, "morally", this result can be interpreted as an indication that the existence of OWFs is equivalent to $\mathsf{EXP} \neq \mathsf{BPP}$.

1.2 Characterizing Average-case Hardness of Levin-Kolmogorov Complexity

While the definition of time-bounded Kolmogorov complexity, K^t, is simple and clean, as noted by Leonid Levin [Lev73] in 1973, an annoying aspect of this notion is that it needs to be parametrized by the time-bound t. To overcome this issue, Levin proposed an elegant "non-parametrized" version of Kolmogorov complexity that directly incorporates the running time as a *cost*. To capture the idea

[4] By "mild" average-case hardness, we here mean that no PPT algorithm is able to solve the problem with probability $1 - \frac{1}{p(n)}$ on inputs of length n, for all polynomials $p(\cdot)$

that polynomial-time computations are "cheap", Levin's definition only charges logarithmically for running time. More precisely, let the Levin-Kolmogorov Complexity of the string, $Kt(x)$, be defined as follows:

$$Kt(x) = \min_{\Pi \in \{0,1\}^*, t \in \mathbb{N}} \{|\Pi| + \lceil \log t \rceil : U(\Pi, 1^t) = x\},$$

where U is a universal Turing machine, and we let $U(\Pi, 1^t)$ denote the output of the program Π after t steps. Note that, just like the standard notion of Kolmogorov complexity, $Kt(x)$ is bounded by $|x|+O(1)$—we can simply consider a program that has the string x hard-coded and directly halts.

Let MKtP denote the decisional Levin-Kolmogorov complexity problem; namely, the language of pairs (x, k) where $k \in \{0,1\}^{\lceil \log |x| \rceil}$ having the property that $Kt(x) \leq k$. MKtP is no longer seems to be in NP, as there may be strings x that can be described by a short program Π (with description size e.g., $n/10$) but a "largish" running time (e.g., $2^{n/10}$); the resulting string x thus would have small Kt-complexity ($n/5$), yet verifying that the witness program program Π indeed outputs x would require executing it which would take exponential time. In fact, Allender et al [ABK+06] show that MKtP actually is EXP-complete w.r.t. P/poly reductions; in other words, MKtP $\in$ P/poly if and only if EXP $\subseteq$ P/poly.

We will be studying (mild) average-case hardness of the MKtP problem, and consider two standard (see e.g. [BT08]) notions of average-case tractability for a language L with respect to the uniform distribution over instances:

- **2-sided error average-case heuristics:** We say that $L \in \mathsf{Heur_{neg}BPP}$ if for every polynomial $p(\cdot)$, there exists some PPT heuristic $\mathcal{H}$ that decides L (w.r.t. uniform n-bit strings) with probability $1 - \frac{1}{p(n)}$.
- **errorless average-case heuristics:** We say that $L \in \mathsf{Avg_{neg}BPP}$ if for every polynomial $p(\cdot)$, there exists some PPT heuristic $\mathcal{H}$ such that (a) for every instance x, with probability 0.9, $\mathcal{H}(x)$ either outputs $L(x)$ or $\perp$, and (b), $\mathcal{H}(x)$ outputs $\perp$ with probability at most $\frac{1}{p(n)}$ given uniform n-bits strings x.

In other words, the difference between an errorless and a 2-sided error heuristic $\mathcal{H}$ is that an errorless heuristic needs to (with probability 0.9 over its own randomness but not the instance x) output either $\perp$ (for "I don't know") or the correct answer $L(x)$, whereas a 2-sided error heuristic may simply make mistakes without "knowing it".

To better understand the class $\mathsf{Avg_{neg}BPP}$, it may be useful to compare it to the class $\mathsf{Avg_{neg}P}$ (languages solvable by deterministic errorless heuristics): $L \in \mathsf{Avg_{neg}P}$ if for every polynomial $p(\cdot)$, there exists some deterministic polynomial-time heuristic $\mathcal{H}$ such that (a) for every input x, $\mathcal{H}(x)$ outputs either $L(x)$ or $\perp$, and (b) the probability over uniform n-bit inputs x that $\mathcal{H}$ outputs $\perp$ is bounded by $\frac{1}{p(n)}$. In other words, the *only* way an errorless heuristic may make a "mistake" is by saying $\perp$ ("I don't know"); if it ever outputs a non-$\perp$ response, this response needs to be correct. (Compare this to a 2-sided error heuristic that only makes mistakes with a small probability, but we do not know when they happen). $\mathsf{Avg_{neg}BPP}$ is simply the natural "BPP-analog" of $\mathsf{Avg_{neg}P}$ where the heuristic is allowed to be randomized.

2-sided error average-case hardness of MKtP and OWFs. Our first main result shows that the characterization of [LP20] can be extended to work also w.r.t. MKtP. More precisely,

Theorem 1. $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ *iff infinitely-often OWFs exist.*

We highlight that whereas [LP20] characterized "standard" OWF, the above theorem only characterizes *infinitely-often* OWFs—i.e., functions that are hard to invert for infinitely many inputs lengths (as opposed to all input lengths). The reason for this is that [LP20] considered an "almost-everywhere" notion of average-case hardness of K^t, whereas the statement $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ only considers an infinitely-often notion of average-case hardness. (As we demonstrate in the full paper, we can also obtain a characterization of standard "almost-everywhere" OWFs by assuming that MKtP is "almost-everywhere" mildly average-case hard, but for simplicity, in the main body of the paper, we focus our attention on the more standard complexity-theoretic setting of infinitely-often hardness).

On a high-level, the proof of Theorem 1 follows the same structure as the characterization of [LP20]. The key obstacle to deal with is that since MKtP is not known to be in NP, there may not exists some polynomial time-bound that bounds the running-time of a program Π that "witnesses" the Kt-complexity of a string x; this is a serious issue as the OWF construction in [LP20] requires knowing such a running-time bound (and indeed, the running-time of the OWF depends on it). To overcome this issue, we rely on a new insight about Levin-Kolmogorov Complexity.

We say that the program Π is a *Kt-witness* for the string x if Π generates x within t steps while minimizing $|\Pi| + \log t$ among all other programs (i.e., Π is a witness for the Kt-complexity of x). The crucial observation (see Fact 31) is that for every $0 < \varepsilon < 1$, except for an ε fraction of n-bit strings x, x has a Kt-witness Π that runs in time $O(\frac{1}{\varepsilon})$. That is, "most" strings have a Kt-witness that has a "short" running time. To see this, recall that as mentionned above, for every string x, $Kt(x) \leq |x| + O(1)$; thus, every string $x \in \{0,1\}^n$ with a Kt-witnesses Π with running time exceeding $O(\frac{1}{\varepsilon})$, must satisfy that $|\Pi| + \log O(\frac{1}{\varepsilon}) \leq Kt(x) \leq n + O(1)$, so $|\Pi| \leq n + O(1) - \log(\frac{O(1)}{\varepsilon}) = n + O(1) + \log \varepsilon$. Since the length of Π is bounded by $n + O(1) + \log \varepsilon$, it follows that we can have at most $O(\varepsilon)2^n$ strings x where the Kt-witness for x has a "long" running time.

We can next use this observation to consider a more computationally tractable version of Kt-complexity where we cut off the machine's running time after $\frac{1}{\varepsilon}$ steps (where ε is selected as an appropriate polynomial), and next follow a similar paradigm as in [LP20].

Errorless average-case hardness of MKtP and $\mathsf{EXP} \neq \mathsf{BPP}$. We next show how to extend the result of Allender et al [ABK+06] to show that MKtP is not just EXP-complete in the worst-case, but also EXP-average-case complete;

furthermore, we are able to show completeness w.r.t. BPP (as opposed to $\mathsf{P/poly}$) reductions. We highlight, however, that completeness is shown in a "non-black-box" way (whereas [ABK+06] present a $\mathsf{P/poly}$ truthtable reduction). By non-black-box we here mean that we are not able to show how to use any algorithm that solves MKtP (on average) as an oracle (i.e., as a black-box) to decide EXP (in probabilistic polynomial time); rather, we directly show that if $\mathsf{MKtP} \in \mathsf{Avg_{neg}BPP}$, then $\mathsf{EXP} \subseteq \mathsf{BPP}$.[5]

Theorem 2. $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$ *iff* $\mathsf{EXP} \neq \mathsf{BPP}$*.*

Theorem 2 follows a similar structure as the EXP-completeness results of [ABK+06]. Roughly speaking, Allender et al observe that by the result of Nisan and Wigderson [NW94], the assumption that $\mathsf{EXP} \not\subseteq \mathsf{P/poly}$ implies the existence of a (subexponential-time computable) pseudorandom generator that fools polynomial-size circuits. But using a Kt-oracle, it is easy to break the PRG (as outputs of the PRG have small Kt-complexity since its running time is "small"). We first observe that the same approach can be extended to show that MKtP is (errorless) average-case hard w.r.t. polynomial-size circuits (under the assumption that $\mathsf{EXP} \not\subseteq \mathsf{P/poly}$). We next show that if we instead rely on a PRG construction of Impagliazzo and Wigderson [IW98], it suffices to rely on the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$ to show average-case hardness of MKtP w.r.t. PPT algorithms.

Interpreting the combination of Thm 1 and Thm 2. By combining Theorem 1 and Theorem 2, we get that the only "gap" towards getting (infinitely-often) one-way functions from the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$ is the seemingly "minor" technical gap between two-sided error and errorless average-case hardness of the MKtP problem (i.e., proving $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$). Furthermore, note that this "gap" *fully characterizes* the possibility of basing (infinitely-often) OWFs on the assumption that $\mathsf{EXP} \neq \mathsf{BPP}$: Any proof that $\mathsf{EXP} \neq \mathsf{BPP}$ implies infinitely-often OWFs also shows the implication $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$.

As a corollary of Theorem 1 and Theorem 2, we next demonstrate that the implication $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ implies that $\mathsf{NP} \neq \mathsf{P}$ (in fact, even average-case hardness of NP).

Theorem 3. *If* $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$*, then* $\mathsf{NP} \neq \mathsf{P}$*.*

This results can be interpreted in two ways. The pessimistic way is that closing this gap betwen 2-sided error, and errorless, heuristics will be very hard. The optimistic way, however, is to view it as a new and algorithmic approach towards proving that $\mathsf{NP} \neq \mathsf{P}$: To demonstrate that $\mathsf{NP} \neq \mathsf{P}$, it suffices to demonstrate that MKtP can be solved by an errorless heuristic, given access to a two-sided error heuristic for the same problem.

1.3 Space-bounded Notions of Kolmogorov Complexity

We additionally consider other alternative notions of resource-bounded Kolmogorov complexity. In more detail, we consider a space-bounded notion of

[5] This non-black box aspect of our results stems from its use of [IW98].

Kolmogorov complexity K^s and a space-bounded notion of *conditional* Kolmogorov complexity, and show that these notions, respectively, characterize log-space computable one-way functions, or one-way functions in NC^0.

Characterizing OWFs in Log-space The s-space bounded Kolmogorov complexity, $K^s(x)$, of a string $x \in \{0,1\}^*$ is defined as

$$K^s(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \forall i \in [|x|], U(\Pi(i), 1^{2^{s(|x|)}}) = x_i \text{ and } \Pi(i) \text{ uses at most } s(|x|) \text{ space}\}$$

(Since we will be limiting the amount of space, we consider a notion of Kolmogorov complexity where the program Π needs to output just bit x_i of the string x, given the index i as input.) Given some function $s(\cdot)$, define $\mathsf{MKSP}[s]$ analogously to MKtP. We will be interested in the regime where $s(n) = O(\log n)$. Using a proof that closely follows [LP20] (and observing that the components needed in this proof can be computed in log space), we obtain the following characterization of log-space computable OWFs.

Theorem 4. *Infinitely-often one-way functions in log-space exist if and only if* $\mathsf{MKSP}[O(\log n)] \notin \mathsf{Heur_{neg}BPP}$.

(We can also get a characterization of "standard" (i.e., almost-everywhere) OWFs in log-space if we assume that $\mathsf{MKSP}[s]$ is almost-everywhere average-case hard; see the full paper for more details.)

We remark that by the results of [AIK06], the existence of a log-space computable OWF implies a OWF that is uniform NC^0 computable; in fact, as observed by [RS21] (see Remark 31), by a slight tweak of the construction of [AIK06], one actually gets a OWF that is *log-space* uniform NC^0 computable. In other words, the existence of log-space computable OWFs is equivalent to the existence of log-space uniform NC^0-computable OWFs. Theorem 4 thus also characterizes OWFs computable in log-space uniform NC^0.[6]

Characterizing OWF in Uniform NC^0. We finally turn to consider the question of characterizing OWF in just uniform (as opposed to log-space uniform) NC^0. To do this, we consider generalization of space-bounded Kolmogorov complexity which considers a *conditional* notion of Kolmogorov complexity.

The *conditional Kolmogorov complexity* [ZL70,Lev73,Tra84,LM91] of a string x given the string str is the length of the shortest program Π that given the "auxiliary input" str outputs x. We here consider a variant of $\mathsf{MKSP}[s]$, which considers conditional Kolmogorov complexity instead of the (unconditional) version, and where the "auxiliary input" str is generated by some deterministic polynomial-time machine F. More precisely, given some Turing machine F, define the F-conditional $s(\cdot)$-space bounded Kolmogorov complexity, $cK^{F,s}(x)$, as

[6] We remark that this observation was added after becoming aware of [RS21].

follows:

$$cK^{F,s}(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \forall i \in [|x|], U(\Pi(i, \mathsf{str}), 1^{2^{s(|x|)}}) = x_i \\ \text{and } \Pi(i, \mathsf{str}) \text{ uses at most } s(|x|) \text{ space}\}$$

where $\mathsf{str} = F(1^{|x|})$. We next define a decisional version, $\mathsf{McKSP}[F, s]$, analogously to $\mathsf{MKSP}[s]$, and get the following theorem by appropriately generalizing the proof of Theorem 4 and leveraging the result of [AIK06]:

Theorem 5. *Infinitely-often OWFs in uniform* NC^0 *exist iff there exists some polynomial-time Turing machine* F *such that* $\mathsf{McKSP}[F, O(\log n)] \notin \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$.

(As before, we can also get a characterization of "standard" (i.e., almost-everywhere) OWFs in uniform NC^0 if we assume that McKSP is almost-everywhere average-case hard; see the full paper for more details.)

1.4 Concurrent Works

A concurrent and independent work by Hanlin and Santhanam [RS21] presents related but orthogonal characterizations of MKtP and OWFs in NC_0. W.r.t., MKtP, both works essentially show an equivalence between mild average-case hardness of MKtP and the existence of OWFs; we next show that errorless average-case hardness of MKtP is equivalent to $\mathsf{EXP} \neq \mathsf{BPP}$, whereas they instead consider an incomparable notion of two-sided error hardness with a "tiny" error and show that such average-case hardness of MKtP w.r.t. non-uniform polynomial-time adversaries is equivalent to the assumption that $\mathsf{EXP} \notin \mathsf{P/poly}$. W.r.t. OWF in NC_0, [RS21] shows that, surprisingly, an alternative notion of time-bounded Kolmogorov complexity, KT, [ABK$^+$06] which *charges* for running-time (as opposed to bounding it) characterizes OWFs in log-space uniform NC^0. In contrast, we present a characterization in terms of space-bounded Kolmogorov complexity (and also of uniform NC^0-computable OWFs).

Resource bounded notions of *conditional* Kolmogorov complexity are useful also in other (related) contexts. In a companion paper to the current work [LP21], we rely on a notion of time-bounded conditional Kolmogorov complexity to characterize OWFs; the advantage of using this notion is that [LP21] shows that the time-bounded conditional Kolmogorov complexity problem, McKTP, is NP-complete. Taken together, the current work and [LP21], demonstrate that the existence of OWFs can be characterized through the average-case hardness of (essentially) EXP-complete (this work) and NP-complete ([LP21]) languages.

We additionally note the concurrent and independent work of [ACM$^+$21] which bases OWFs on average-case hardness of different conditional Kolmogorov complexity style problem. Their conditional Kolmogorov complexity problem—which they show is NP-complete—instead considers a conditional variant of the KT notion of [ABK$^+$06]; [ACM$^+$21], however, they only show a one-directional implication between average-case hardness of their problem and OWFs (and only a weak converse in the other direction).

2 Preliminaries

We assume familiarity with basic concepts and computational classes such as Turing machines, polynomial-time algorithms, probabilistic polynomial-time (PPT) algorithms, NP, EXP, BPP, log-space (or alternatively L), and $\mathsf{P/poly}$. In this work, following [AIK06], we mostly consider *polynomial-time uniform* versions of NC^0 and $\mathsf{L/poly}$: we let *uniform* NC^0 be the class of functions[7] that admit polynomial-time uniform NC^0 circuits, and *uniform* $\mathsf{L/poly}$ be the class of functions computed by log-space Turing machines with a polynomial-time computable advice. 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 $\mathcal{U}_n$ denote the the uniform distribution over $\{0,1\}^n$. Given a string x, we let $[x]_j$ denote the first j bits of x.

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 that for every PPT attacker A, there exists some negligible function $\mu(\cdot)$ such that A only succeeds in inverting the function with probability $\mu(n)$ for *all* input lengths n. (That is, hardness holds "almost-everywhere".) We will also consider a weaker notion of an *infinitely-often* one-way function [OW93], which only requires that the success probability is bounded by $\mu(n)$ for *infinitely many* inputs lengths n. (That is, hardness only holds "infinitely-often".)

Definition 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 $n \in \mathbb{N}$,*

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

f is said to be an infinitely-often one-way function (ioOWF) *if the above condition holds for infinitely many $n \in \mathbb{N}$ (as opposed to all).*

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. *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 $n \in N$,*

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

[7] We abuse the notation and say that a function f is in a class $\mathcal{C}$ if each bit on the output of f is computable in $\mathcal{C}$.

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. f is said to be an* weak infinitely-often one-way function (weak ioOWF) *if the above condition holds for infinitely many $n \in \mathbb{N}$ (as opposed to all).*

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

Theorem 6 ([Yao82]). *Assume there exists a weak one-way function (resp. infinitely-often one-way function). Then there exists a one-way functions (resp. infinitely-often one-way function).*

We observe that Yao's construction remains in log-space (resp uniform $\mathsf{L/poly}$) if the weak one-way function it takes is in log-space (resp uniform $\mathsf{L/poly}$) [AIK06,Gol01].

2.2 Levin's Notion of Kolmogorov Complexity

Let U be some fixed Universal Turing machine that can emulate any Turing machine M with polynomial overhead. Given a description $\Pi \in \{0,1\}^*$ which encodes a pair (M, w) where M is a (single-tape) Turing machine and $w \in \{0,1\}^*$ is an input, let $U(\Pi, 1^t)$ denote the output of $M(w)$ when emulated on U for t steps. Note that (by assumption that U only has polynomial overhead) $U(\Pi, 1^t)$ can be computed in time $\mathsf{poly}(|\Pi|, t)$. We turn to defining Levin's notion of Kolmogorov complexity [Lev73]:

$$Kt(x) = \min_{\Pi \in \{0,1\}^*, t \in \mathbb{N}} \{|\Pi| + \lceil \log t \rceil : U(\Pi, 1^t) = x\}.$$

Its decisional variant, the Minimum Kt Complexity Problem MKtP, is defined as follows:

- Input: A string $x \in \{0,1\}^n$ and a size parameter $k \in \{0,1\}^{\lceil \log n \rceil}$.
- Decide: Does (x, k) satisfy $Kt(x) \leq k$?

As is well known, we can always produce a string by hardwiring the string in (the tape of) a machine that does nothing and just halts, which yields the following central fact about (Levin)-Kolmogorov complexity.

Fact 21 ([Sip96]) *There exists a constant c such that for every $x \in \{0,1\}^*$ it holds that $Kt(x) \leq |x| + c$.*

2.3 Average-case Complexity

We will consider average-case complexity of languages L with respect to the *uniform* distribution of instances. Let $\mathsf{Heur_{neg}BPP}$ denote the class of languages that can be decided by PPT heuristics that only make mistakes on a inverse polynomial fraction of instances. More formally:

Definition 3 ($\mathsf{Heur_{neg}BPP}$). *For a decision problem $L \subset \{0,1\}^*$, we say that $L \in \mathsf{Heur_{neg}BPP}$ if for all polynomial $p(\cdot)$, there exists a probabilistic polynomial-time heuristic $\mathcal{H}$, such that for all sufficiently large n,*

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = L(x)] \geq 1 - \frac{1}{p(n)}.$$

We will refer to languages in $\mathsf{Heur_{neg}BPP}$ as languages that admit *2-sided error* heuristics. We will also consider a more restrictive type of *errorless* heuristics $\mathcal{H}$: for every instance x, with probability 0.9 (over the randomness of only $\mathcal{H}$), $\mathcal{H}(x)$ either outputs $L(x)$ or $\perp$ (for 'I don't know'). More formally,

Definition 4 ($\mathsf{Avg_{neg}BPP}$). *For a decision problem $L \subset \{0,1\}^*$, we say that $L \in \mathsf{Avg_{neg}BPP}$ if for all polynomial $p(\cdot)$, there exists a probabilistic polynomial-time heuristic $\mathcal{H}$, such that for all sufficiently large n, for every $x \in \{0,1\}^n$,*[8]

$$\Pr[\mathcal{H}(x) \in \{L(x), \perp\}] \geq 0.9,$$

and

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = \perp] \leq \frac{1}{p(n)}.$$

We will refer to languages in $\mathsf{Avg_{neg}BPP}$ as languages that admit *errorless* heuristics. As explained in the introduction, to better understand the class $\mathsf{Avg_{neg}BPP}$, it may be useful to compare it to the class $\mathsf{Avg_{neg}P}$ (languages solvable by *deterministic* errorless heuristics): $L \in \mathsf{Avg_{neg}P}$ if for every polynomial $p(\cdot)$, there exists some deterministic polynomial-time heuristic $\mathcal{H}$ such that (a) for every input x, $\mathcal{H}(x)$ outputs either $L(x)$ or $\perp$, and (b) the probability over uniform n-bit inputs x that $\mathcal{H}$ outputs $\perp$ is bounded by $\frac{1}{p(n)}$. In other words, the *only* way an errorless heuristic may make a "mistake" is by saying $\perp$ ("I don't know"), whereas for a 2-sided error heuristic we do not know when mistakes happen. $\mathsf{Avg_{neg}BPP}$ is simply the natural "BPP-analog" of $\mathsf{Avg_{neg}P}$ where the heuristic is allowed to be randomized.

2.4 Computational Indistinguishability

We recall the definition of (computational) indistinguishability [GM84] along with its infinitely-often variant.

Definition 5. *Two ensembles $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ are said to be* $\varepsilon(\cdot)$-indistinguishable*, if for every* PPT *machine D (the "distinguisher") whose running time is polynomial in the length of its first input, there exists some $n_0 \in \mathbb{N}$ so that for every $n \geq n_0$:*

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

We say that $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ are infintely-often $\varepsilon(\cdot)$-indistinguishable (io-ε-indistinguishable) *if the above condition holds for infinitely many $n \in \mathbb{N}$ (as opposed to all sufficiently large ones).*

[8] We remark that the constant 0.9 can be made arbitrarily small—any constants bounded away from $\frac{2}{3}$ works as we can amplify it using a standard Chernoff-type argument.

2.5 Pseudorandom Generators

We recall the standard definition of pseuodrandom generators (PRGs) and its infinitely-often variant.

Definition 6. *Let* $g : \{0,1\}^n \to \{0,1\}^{m(n)}$ *be a polynomial-time computable function. g is said to be a* $\varepsilon(\cdot)$-pseudorandom generator (ε-PRG) *if for any* PPT *algorithm* $\mathcal{A}$ *(whose running time is polynomial in the length of its first input), for all sufficiently large* n*,*

$$|\Pr[x \leftarrow \{0,1\}^n : \mathcal{A}(1^n, g(x)) = 1] - \Pr[y \leftarrow \{0,1\}^{m(n)} : \mathcal{A}(1^n, y) = 1]| < \varepsilon(n).$$

g is said to be an infinitely-often $\varepsilon(\cdot)$-pseudorandom generator (io-ε-PRG) *if the above condition holds for infinitely many* $n \in \mathbb{N}$ *(as opposed to all).*

Although the standard cryptographic definition of a PRG g requires that g runs in polynomial time, when used for the other purposes (e.g., for derandomizing BPP), we allow the PRG g to have an exponential running time [TV02]. We refer to such PRGs (resp ioPRGs) as *inefficient* PRGs (resp *inefficient* ioPRGs).

2.6 Conditionally Entropy-preserving PRGs

Liu and Pass [LP20] introduced variant of a PRG referred to as an *entropy-preserving* pseudorandom generator (EP-PRG). Roughly speaking, an EP-PRG is a pseudorandom generator that expands n-bits to $n + O(\log n)$ bits, having the property that the output of the PRG is not only pseudorandom, but also preserves the entropy of the input (i.e., the seed): The Shannon-entropy of the output is $n - O(\log n)$. [LP20] did not manage to construct an EP-PRG from OWFs, but rather constructed a relaxed form of an EP-PRG, called a *conditionally-secure* entropy-preserving PRG (condEP-PRG), which relaxes both the pseudorandomness, and entropy-preserving properties of the PRG, to hold only conditioned on some event E. We will here consider also an infinitely-often variant:

Definition 7. *An efficiently computable function* $G : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$ *is a* $\mu(\cdot)$-conditionally secure entropy-preserving pseudorandom generator (μ-condEP-PRG) *if there exist 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):** $\{G(\mathcal{U}_n \mid E_n)\}_{n\in\mathbb{N}}$ *and* $\{\mathcal{U}_{n+\gamma \log n}\}_{n\in\mathbb{N}}$ *are* $\mu(n)$*-indistinguishable;*
- **(entropy-preserving):** *For all sufficiently large* $n \in \mathbb{N}$, $H(G(\mathcal{U}_n \mid E_n)) \geq n - \alpha \log n$.

G *is referred to as an* $\mu(\cdot)$-conditionally secure entropy-preserving infinitely-often pseudorandom generator (μ-condEP-ioPRG) *if it satisfies the above definition except that we replace* $\mu(n)$*-indistinguishability with io-*$\mu(n)$*-indistinguishability.*

We say that G has *rate-1 efficiency* if its running time on inputs of length n is bounded by $n + O(n^\varepsilon)$ for some constant $\varepsilon < 1$. We recall that the existence of rate-1 efficient condEP-PRGs can be based on the existence of OWFs, and that the same theorem holds in the infinitely-often setting.

Theorem 7 ([LP20]). *Assume that OWFs (resp. ioOWFs) exist. Then, for every $\gamma > 1$, there exists a rate-1 efficient μ-condEP-PRG (resp. μ-condEP-ioPRG) $G_\gamma : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$, where $\mu = \frac{1}{n^2}$.*

3 2-Sided Error Average-case Hardness of MKtP and OWFs

In this section, we prove our main characterization of OWFs through 2-sided error average-case hardness of MKtP.

Theorem 8. $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ *iff infinitely-often OWFs exist.*

We remark that, in Appendix of the full paper, we also characterize "standard" (as opposed to infinitely-often) OWFs through (almost-everywhere) mild average-case hardness of MKtP.

Theorem 8 follows directly from Theorem 9 (which is proven in Section 3.1) and Theorem 10 (which is proven in Section 3.2).

3.1 OWFs from Two-sided Error Avg-case Hardness of MKtP

In this section, we show that if weak ioOWFs do not exists, then we can compute the Kt-complexity of random strings with high probability (and thus MKtP is in $\mathsf{Heur_{neg}BPP}$). On a high-level, we will be using the same proof approach as in [LP20]. One immediate obstacle to relying on the proof in [LP20] is that it relies on the fact that the program Π (which we refer to as the "witness") that certifies the time-bounded Kolmogorov complexity K^t of a string x, has some a-priori *polynomial* running time, namely $t(\cdot)$; this polynomial bound gets translated into the running time of the constructed OWF. Unfortunately, this fact no longer holds when it comes to Kt-complexity: We say that the program Π is a *Kt-witness* for the string x if Π generates x within t steps while minimizing $|\Pi| + \log t$ among all other programs (i.e., Π is a witness for the Kt-complexity of x). Note that given a Kt-witness of a string x, there is no a-priori polynomial time-bound on the running time of Π, since only the *logarithm* of the running time gets included in the complexity measure. For instance, it could be that the Kt-witness is a program Π of length $n/10$ that requires running time $2^{n/10}$, for a total Kt-complexity of $n/5$. Nevertheless, the crucial observation we make is that *for most strings* x, the running-time of the Kt-witness actually is small: For every $0 < \varepsilon < 1$, except for an ε fraction of n-bit strings x, x has a Kt-witness Π that runs in time $O(\frac{1}{\varepsilon})$.

More formally:

Fact 31 *For all $n \in \mathbb{N}$, $0 < \varepsilon < 1$, there exists $1 - \varepsilon$ fraction of strings $x \in \{0,1\}^n$ such that there exist a Turing machine Π_x and a running time parameter t_x satisfying $U(\Pi_x, 1^{t_x}) = x$, $|\Pi_x| + \lceil \log t_x \rceil = Kt(x)$, and $t_x \leq 2^c/\varepsilon$ (where c is as in Fact 21).*

Proof: Consider some $n \in \mathbb{N}$, $0 < \varepsilon < 1$, and some set $S \subset \{0,1\}^n$ such that $|S| > \varepsilon 2^n$. For any string $x \in \{0,1\}^n$, let (Π_x, t_x) be a pair of strings such that $U(\Pi_x, 1^{t_x}) = x$ and $|\Pi_x| + \lceil \log t_x \rceil = Kt(x)$; that is, (Π_x, t_x) is the optimal compression for x. Note that for any $x \in \{0,1\}^n$, such (Π_x, t_x) always exists due to Fact 21.[9] Let c be the constant from Fact 21.

We assume for contradiction that for any $x \in S$, $t_x > 2^c/\varepsilon$. Note that by Fact 21, it holds that $Kt(x) \leq |x| + c$. Thus, $|\Pi_x| = Kt(x) - \lceil \log t_x \rceil \leq n + c - \lceil \log 2^c/\varepsilon \rceil \leq n - \log 1/\varepsilon$. Consider the set $Z = \{\Pi_x : x \in S\}$ of all (descriptions of) Turing machines Π_x. Since $|\Pi_x| \leq n - \log 1/\varepsilon$, it follows that $|Z| \leq 2^{n - \log 1/\varepsilon} = \varepsilon 2^n$. However, for each machine Π in Z, it could produce only a single string in S. So $|Z| \geq |S| > \varepsilon 2^n$, which is a contradiction. ■

We now show how to adapt the proof in [LP20] by relying on the above fact.

Theorem 9. *If $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$, then there exists a weak ioOWF (and thus also an ioOWF).*

Proof: We start with the assumption that $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$; that is, there exists a polynomial $p(\cdot)$ such that for all PPT heuristics $\mathcal{H}'$ and infinitely many n,

$$\Pr[x \leftarrow \{0,1\}^n, k \leftarrow \{0,1\}^{\lceil \log n \rceil} : \mathcal{H}'(x,k) = \mathsf{MKtP}(x,k)] < 1 - \frac{1}{p(n)}.$$

Let c be the constant from Fact 21. Consider the function $f : \{0,1\}^{n+c+\lceil \log(n+c) \rceil} \rightarrow \{0,1\}^*$, which given an input $\ell || \Pi'$ where $|\ell| = \lceil \log(n+c) \rceil$ and $|\Pi'| = n + c$, outputs $\ell + \lceil \log t \rceil || U(\Pi, 1^t)$ where Π is the ℓ-bit prefix of Π', t is the (smallest) integer $\leq 2^{c+2}p(n)$ such that Π (when interpreted as a Turing machine) halts in step t. (If Π does not halt in $2^{c+2}p(n)$ steps, f picks $t = 2^{c+2}p(n)$.) That is,

$$f(\ell || \Pi') = \ell + \lceil \log t \rceil || U(\Pi, 1^t).$$

Observe that f is only defined over some input 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 $m = n + c + \lceil \log(n+c) \rceil$ for some n and outputs $f(x)$.

We now show that f is a $\frac{1}{q(\cdot)}$-weak ioOWF where $q(n) = 2^{2c+4}np(n)^2$, which concludes the proof of the theorem. Assume for contradiction that f is not a

[9] We note that the choice of (Π_x, t_x) for some x is not unique. Our argument holds if any such (Π_x, t_x) is chosen.

$\frac{1}{q(\cdot)}$-weak ioOWF; 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 all sufficiently large input lengths $m = n + c + \lceil \log(n+c) \rceil$. We first claim that we can use $\mathcal{A}$ to construct a PPT heuristic $\mathcal{H}^*$ such that

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}^*(x) = Kt(x)] \geq 1 - \frac{1}{p(n)}.$$

If this is true, consider the heuristic $\mathcal{H}$ which given a string $x \in \{0,1\}^n$ and a size parameter $k \in \{0,1\}^{\lceil \log n \rceil}$, outputs 1 if $\mathcal{H}^*(x) \leq k$, and outputs 0 otherwise. Note that if $\mathcal{H}^*$ succeeds on some string x, $\mathcal{H}$ will also succeed. Thus,

$$\Pr[x \leftarrow \{0,1\}^n, k \leftarrow \{0,1\}^{\lceil \log n \rceil} : \mathcal{H}(x,k) = \mathsf{MKtP}(x,k)] \geq 1 - \frac{1}{p(n)},$$

which is a contradiction.

It remains to construct the heuristic $\mathcal{H}^*$ that computes $Kt(x)$ with high probability over random inputs $x \in \{0,1\}^n$, using $\mathcal{A}$. 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 $\frac{2p(n)}{q(n)}$. Consider some such "good" randomness r for which $\mathcal{A}_r$ succeeds to invert f with probability $1 - \frac{2p(n)}{q(n)}$.

On input $x \in \{0,1\}^n$, our heuristic $\mathcal{H}_r^*$ runs $\mathcal{A}_r(i||x)$ for all $i \in [n+c]$ where i is represented as a $\lceil \log(n+c) \rceil$-bit string, and outputs the smallest i where the inversion on $(i||x)$ succeeds. Let $\varepsilon = \frac{1}{4p(n)}$, and S be the set of strings $x \in \{0,1\}^n$ for which $\mathcal{H}_r^*(x)$ fails to compute $Kt(x)$ and x satisfies the requirements in Fact 31. Note that the probability that a random $x \in \{0,1\}^n$ does not satisfy the requirements in Fact 31 is at most ε. Thus, $\mathcal{H}_r^*$ fails with probability at most (by a union bound)

$$\mathsf{fail}_r \leq \varepsilon + \frac{|S|}{2^n}.$$

Consider any string $x \in S$ and let $w = Kt(x)$ be its Kt-complexity. Note that x satisfies the requirements in Fact 31; that is, there exist a Turing machine Π_x and a running time parameter t_x such that $U(\Pi_x, 1^{t_x}) = x$, $|\Pi_x| + \lceil \log t_x \rceil = Kt(x)$, and $t_x \leq 2^c/\varepsilon = 2^{c+2}p(n)$. By Fact 21, we have that $|\Pi_x| \leq w \leq n + c$. Thus, for all strings $(\ell||\Pi') \in \{0,1\}^{n+c+\lceil \log(n+c) \rceil}$ such that $\ell = |\Pi_x|$, $[\Pi']_{|\ell|} = \Pi_x$, it holds that $f(\ell||\Pi') = (w||x)$. Since $\mathcal{H}_r^*(x)$ fails to compute $Kt(x)$, $\mathcal{A}_r$ must fail to invert $(w||x)$. But, since $|\Pi_x| \leq n + c$, the output $(w||x)$ is sampled with probability at least

$$\frac{1}{n+c} \cdot \frac{1}{2^{|\Pi_x|}} \geq \frac{1}{n+c} \frac{1}{2^{n+c}} \geq \frac{1}{n 2^{2c+1}} \cdot \frac{1}{2^n}$$

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

$$|S| \cdot \frac{1}{n 2^{2c+1}} \cdot \frac{1}{2^n} = \frac{1}{n 2^{2c+1}} \cdot \frac{|S|}{2^n} \geq \frac{\mathsf{fail}_r - \varepsilon}{n 2^{2c+1}}$$

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

$$\mathsf{fail}_r \leq \frac{2^{2c+2} np(n)}{q(n)} + \varepsilon$$

Finally, by a union bound, we have that $\mathcal{H}^*$ (using a uniform random tape r) fails in computing Kt with probability at most

$$\frac{1}{2p(n)} + \frac{2^{2c+2} np(n)}{q(n)} + \varepsilon = \frac{1}{2p(n)} + \frac{2^{2c+2} np(n)}{2^{2c+4} np(n)^2} + \frac{1}{4p(n)} = \frac{1}{p(n)}.$$

Thus, $\mathcal{H}^*$ computes Kt with probability $1 - \frac{1}{p(n)}$ for all sufficiently large $n \in \mathbb{N}$, which is a contradiction. ■

3.2 Two-sided Error Avg-case Hardness of MKtP from ioOWFs

In this section, we will prove the following theorem:

Theorem 10. *If ioOWFs exist, then* $\mathsf{MKtP} \notin \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$.

Proof: The theorem follows immediately from Theorem 7 and Theorem 11 that will be stated and proved below. ■

Recall that Theorem 7 shows that ioOWFs imply the existence of rate-1 efficient condEP-ioPRGs. Theorem 11 below will show that the existence of rate-1 efficient condEP-ioPRGs implies that $\mathsf{MKtP} \notin \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$. We remark that the proof of this theorem closely follows the proof in [LP20] and relying with only relatively minor modifications to observe that the properties used of the time-bounded Kolmogorov complexity function actually also hold for K^t—namely that random strings have "high" K^t-complexity, whereas outputs of a PRG have "low" K^t-complexity.[10]

Theorem 11. *Assume that for some $\gamma \geq 4$, there exists a rate-1 efficient μ-condEP-ioPRG $G : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$ where $\mu(n) = 1/n^2$. Then,* $\mathsf{MKtP} \notin \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$.

Proof: Let $G : \{0,1\}^n \to \{0,1\}^{m(n)}$ where $m(n) = n + \gamma \log n$ be a rate-1 efficient $\frac{1}{n^2}$-condEP-ioPRG with entropy loss constant α. Let $p(n) = 2n^{2(\alpha+\gamma+2)}$. We assume for contradiction that $\mathsf{MKtP} \in \mathsf{Heur}_{\mathsf{neg}}\mathsf{BPP}$; that is, there exists some PPT $\mathcal{H}$ that decides MKtP with probability at least $1 - \frac{1}{p(m')}$ where $m'(m) = m + \lceil \log m \rceil$ (on input length m') for all sufficiently large n, $m(n)$, and $m'(m)$. Recall that G is associated with a sequence of events $\{E_n\}_{n \in \mathbb{N}}$.

[10] There are also some other minor differences due to the fact that the proof in [LP20] considered the hardness of *computing* (or approximating) K^t, whereas we here consider a *decisional* problem with a random threshold k, but the proof in [LP20] extends in a relatively straightforward way to deal also with decisional problems with a random threshold k.

We show that $\mathcal{H}$ can be used to break the condEP-ioPRG G. Towards this, recall that a random string has high Kt-complexity with high probability: for $m = m(n)$, we have,

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

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 G, must have "low" Kt complexity: For every sufficiently large $n, m = m(n)$, we have that,

$$\Pr_{z \in \{0,1\}^n}[Kt(G(z)) > m - \frac{\gamma}{2} \log n] = 0, \quad (2)$$

since $G(z)$ can be represented by combining a seed z of length n with the code of G (of constant length), and the running time of $G(z)$ is bounded by $1.1n$ for all sufficiently large n (since G is rate-1 efficient), so $Kt(G(z)) = n + O(1) + \lceil \log(1.1n) \rceil = (m - \gamma \log n) + O(1) + \lceil \log(1.1n) \rceil \leq m - \gamma/2 \log n$ for sufficiently large n (since recall that $\gamma \geq 4$).

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 $k = m - \frac{\gamma}{2} \log n$ and outputs 1 if $\mathcal{H}(x, k)$ outputs 1 and 0 otherwise. Consider some sufficiently large n, $m(n)$, and $m'(n)$. 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 0 with probability at least $1 - \frac{2}{n^{\gamma/2}}$.*

Proof: Note that $\mathcal{A}(1^n, x)$ will output 0 if x is a string with Kt-complexity larger than $m - \gamma/2 \log n$ and $\mathcal{H}$ succeeds on input (x, k). Thus,

$$\begin{aligned}
&\Pr[\mathcal{A}(1^n, x) = 0] \\
&\geq \Pr[Kt(x) > m - \gamma/2 \log n \wedge \mathcal{H} \text{ succeeds on } (x, k)] \\
&\geq 1 - \Pr[Kt(x) \leq m - \gamma/2 \log n] - \Pr[\mathcal{H} \text{ fails on } (x, k)] \\
&\geq 1 - \frac{1}{n^{\gamma/2}} - \frac{1}{p(m')} \\
&\geq 1 - \frac{2}{n^{\gamma/2}}.
\end{aligned}$$

where the probability is over a random $x \leftarrow \mathcal{U}_m$, $k \leftarrow \lceil \log m \rceil$ and the randomness of $\mathcal{A}$ and $\mathcal{H}$. ■

claim 2 *$\mathcal{A}(1^n, G(\mathcal{U}_n \mid E_n))$ outputs 0 with probability at most $1 - \frac{1}{n} + \frac{2}{n^2}$*

Proof: Recall that by assumption, $\mathcal{H}(x, k)$ fails to decide whether $(x, k) \in \mathsf{MKtP}$ for a random $x \in \{0,1\}^m, k \in \{0,1\}^{\lceil \log m \rceil}$ with probability at most $\frac{1}{p(m')}$ (where $m' = m + \lceil \log m \rceil$). 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 decide MKtP with probability at most $\frac{n^2}{p(m')}$. Fix some "good" randomness r such that $\mathcal{H}_r$ decides MKtP 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 $y \in G(\mathcal{U}_n \mid E_n)$. 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 $S=\{y \in G(\mathcal{U}_n \mid E_n) : \mathcal{A}_r(1^n, y)=0 \wedge y \text{ is good}\}$, and let $S'=\{y \in G(\mathcal{U}_n \mid E_n) : \mathcal{A}_r(1^n, y)=0 \wedge y \text{ is bad}\}$. Since

$$\Pr[\mathcal{A}_r(1^n, G(\mathcal{U}_n \mid E_n))=0]=\Pr[G(\mathcal{U}_n \mid E_n) \in S]+\Pr[G(\mathcal{U}_n \mid E_n) \in S'],$$

and $\Pr[G(\mathcal{U}_n \mid E_n) \in S']$ is at most the probability that $G(\mathcal{U}_n \mid E_n)$ 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 0, then by Equation 2, $Kt(y) \leq m-\gamma/2 \log n=k$, so $\mathcal{H}_r$ fails to decide MKtP on input (y, k).

Thus, the probability that $\mathcal{H}_r$ fails (to decide MKtP) on a random input $(y, k) \in\{0,1\}^{m'}$ is at least

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

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

We conclude that for every good randomness r, $\mathcal{A}_r$ outputs 0 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 $\gamma \geq 4$, that $\mathcal{A}$ distinguishes $\mathcal{U}_m$ and $G(\mathcal{U}_n \mid E_n)$ with probability of at least

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

for all sufficiently large $n \in \mathbb{N}$. ■

4 Errorless Avg-case Hardness of MKtP and EXP ≠ BPP

In this section, we will prove the following theorem:

Theorem 12. $\mathsf{EXP} \neq \mathsf{BPP}$ *if and only if* $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$.

Roughly speaking, the above theorem is proved in two steps:

- We first observe that, assuming $\mathsf{EXP} \neq \mathsf{BPP}$, there exists an (inefficient, infinitely-often) pseudorandom generator [IW98] that maps a n^ε-bit seed to a n-bit string in time $O(2^{n^\gamma})$ (for some $0 < \varepsilon, \gamma < 1$).
- We will next show that an errorless heuristic for MKtP can be used to break such PRGs (since the Kt-complexity of the output of the PRG is at most $n^\varepsilon + n^\gamma + O(1) \leq n - 1$), which is a contradiction and concludes the proof.

Recall that Impagliazzo and Wigderson [IW98] showed that BPP can be derandomized (on average) in subexponential time by assuming $\mathsf{EXP} \neq \mathsf{BPP}$. The central technical contribution in their work can be stated as proving the existence of an inefficient PRG assuming $\mathsf{EXP} \neq BPP$:

Theorem 13 (implicit in [IW98], explicitly stated in e.g., [TV02, Theorem 3.9]). *Assume that* $\mathsf{EXP} \neq \mathsf{BPP}$*. Then, for all* $\varepsilon > 0$*, there exists an inefficient io-*$\frac{1}{10}$*-PRG* $G : \{0,1\}^{n^\varepsilon} \to \{0,1\}^n$ *that runs in time* $2^{O(n^\varepsilon)}$.

We note that the proof in [IW98], is non black-box. In particular, it does not show how to solve EXP in probabilistic polynomial-time having black-box access to an attacker that breaks the PRG.

It remains to show that if there exists an (inefficient) ioPRG $G : \{0,1\}^{n^\varepsilon} \to \{0,1\}^n$ with running time $O(2^{n^\gamma})$ (for some $0 < \varepsilon, \gamma < 1$), then $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$. We recall that a string's Kt-complexity is the minimal sum of (1) the description length of a Turing machine that prints the string and (2) the logarithm of its running time. Note that the output of G could be printed by a machine with the code of G (of constant length) and the seed (of length n^ε) hard-wired in it within $O(2^{n^\gamma})$ time. Thus, strings output by G have Kt-complexity less than or equal to $O(1) + n^\varepsilon + n^\gamma \leq n - 1$. On the other hand, random strings have high Kt-complexity (e.g., $> n - 1$) with high probability (e.g., $\geq \frac{1}{2}$). It follows that an errorless heuristic for MKtP can be used to break G. Let us highlight why it is important that we have an *errorless* heuristic (as opposed to a 2-sided error heuristic): while a 2-sided error heuristic would still work well on random strings, we do not have any guarantees on its success probability given pseudorandom strings (as they are sparse); an errorless heuristics, however, will either correctly decide those strings, or output $\bot$ (in which case, we can also guess that the string is pseudorandom).

We proceed to a formal statement of the theorem, and its proof.

Theorem 14. *Assume that there exist constants* $0 < \varepsilon, \gamma < 1$ *and an inefficient io-*$\frac{1}{10}$*-PRG* $G : \{0,1\}^{n^\varepsilon} \to \{0,1\}^n$ *with running time* $O(2^{n^\gamma})$*. Then,* $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$*.*

Proof: We assume for contradiction that $\mathsf{MKtP} \in \mathsf{Avg_{neg}BPP}$, which in turn implies that there exists an errorless PPT heuristic $\mathcal{H}$ such that for all sufficiently large n, every $x \in \{0,1\}^n$ and $k \in \{0,1\}^{\lceil \log n \rceil}$,

$$\Pr[\mathcal{H}(x,k) \in \{\mathsf{MKtP}(x,k), \bot\}] \geq 0.9, \tag{3}$$

and

$$\Pr[x \leftarrow \{0,1\}^n, k \leftarrow \{0,1\}^{\lceil \log n \rceil} : \mathcal{H}(x,k) = \bot] \leq \frac{1}{2n^2}.$$

Fix some sufficiently large n, and let $k = n - 1$. It follows by an averaging argument that

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x, n-1) = \bot] \leq \frac{1}{2n^2} \cdot 2^{\lceil \log n \rceil} \leq \frac{1}{n}. \tag{4}$$

We next show that we can use $\mathcal{H}$ to break the PRG G. On input $x \in \{0,1\}^n$, our distinguisher $\mathcal{A}(1^{n^\varepsilon}, x)$ outputs 1 if $\mathcal{H}(x, n-1) = 1$ or $\mathcal{H}(x, n-1) = \bot$. $\mathcal{A}$ outputs 0 if and only if $\mathcal{H}(x, n-1) = 0$. The following two claims conclude that $\mathcal{A}$ distinguishes $\mathcal{U}_n$ and $G(\mathcal{U}_{n^\varepsilon})$ with probability at least 0.2.

Claim 1. $\mathcal{A}(1^{n^\varepsilon}, \mathcal{U}_n)$ will output 0 with probability at least $0.4 - \frac{1}{n}$.

Proof: Note that the probability that a random string $x \in \{0,1\}^n$ is of Kt-complexity at most $n-1$ is at most $\frac{2^{n-1}}{2^n} = \frac{1}{2}$ (since the total number of machines with description length $\leq n-1$ is 2^{n-1}). And the probability that $\mathcal{H}(x, n-1)$ outputs $\bot$ is at most $\frac{1}{n}$ (over random $x \in \{0,1\}^n$) by Equation 4. In addition, the probability that $\mathcal{H}(x, n-1)$ fails to output either $\mathsf{MKtP}(x, n-1)$ or $\bot$ is at most 0.1 by Equation 3. Thus, by a union bound,

$$\begin{aligned}
&\Pr[\mathcal{A}(1^{n^\varepsilon}, \mathcal{U}_n) = 0] \\
\geq& 1 - \Pr[Kt(\mathcal{U}_n) \leq n-1] - \Pr[\mathcal{H}(\mathcal{U}_n, n-1) = \bot] - \Pr[\mathcal{H}(\mathcal{U}_n, n-1) \text{ fails}] \\
\geq& 1 - \frac{1}{2} - \frac{1}{n} - 0.1 \\
=& 0.4 - \frac{1}{n}.
\end{aligned}$$

■

Claim 2. $\mathcal{A}(1^{n^\varepsilon}, G(\mathcal{U}_{n^\varepsilon}))$ will output 0 with probability at most 0.1.

Proof: We first show that for all $z \in \{0,1\}^{n^\varepsilon}$, $Kt(G(z)) \leq n^\varepsilon + n^\gamma + O(1) \leq n - 1 = s$. Note that the string $G(z)$ could be produced by a machine with the code of G (of length $O(1)$) and the seed z (of length n^ε) in time $O(2^{n^\gamma})$ (which adds $\log O(2^{n^\gamma}) = n^\gamma + O(1)$ to its Kt-complexity). In addition, recall that $\mathcal{H}$ is a probabilistic errorless heuristics. Thus, $\mathcal{H}(G(z), n-1)$ will output 0 with probability at most 0.1 (by Equation 3), and the claim follows. ■

This conclude the proof of Theorem 14. ■

We are now ready to conclude the proof of Theorem 12.
Proof: [of Theorem 12] We show each direction separately:

- To show that $\mathsf{EXP} \neq \mathsf{BPP} \implies \mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$, assume that $\mathsf{EXP} \neq \mathsf{BPP}$ and let $\varepsilon = \frac{1}{3}$, and $\gamma = \frac{1}{2}$. By Theorem 13, there exists an io-$\frac{1}{10}$-PRG $G : \{0,1\}^{n^\varepsilon} \to \{0,1\}^n$ with running time $2^{O(n^\varepsilon)} \leq O(2^{n^\gamma})$. We conclude by Theorem 14 that $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$.
- To show that $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{EXP} \neq \mathsf{BPP}$, assume that $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$; this trivially implies that $\mathsf{MKtP} \notin \mathsf{BPP}$. We observe that $\mathsf{MKtP} \in \mathsf{EXP}$ as by Fact 21, $Kt(x) \leq |x| + O(1)$ and thus the running-time for a Kt-witness, Π, for x is bounded by $2^{|x|+O(1)}$. Thus, $\mathsf{EXP} \not\subseteq \mathsf{BPP}$, which in particular means that $\mathsf{EXP} \neq \mathsf{BPP}$.

■

5 On the implication $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$

Recall that in Theorem 12, we showed that if one assumes an (extremely) weak lowerbound (namely, $\mathsf{EXP} \neq \mathsf{BPP}$), then the problem MKtP is hard on average for errorless heuristics. Furthermore, in Theorem 9, we showed that if the problem MKtP is hard-on-average for 2-sided error heuristics that only make a small number of mistakes, then (infinitely-often) one-way functions exist. Combining the two theorems together, we have that the implication $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \implies \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ fully characterizes when we can base the existence of (infinitely-often) one-way functions on $\mathsf{EXP} \neq \mathsf{BPP}$. Formally,

Theorem 15. $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \Rightarrow \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ *holds iff* $\mathsf{EXP} \neq \mathsf{BPP} \Rightarrow$ *the existence of ioOWFs.*

Proof: The proof immediately follows from Theorem 12 and Theorem 9. ■

Perhaps surprisingly, we observe that the implication itself (without any assumptions) implies that $\mathsf{NP} \neq \mathsf{P}$. The pessimistic way to interpret this is that closing the gap between 2-sided error, and errorless, heuristics will be very hard (as it requires proving that $\mathsf{NP} \neq \mathsf{P}$). The optimistic way to interpret it, however, is as a new and algorithmic approach towards proving that $\mathsf{NP} \neq \mathsf{P}$: To demonstrate that $\mathsf{NP} \neq \mathsf{P}$, it suffices to demonstrate that MKtP can be solved by an errorless heuristic, given access to a two-sided error heuristic for the same problem. (As we shall point out shortly, this approach also does not "overshoot" the NP vs P problem by too much: any proof of the existence of infinitely often one-way functions, needs to show this implication.)

Theorem 16. *If it holds that* $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \Rightarrow \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$*, then* $\mathsf{NP} \neq \mathsf{P}$*.*

Proof: Assume for contradiction that $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \Rightarrow \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ holds, yet $\mathsf{NP} = \mathsf{P}$. Recall that $\mathsf{BPP} \subseteq \mathsf{NP^{NP}}$ [Sip83,Lau83], so it follows that $\mathsf{P} = \mathsf{BPP}$, and thus by the time-hierarchy Theorem [HS65], $\mathsf{EXP} \neq \mathsf{BPP}$. Then, by Theorem 12, $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP}$. It follows from our assumption that $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \Rightarrow \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ and from Theorem 15 that ioOWFs exist, which contradicts the assumption that $\mathsf{NP} = \mathsf{P}$. ■

We remark that the above theorem could be strengthened to show even that NP is average-case hard (w.r.t. deterministic errorless heuristics), since Buhrman, Fortnow, and Pavan [BFP03] have showed that unless this is the case, $\mathsf{P} = \mathsf{BPP}$, which suffices to complete the rest of the proof.

Finally, we remark that the implication $\mathsf{MKtP} \notin \mathsf{Avg_{neg}BPP} \Rightarrow \mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$ must be true if infinitely-often one-way functions exist since by Theorem 10, the existence of ioOWFs implies $\mathsf{MKtP} \notin \mathsf{Heur_{neg}BPP}$, which in turn implies that the implication trivially holds.

6 Acknowledgments

We are very grateful to Salil Vadhan for helpful discussions about the PRG construction of [IW98]. The first author also wishes to thank Hanlin Ren for helpful discussions about Levin's notion of Kolmogorov Complexity.

Deferred Content Due to the page limit, we refer the read to the full paper for characterization of log-space computable OWFs and OWFs in uniform NC^0 through some notions of resource-bounded Kolmogorov complexity. We will also present characterizations of "standard" OWFs (instead of infinitely-often) in the appendix of the full paper.

References

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.

ACM+21. Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-way functions and a conditional variant of mktp. *manuscript*, 2021.

AGGM06. Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In *STOC '06*, pages 701–710, 2006.

AIK06. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in ncˆ0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996.

BB15. Andrej Bogdanov and Christina Brzuska. On basing size-verifiable one-way functions on NP-hardness. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2015.

BFP03. Harry Buhrman, Lance Fortnow, and Aduri Pavan. Some results on derandomization. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 212–222. Springer, 2003.

Blu82. Manuel Blum. Coin flipping by telephone - A protocol for solving impossible problems. In *COMPCON'82, Digest of Papers, Twenty-Fourth IEEE Computer Society International Conference, San Francisco, California, USA, February 22-25, 1982*, pages 133–137. IEEE Computer Society, 1982.

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.

Bra83. Gilles Brassard. Relativized cryptography. *IEEE Transactions on Information Theory*, 29(6):877–893, 1983.

BT03. Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for np problems. In *FOCS '03*, pages 308–317, 2003.

BT08. Andrej Bogdanov and Luca Trevisan. Average-case complexity. Manuscript, 2008. `http://arxiv.org/abs/cs.CC/0606037`.

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.

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

FS90. Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.

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

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.

GWXY10. S. Dov Gordon, Hoeteck Wee, David Xiao, and Arkady Yerukhimovich. On the round complexity of zero-knowledge proofs based on one-way permutations. In *LATINCRYPT*, pages 189–204, 2010.

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.

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.

HMX10. Iftach Haitner, Mohammad Mahmoody, and David Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In *IEEE Conference on Computational Complexity*, pages 76–87, 2010.

HS65. Juris Hartmanis and Richard E Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

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.

IW98. R Impagliazzo and A Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 734–743. IEEE, 1998.

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.

Lau83. Clemens Lautemann. BPP and the polynomial hierarchy. *Inf. Process. Lett.*, 17(4):215–217, 1983.

Lev73. Leonid A. Levin. Universal search problems (russian), translated into english by ba trakhtenbrot in [Tra84]. *Problems of Information Transmission*, 9(3):265–266, 1973.

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

Liv10. Noam Livne. On the construction of one-way functions from average case hardness. In *ICS*, pages 301–309. Citeseer, 2010.

LM91. Luc Longpré and Sarah Mocas. Symmetry of information and one-way functions. In Wen-Lian Hsu and Richard C. T. Lee, editors, *ISA '91 Algorithms, 2nd International Symposium on Algorithms, Taipei, Republic of China, December 16-18, 1991, Proceedings*, volume 557 of *Lecture Notes in Computer Science*, pages 308–315. Springer, 1991.

LP20. Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1243–1254. IEEE, 2020.

LP21. Yanyi Liu and Rafael Pass. On one-way functions from np-complete problems. *Electron. Colloquium Comput. Complex.*, 28:59, 2021.

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

NW94. Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

OW93. Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

PV20. Rafael Pass and Muthuramakrishnan Venkitasubramaniam. Is it easier to prove theorems that are guaranteed to be true? In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1255–1267. IEEE, 2020.

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

RS21. Hanlin Ren and Rahul Santhanam. Hardness of KT characterizes parallel cryptography. *Electron. Colloquium Comput. Complex.*, 28:57, 2021.

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.

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.

Sip96. Michael Sipser. Introduction to the theory of computation. *ACM Sigact News*, 27(1):27–29, 1996.

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.

TV02. Luca Trevisan and Salil Vadhan. Pseudorandomness and average-case complexity via uniform reductions. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 0129–0129. IEEE Computer Society, 2002.

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.

ZL70. A. K. Zvonkin and L. A. Levin. the Complexity of Finite Objects and the Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms. *Russian Mathematical Surveys*, 25(6):83–124, December 1970.