Electronic Colloquium on Computational Complexity, Report No. 55 (2021)

Cryptography from Sublinear-Time Average-Case Hardness of Time-Bounded Kolmogorov Complexity*

Yanyi Liu
Cornell University
yl2866@cornell.edu

Rafael Pass[†]
Cornell Tech
rafael@cs.cornell.edu

April 20, 2021

Abstract

Let $\mathsf{MK}^t\mathsf{P}[s]$ be the set of strings x such that $K^t(x) \leq s(|x|)$, where $K^t(x)$ denotes the t-bounded Kolmogorov complexity of the truthtable described by x. Our main theorem shows that for an appropriate notion of mild average-case hardness, for every $\varepsilon > 0$, polynomial $t(n) \geq (1+\varepsilon)n$, and every "nice" class $\mathcal{F}$ of super-polynomial functions, the following are equivalent:

- the existence of some function $T \in \mathcal{F}$ such that T-hard one-way functions (OWF) exists (with non-uniform security);
- the existence of some function $T \in \mathcal{F}$ such that $\mathsf{MK}^t\mathsf{P}[T^{-1}]$ is mildly average-case hard with respect to *sublinear-time* non-uniform algorithms (with running-time n^δ for some $0 < \delta < 1$).

For instance, existence of subexponentially-hard (resp. quasi-polynomially-hard) OWFs is *equivalent* to mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[\mathsf{poly}\log n]$ (resp. $\mathsf{MK}^t\mathsf{P}[2^{O(\sqrt{\log n})}]$) w.r.t. sublinear-time non-uniform algorithms.

We additionally note that if we want to deduce T-hard OWFs where security holds w.r.t. *uniform* T-time probabilistic attackers (i.e., uniformly-secure OWFs), it suffices to assume sublinear time hardness of $\mathsf{MK}^t\mathsf{P}$ w.r.t. uniform probabilistic sublinear-time attackers. We complement this result by proving lower bounds that come surprisingly close to what is required to *unconditionally* deduce the existence of (uniformly-secure) OWFs: $\mathsf{MK}^t\mathsf{P}[\mathsf{poly}\log n]$ is *worst-case* hard w.r.t. uniform probabilistic sublinear-time algorithms, and $\mathsf{MK}^t\mathsf{P}[n - \log n]$ is mildly average-case hard for all $O(t(n)/n^3)$-time deterministic algorithms.

[*] A preliminary version of this paper will appear in the proceedings of *STOC'21.* This is the full version.

[†] 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.

ISSN 1433-8092

1 Introduction

Given a truthtable $x \in \{0,1\}^n$ of a Boolean function, what is the size of the smallest "program" that computes x? This problem has fascinated researchers since the 1950 [Tra84, Yab59a, Yab59b], and various variants of it have been considered depending on how the notion of a program is formalized. For instance, when the notion of a program is taken to be circuits (e.g., with AND,OR,NOT gates), then it corresponds to the Minimum Circuit Size problem (MCSP) [KC00, Tra84], and when the notion of a program is taken to be a time-bounded Turing machine, then it corresponds to a Minimum Time-Bounded Kolmogorov complexity problem (MKTP) [Kol68, Ko86, Sip83, Har83, All01, ABK+06]. Our focus here is on the latter scenario. Given a string x describing a truthtable, let $K^t(x)$ denote the t-bounded Kolmogorov complexity of x—that is, the length of the shortest string Π such that for every $i \in [n]$, $U(\Pi, i) = x_i$ within time $t(|\Pi|)$, where U is a fixed Universal Turing machine.[1] Given a threshold, $s(\cdot)$, and a polynomial time-bound, $t(\cdot)$, let $\mathsf{MK}^t\mathsf{P}[s]$ denote the set of strings x such that $K^t(x) \leq s(|x|)$. $\mathsf{MK}^t\mathsf{P}[s]$ is clearly in NP, but it is unknown whether it is NP-complete.

Average-case complexity of this problem also has a long history, starting in the 1960s [Tra84]. In [LP20], we recently showed that when the threshold $s(\cdot)$ is "large" (more precisely, when $s(n) = n - c\log n$, for some constant c), then average-case hardness of this language (w.r.t., the uniform distribution of instances) is equivalent to the existence of one-way functions (OWF), providing the first natural problem that characterizes OWFs.[2] More precisely, it was shown that for any $\varepsilon > 0$, any polynomial $t(n) > (1+\varepsilon)n$, there exist some c, $s(n) = n - c\log n$, such that the existence of OWFs (secure against uniform PPT attackers) is equivalent to mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[s]$, where a language is said to be mildly hard-on-average (HoA) if there exists some polynomial $p(\cdot)$, such that no PPT algorithm can decide the language with probability $1 - \frac{1}{p(n)}$ over random n-bit instances for infinitely many n. The result of [LP20] also directly extends to the non-uniform setting: non-uniformly secure OWF (i.e., OWF secure against non-uniform polynomial-time algorithms) are equivalent to average-case hardness of $\mathsf{MK}^t\mathsf{P}[s]$ w.r.t. non-uniform polynomial-time attackers.

In this work, we consider what happens when the threshold $s(n)$ is smaller—for instance, when $s(n) = \mathsf{poly}\log n$ or $s(n) = 2^{O(\sqrt{\log n})}$. Does it make the problem harder or easier?

We here focus our attention mostly on the setting of non-uniform hardness (i.e., hardness against non-uniform algorithms); unless when we explictly refer to uniform hardness, we refer to hardness against T-time algorithms as hardness against T-time non-uniform algorithms (which is equivalent to hardness against T-time non-uniform probabilistic algorithms).

Cryptography from Sublinear-time Avg-case Hardness of $\mathsf{MK}^t\mathsf{P}$ Roughly speaking, our main theorem demonstrates that for an *appropriate* (more on this below) notion of mild average-case hardness:

- mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[\mathsf{poly}\log n]$ even with respect to just *sublinear time algorithms*—running in time n^δ for some $\delta < 1$—is equivalent to the existence of *subexponentially hard* OWFs.

[1] There are many ways to define time-bounded Kolmogorov complexity. We here consider the "local compression" version—which corresponds to the above truthtable compression problem—and where the running-time bound is a function of the length of the program. A different version of (time-bounded) Kolmogorov complexity instead considers the size of the shortest program that outputs the *whole* string x. This other notion refers to a "global compression" notion, but is less appealing from the point of view of truthtable compression, as the running-time of the program can never be smaller than the length of the truthtable x.

[2] Strictly speaking, [LP20] considered the "global compression" version of Kolmogorov complexity, but when the threshold is large, these notion are essentially equivalent, and the result from [LP20] directly applies also the "local compression" notion of Kolmogorov complexity considered here.

- mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[2^{O(\sqrt{\log n})}]$ with respect to just sublinear algorithms is equivalent to quasi-polynomially hard OWFs.

That is, two curious phenomena happen: (1) proving the existence of (subexponential) OWFs is *equivalent* to proving a *sublinear* average-case lower bound for a natural problem, and (2) the threshold, $s(n)$, for the $\mathsf{MK}^t\mathsf{P}[s]$ problem captures the *quantitative hardness* of OWFs. In fact, our result is more general and shows a smooth translation between the thresholds s and the hardness of the OWF: roughly speaking, we show that for "nice" classes $\mathcal{F}$ of super-polynomial time-bounds (that are closed under polynomial composition), the existence of $\mathcal{F}$-hard OWF is equivalent to mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[\mathcal{F}^{-1}]$ w.r.t. sublinear-time algorithms, where $\mathcal{F}^{-1}$ is the class of inverses to functions in $\mathcal{F}$.

A sliding-scale property for $\mathsf{MK}^t\mathsf{P}$ Along the way, we show that the $\mathsf{MK}^t\mathsf{P}[s]$ problem satisfies an intriguing "sliding-scale" property: mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[s]$ with respect to sublinear algorithms, is equivalent to mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[n/O(1)]$ with respect to algorithms with "large" running-time, where the actual running-time bound grows inversely with the threshold s. Thus, in a precise sense, when the threshold s is smaller, the problem becomes harder—in fact, it becomes equivalent to the large threshold case but with respect to stronger attackers. Intriguingly, our proof of this statement passes through the notion of a OWF (and relies on cryptographic techniques), and we see no direct way of showing it without doing so. We believe this highlights how our established connection between $\mathsf{MK}^t\mathsf{P}$ and OWF sheds new lights on time-bounded Kolmogorov complexity.

Unconditional Lower Bounds We remark that one direction of equivalence holds also w.r.t. uniform attackers: more specifically, to deduce T-hard OWFs where security holds w.r.t. *uniform* T-time probabilistic attackers (i.e., uniformly-secure OWFs), it suffices to assume sublinear time hardness of $\mathsf{MK}^t\mathsf{P}$ w.r.t. uniform sublinear-time attackers.

We complement this results by establishing lower bounds that come surprisingly close to what is required to *unconditionally* deduce the existence of subexponentially-hard uniformly-secure OWFs (and thus that $\mathsf{NP} \notin \mathsf{BPTIME}(2^{n^\alpha})$ for some $\alpha > 0$): (a) $\mathsf{MK}^t\mathsf{P}[\mathsf{poly}\log n]$ is *worst-case* hard w.r.t. sublinear-time uniform probabilistic algorithms—this falls short as we need average-case hardness, and (b) $\mathsf{MK}^t\mathsf{P}[n - \log n]$ is mildly average-case hard for all $t(n)/(n^3)$-time uniform deterministic[3] algorithms—this falls short as the threshold is too large.

1.1 Our Results in More Detail

We proceed to formalizing our result in more detail.

Two-sided Error Average-case Hardness for Sparse Languages Recall that we are interested in studying appropriate notions of average-case hardness of $\mathsf{MK}^t\mathsf{P}[s]$ when $s(\cdot)$ is "small". A problem with such languages is that they are *sparse*, so they are trivially *easy-on-average*: for instance, when $s(n) \leq n/2$, there are at most $2^{n/2}$ YES-instances in the languages, and thus the trivial heuristic that always outputs NO succeeds with overwhelming probability. The notion of an *errorless* μ-heuristic provides a meaningful way to capture a notion of average-case hardness for sparse languages: we restrict our attention to algorithms A that output $\perp$ only with probability μ, but when the algorithm does not output $\perp$, it is required to *always* provide a correct answer. Any

[3] Under standard derandomization assumptions, this lower bound also directly extends to probabilistic algorithms.

such errorless heuristic yields a *one-sided* error heuristics that never errs on YES, and errs only on a fraction $\mu(n)$ of NO instances (by simply outputting YES when A outputs $\perp$).

We here introduce a natural notion of *two-sided error* heuristics for sparse languages L: we say that A is a μ-heuristic* for L w.r.t. input length n, if A errs on at most a fraction $\mu(n^*)$ of either YES or NO instances of length n, where n^* is $\log |L \cap \{0,1\}^n|$; We say that L is mildly hard-on-average* (HoA*) if there exists some polynomial $p(\cdot)$ such that L does not have a $\frac{1}{p(\cdot)}$-heuristic* w.r.t. infinitely many input lengths n.

In other words, the notion of a heuristic* relaxes the notion of an errorless heuristic by allowing the heuristic to also make mistakes on YES instances, but strengthens the standard notion of a two-sided error heuristic by requiring the heuristic to succeed not only with high probability over random instances, but also *conditioned* on YES (and NO) instances. The reason we model the error probability, μ, as a function of the logarithm of the number of YES-instances, n^*, as opposed to just n (i.e., the logarithm of the number of instances) is to ensure that this notion meaningfully relaxes the notion of a one-sided heuristic, also for very sparse languages.[4]

We emphasize that the notion of mild average-case* hardness lies in between mild average-case hardness w.r.t. errorless heuristics and mild average-case hardness: Any language that is mildly HoA is also mildly HoA*; Any language that is mildly HoA* is also mildly HoA w.r.t. errorless heuristics.

The Main Theorem We are now ready to state our main theorem. We restrict our attention to "nice" classes of running-times, where the class of functions $\mathcal{F}$ is said to be nice if (1) every function $T \in \mathcal{F}$ is time-constructible and strictly monotonically increasing in the sense that there exists some constant $\nu > 0$ such that for every $n > 1$, $T(n+1) - T(n) \geq \nu$, and (2) $\mathcal{F}$ is closed under polynomial composition: $t \in \mathcal{F}$ implies that $t(n^{\varepsilon})^{\varepsilon'} \in \mathcal{F}$ for every $0 < \varepsilon, \varepsilon' < 1$. Examples of "nice" classes of super-polynomial functions include: (a) the class of subexponential functions, $\mathcal{F}_{\mathsf{subexp}} = \{2^{cn^{\varepsilon}}\}_{c>0, 0<\varepsilon<1}$, (b) the class of quasi-polynomial functions $\mathcal{F}_{\mathsf{qpoly}} = \{n^{c \log n}\}_{c>0}$, or (c) various classes of just slightly super-polynomials functions such as $\{n^{c_0 + c_1 \log\log n}\}_{c_0, c_1 > 0}$, or $\{n^{c_0 + c_1 \log\log\log\log n}\}_{c_0, c_1 > 0}$. (Note that the class of exponential functions is not "nice" as it is not closed under polynomial composition.)

Given a class of functions $\mathcal{F}$, let $\mathcal{F}^{-1}$ denote the class of inverse function: $\mathcal{F}^{-1} = \{f \text{ s.t. } f^{-1} \in \mathcal{F}\}$. For instance, $\mathcal{F}^{-1}_{\mathsf{subexp}} = \{c \log^{\beta} n\}_{c>0, \beta>1}$, and $\mathcal{F}^{-1}_{\mathsf{qpoly}} = \{2^{c\sqrt{\log n}}\}_{c>0}$. Our main theorem is as follows:

Theorem 1.1. *Let $\mathcal{F}$ be a "nice" class of super-polynomial functions, let $\varepsilon > 0$, and let $t(n)$ be a polynomial $t(n) \geq (1+\varepsilon)n$. The following are equivalent:*

(a) There exists a function $T \in \mathcal{F}$ such that T-hard (non-uniformly secure) one-way functions exist.

(b) There exists a function $s \in \mathcal{F}^{-1}$, a constant $\tau \geq 0$, and a constant $0 < \delta < 1$, such that $\mathsf{MK}^t\mathsf{P}[s(n) + \tau]$ is mildly HoA w.r.t., non-uniform algorithms with running-time bounded by n^{δ}.*

(c) There exists a function $T \in \mathcal{F}$ such that for any constant $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA w.r.t., non-uniform algorithms with running-time bounded by T.*

Note that the equivalence between (a) and (b) demonstrates the above-mentioned quantitative characterization of the hardness of OWFs through the threshold s of $\mathsf{MK}^t\mathsf{P}[s]$, and the equivalence of (b) and (c) demonstrates the above-mentioned "sliding scale" property of $\mathsf{MK}^t\mathsf{P}[s]$.

[4] If we hadn't, then a $1/n$-heuristic* could not make *any* mistakes on YES instances when the languages contains less than n YES-instances.

Additionally, we highlight that the implications that (b) implies (c) implies (a) hold also in the setting of uniform security (i.e., w.r.t., uniform algorithms). It is only in the implication that (a) implies (b) where we require security w.r.t.. non-uniform algorithms.

The Lower Bounds Our first lower bound demonstrates *worst-case hardness* of $\mathsf{MK}^t\mathsf{P}[s]$ with respect to uniform probabilistic sublinear-time algorithms, even for very small thresholds $s(\cdot)$.

Theorem 1.2. *Consider any $\delta < 1$, and any $\omega(1) < s(\cdot) \leq n - n^\delta - 2$. Then $\mathsf{MK}^t\mathsf{P}[s] \notin \mathsf{BPTIME}(n^\delta)$.*

The idea behind the proof of this theorem is simple: a sublinear-time algorithm can only read a small part of the input, and thus can never hope to *always* distinguish between strings with high Kolmogorov complexity and those with small.

Our second lower bound demonstrates that when the threshold s is large, $s(n) = n - \log n$, then $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA (and thus also mildly HoA*) with respect to not only sublinear uniform algorithms, but even for algorithms that run in time $t(n)/n^3$. In particular, we can get a lower-bound w.r.t. all a-priori bounded polynomial-time algorithms, as long as we pick t to be sufficiently larger than the bound:

Theorem 1.3. *Consider any constant $\alpha > 0$ and any $t(n) > 0$, $s(n) = n - \alpha \log n$ and any $T(n) \leq t(n)/(n^{\alpha+1} \log^3 n)$. Then $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA w.r.t. deterministic T-time uniform algorithms.*

This theorem extends a recent lower bound by Hirahara [Hir20] that establishes average-case hardness of $\mathsf{MK}^t\mathsf{P}[n-1]$ w.r.t. *errorless* deterministic heuristics where $t = n^{\omega(1)}$. As far as we know, our result is the first lower bound demonstrating *two-sided error* average-case hardness for time-bounded Kolmogorov complexity.

1.2 Related Work: Hardness Magnification

In the last few years, there has been an exciting thread of work on *hardness magnification* [OS18, MMW19, CT19, OPS19, CMMW19, Oli19, CJW19, CHO$^+$20] (see also [Sri03, AK10, LW13, MP20] for related previous work), showing how weak lower bounds for certain sparse languages imply breakthrough separations in complexity theory, such as $\mathsf{NP} \not\subseteq \mathsf{P}/poly$ or $\mathsf{EXP} \not\subseteq \mathsf{NC}^1$. In particular, [CJW19] show such results for all sparse languages. On a high-level, these results are proven by showing how to "compress" an instance in the sparse language (e.g., by sampling parts of the bits of the instance) into another instance (in a different languages) that is much smaller, yet the process preserves (either always, or with high probability) the validity of the statement—since the instance now has become much smaller, we can afford to run a stronger attacker on it, while still making sure the attacker is "weak" with respect to the original instance size.

Conceptually speaking, our results fall into this thread of work, and we also rely on the above-mentioned subsampling technique. However, before our work, no types of hardness magnification results were known for OWFs.

1.3 Related Work: Fine-grained Complexity

We mention a few recent elegant works developing cryptographic schemes assuming fine-grained hardness of some computational problems (i.e., assuming hardness w.r.t. $n^{1+\alpha}$-time attackers for some *fixed* constant $\alpha > 0$) [BRSV17, BRSV18, GR18, LLW19, BABB19, DLW20]. In particular, [BRSV17, GR18, BABB19, DLW20] shows worst-case to average-case reductions for certain natural classes of problems in the fine-grained regimes; [BRSV18] shows the existence of so-called "proofs of work" assuming fine-grained worst-case hardness of these problems; [LLW19, DLW20] constructs a "fine-grained" analog of one-way functions, assuming fine-grained average-case hardness of cetain specific languages. What all these results have in common is that if we start off with a fine-grained lowerbounds, then the resulting cryptographic primitive we get (e.g., a OWF), will also only be secure w.r.t. weak (a-priori bounded polynomial-time) attackers.

In contrast, our results show how to get "real" (as opposed to "fine-grained") OWFs, where the gap between the time needed to evaluate and invert is *super-polynomial* (as opposed to some fixed polynomial), from very weak fine-grained lower bounds. Additionally, our results demonstrate that the average-case problems we consider are necessary for the existence of OWFs. Finally, we only assume *sublinear* hardness (i.e., n^{δ}-hardness for any $\delta > 0$) as opposed to superlinear hardness (i.e., $n^{1+\alpha}$-hardness for $\alpha > 0$).

1.4 Overview of the Proof of the Main Theorem

We proceed to provide a high-level overview of the proof of the main theorem. Consider some "nice" class of functions $\mathcal{F}$, and some polynomial $t(n) \geq (1+\varepsilon)n$. In Part 1, we show that (b) implies (c), in Part 2, we show that (c) implies (a), and in Part 3, we show that (a) implies (b).

Part 1: Hardness Magnification for $\mathsf{MK}^t\mathsf{P}[s]$ Recall that (b) says that there exists a function $s \in \mathcal{F}^{-1}$, such that $\mathsf{MK}^t\mathsf{P}[s(n)]$ is mildly HoA* w.r.t., sublinear-time attackers, whereas (c) says that there exists some function $T \in \mathcal{F}$ such that for any constant $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* w.r.t., T-time attackers. Thus, proving that (b) implies (c) is a hardness magnification result: an average-case lower bound for $\mathsf{MK}^t\mathsf{P}[s(n)]$ (for a small threshold s) w.r.t. *weak* attackers, implies an average-case lower bound for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ w.r.t. *strong* attackers. For concreteness, let us focus on the case when $\mathcal{F} = \mathcal{F}_{\mathsf{subexp}}$ in which case $s(n) = \mathsf{poly}\log n$, but the same proof outline works for general "nice" classes of functions. To prove the hardness magnification result, we need to show how to transform a subexponential-time heuristic $\mathcal{H}'$ for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ into a sublinear-time heuristic $\mathcal{H}$ for $\mathsf{MK}^t\mathsf{P}[s]$. Given an instance x, our heuristic $\mathcal{H}$ for $\mathsf{MK}^t\mathsf{P}[s]$, simply *truncates* the instance to just $\gamma s(n)$ bits[5] (i.e., it keeps the first $\gamma s(n)$ bits) and runs the subexponential-time heuristic $\mathcal{H}'$ on the truncated (short) instance x'. Note that since x' is so short, we can afford to run a subexponential-time heuristic on it, and this just runs in sublinear time in the length of the original instance x.

We now need to argue that $\mathcal{H}$ also succeeds with high probability, conditioned on both YES and NO instances. Let us start with YES instances. First, note that if x is a YES-instance for $\mathsf{MK}^t\mathsf{P}[s]$, then x' will also be YES-instance for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$: the same program, of length $\leq s(n)$, that computes x will also compute x', thus $K^t(x') \leq s(n)$; and since $n' = |x'| = \gamma s(n)$, we have that $K^t(x') \leq n'/\gamma$. But this is not enough to argue that $\mathcal{H}$ succeeds with high probability, as the truncated x' is not distributed as a *random* YES-instance of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$, and we are only guaranteed that $\mathcal{H}'$ succeeds when sampling random YES-instances. However, we can show using a counting argument that the *relative* distance between the distribution of truncations of YES-instances for $\mathsf{MK}^t\mathsf{P}[s]$, and the distribution of YES-instances for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is not too large. We can next use this to argue that $\mathcal{H}$ will also succeed with high probability.

Next, consider a NO-instance x for $\mathsf{MK}^t\mathsf{P}[s]$. By truncating x into x', x' could actually become a YES-instance for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$, so the reduction does not preserve worst-case hardness of the underlying problem. But we do not have to: We only need to show that $\mathcal{H}$ succeeds well on average. To do this, note that random NO-instances for $\mathsf{MK}^t\mathsf{P}[s]$ are statistically close to uniform, and thus the

[5] The actual heuristic $\mathcal{H}$ that we describe in the formal proofs needs to perform a more careful truncation argument due to the fact that $\mathcal{H}'$ may only succeed on infinitely many input lenghts. We refer the reader to the formal proof for further details.

distribution of x' is also statistically close to uniform, which is statistically close to random NO-instances for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$. Thus, if $\mathcal{H}'$ succeeds with high probability over random NO-instances of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$, $\mathcal{H}$ will succeed with high probability over random NO-instances of $\mathsf{MK}^t\mathsf{P}[s]$

Part 2: T-Hard OWFs from T-Average-case Hardness of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ To show that (c) implies (a), we need to construct a T-hard OWF assuming $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* for some $\gamma > 1$ w.r.t. T-time (non-uniform) algorithms. To do this, we leverage the construction from [LP20], which shows a OWF assuming K^t is mildly HoA to compute. We observe that a similar proof can be used to obtain a OWF assuming $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA*—the key difference here is that we need to show how to use a OWF inverter to not only compute K^t over random instances, but also conditioned on YES/NO-instances. We additionally observe that the proof can be extended to yield T-hard OWF assuming T hardness of the underlying problem (by revisiting the hardness amplification theorem of [Yao82] and showing it applies in these more general parameter regimes).

We emphasize that Parts 1 and 2 hold both w.r.t.., uniform and non-uniform algorithms.

Part 3a: Sublinear-time Average-case Hardness of $\mathsf{MK}^t\mathsf{P}[s]$ from T-Hard cond EP-PRFs We finally show that (a) implies (b); that is, that T-hard OWF for some $T \in \mathcal{F}$ implies that there exists $s \in \mathcal{F}^{-1}$ such that $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA* w.r.t. sublinear-time non-uniform algorithms. This is the technically most involved step of the proof, and here we require the OWF to be secure wr.t. *non-uniform* T-time attackers.

Recall that in [LP20], it was shown that OWFs imply that $\mathsf{MK}^t\mathsf{P}[n-O(\log n)]$ is mildly HoA. We here need to extend this results to apply for a much smaller threshold $s \in \mathcal{F}^{-1}$. It is instructive to briefly recall the high-level approach in [LP20]: An object called an *entropy-preserving* pseudorandom generator (EP-PRG) was introduced there. 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 $n - O(\log n)$. In fact, [LP20] did not manage to construct such an EP-PRG from OWFs, but rather constructed a relaxed form of an EP-PRG, called a *conditionally-secure* entropy-preserving PRG (cond-EP PRG), which relaxes both the pseudorandomness, and entropy-preserving properties of the PRG, to hold only conditioned on some event E. [LP20] showed how such a cond EP-PRG can be constructed from OWFs, and next showed that the existence of cond EP-PRGs implies that $\mathsf{MK}^t\mathsf{P}[n-O(\log n)]$ is mildly HoA. Roughly speaking, the idea is that a $\mathsf{MK}^t\mathsf{P}[n-O(\log n)]$ heuristic distinguishes outputs from the PRG and uniform, as uniform string with high probability have high K^t-complexity, whereas outputs of the PRG has small K^t-complexity, and the entropy-preserving property is needed to ensure that the heuristic still works on outputs of the PRG.

Our high-level approach here is similar, but since we need to deal with a much smaller threshold, we need to construct an appropriate *conditionally-secure entropy-preserving* analog of a *pseudorandom function* [GGM84], a *cond EP-PRF*. The entropy-preserving property of such a cond EP-PRF requires that the n-bit prefix of the *truthtable* of the PRF f_s given a seed $s \in \{0,1\}^n$ has Shannon entropy at least $n - O(\log n)$.

We emphasize that proving that the entropy-preserving property of the PRF ensures that the heuristic will work on outputs of the PRF is more subtle than the earlier proof in [LP20] using an entropy-preserving PRG, as now, the entropy-preserving property does not guarantee that the output of the PRF is dense among random string; however, we can still show that it is dense among YES-instances for $\mathsf{MK}^t\mathsf{P}[s]$. We also want to highlight that there is another important subtlety that arises in this proof: The heuristic we are given only needs to work on infinitely many input lengths, so to use this heuristic to break the PRF we need to know on what inputs lenghts to query the heuristic. We solve this problem by using *non-uniformity* and simply assume that the PRF breaker can get these input lengths as non-uniform advice.[6] We emphasize that this is the only step in the proof where non-uniformity is used; all other steps work w.r.t. both uniform and non-uniform algorithms.

Part 3b: T-Hard cond EP-PRFs from T-Hard OWFs So, it just remains to construct a cond EP-PRF. A-priori, this seems easy: why not just use the GGM [GGM84] construction of a PRF from a PRG? The problem is that this transformation does not work if the underlying PRG is a cond EP-PRG: the PRG property only holds conditioned on some event E, which prevents using repeated applications of the PRG. Nevertheless, we show that any standard PRF (which can be constructed from OWFs [GGM84, HILL99] can be combined, in a rather straightforward way, with a cond EP-PRG with *sublinear* stretch (i.e., it expands n bits to $n + n^{\varepsilon}$ bits) to get a cond EP-PRF: first use the cond EP-PRG to expand its seed s into s_1, s_2 where s_1 has n bits and s_2 has $O(n^{\varepsilon})$ bits; next, use s_1 as the description of a truthtable to determine the output of the cond EP-PRF of inputs $x \leq n$, and use s_2 as a seed to the (standard) PRF to determine the outputs of the cond EP-PRF on inputs $x > n$.[7]

Unfortunately, we are not aware of any constructions of cond EP-PRGs with *sublinear stretch* from OWFs. The cond EP-PRG from OWF constructed in [LP20] only expands its seed by $O(\log n)$ bits, and this is inherent in the construction.[8] A central technical contribution of this paper is the construction of a cond EP-PRG with sublinear stretch. The construction is actually very simple, but proving it secure is significantly less so. Given any PRG G (which can be constructed from OWFs [HILL99]), the construction tries to "massage" G to become entropy-preserving. If G were regular, with regularity r, that would be easy: We simply apply pairwise-independent hash functions (that act as strong extractors) h_1 to the input (the seed) of the PRG (parametrized to match the regularity r) to "squeeze" out randomness from the input. However, it is unlikely that G is regular. Instead, we attempt to guess the "degeneracy"[9] of the input x on which G is applied, and extract out the remaining entropy in it, and rely on the fact that we only need a PRG that is secure conditioned on some appropriate event. The construction proceeds as follows:

$$G'(i, x, h_1, h_2) = h_1 || h_2 || h_2 \left(G(x) || [h_1(x)]_{i - O(\log n)} \right)$$

We show that conditioned on the event E that i equals the degeneracy of x, G' preserves the entropy of its input (up to an additive term of $O(\log n)$). More interestingly, we show that, conditioned on E, the output of G', is also pseudorandom. While this latter claim seems intuitive, it is significantly harder to prove. Intuitively, conditioned on E, $[h_1(x)]_{i-O(\log n)}$ is statistically close to a uniform string, and thus leaking it should not harm the pseudorandomness of $G(x)$, so once we apply the extractor h_2 to the combined output, we should be able to extract many pseudorandom bits. This argument is not quite true: even though $[h_1(x)]_{i-O(\log n)}$ is statistically close to a random string of length i, the *length* i itself is already leaking something about the seed x, which makes it hard to formalize this argument. To formally prove it, we need to show a reduction that uses a distinguisher for G' to distinguish the output of G from random; while this reduction can simulate $[h_1(x)]_{i-O(\log n)}$ by outputting random bits, the reduction does not know the degeneracy of the seed x so it does

[6] This issue was not a problem in [LP20] as the PRG used there only expanded n bits to $\ell(n) = n + O(\log n)$ bits. Since $\ell(n+1) - \ell(n) \leq O(1)$ one could argue that constant size advice (which can be incorporated into the description of the TM) suffices to hit infinitely many inputs lenghts on which the heuristic works.

[7] In the actual construction, we also require an additional padding trick to get a cond EP-PRF with small running time. We refer the reader to the techincal sections for further details.

[8] We note that the standard construction of a length-doubling PRG from a PRG with small expansion fails for cond EP-PRGs, for the same reason as the GGM construction fails.

[9] That is, the logarithm of the number of pre-images of $G(x)$.

not know how many random bits to concatenate to $G(x)$! And, simply guessing a length i does not work as the distinguisher could fare badly if the guess is incorrect. We present a method around this problem and manage to formally prove the construction secure.

On a very high-level, the idea is as follows. Assume there exists a distinguisher D' that distinguishes the output of G', conditioned on E, from uniform for infinitely many input lengths n. We may assume without loss of generality that for infinitely many n, D actually outputs 1 with higher probability when given a pseudorandom sample than a uniform sample. We construct a distinguisher D that distinguishes the output of G—*without any conditioning*—from uniform. Given a sample x, D tries *all* lengths $i \in [n]$, and for each such i, estimates the probability that $D'(h_1||h_2||h_2(x|U_i))$ outputs 1 (by running D' many times on new samples U_i and fresh hash functions h_1, h_2). It picks the length i on which $D'(h_1||h_2||h_2(x|U_i))$ outputs 1 with the highest probability, and finally outputs $D'(h_1||h_2||h_2(x|U_i))$ (for new freshly sampled h_1, h_2, U_i). Intuitively, the reason this distinguisher works is that (1) given a pseudorandom sample x, $D(x)$ will output 1 with (roughly) at least as high probability that D' outputs 1 given a pseudorandom sampled conditioned on E, yet (2) when x is truly uniform, then no matter what length i that D' selects, the probability that $D'(h_1||h_2||h_2(x|U_i))$ outputs 1 is roughly the same, and in fact, close to the probability D' outputs 1 given a uniform sample; this follows from the fact the min-entropy of $x||U_i$ is at least the min-entropy of x, which is n, so the output of $h_2(x||U_i)$ will be close to uniform. So we conclude that for infinitely many n, D distinguishes pseudorandom strings from random with roughly the same probability that D' distinguishes the output of the cond EP-PRG, conditioned on E, from uniform.

2 Preliminaries

Given a string x, we let $[x]_j$ denote the first j bits of x. Let $\mathsf{tt}(f)$ denote the truth table of a function f, $\mathsf{tt}_m(f) = [\mathsf{tt}(f)]_m$ denote the m-bit prefix of the truth table. For a truth table z, let $\mathsf{fn}(z)$ denote the function associated with it. We say that a function f is *super-polynomial* if for every $c \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $f(n) \geq n^c$. A function μ is said to be *negligible* if for every polynomial $p(\cdot)$, there exists some $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $\mu(n) \leq \frac{1}{p(n)}$. We say that a family of functions $\mathcal{F}$ is *closed under (sublinear) polynomial compositions* if for any $f \in \mathcal{F}$, for all $0 < \varepsilon_1, \varepsilon_2 < 1$, $(f(n^{\varepsilon_1}))^{\varepsilon_2} \in \mathcal{F}$. We say that a function f is *time-constructible* if for all $n \in \mathbb{N}$, $f(n)$ can be computed by a Turing machine in time $\mathsf{poly}(f(n))$. For a strictly monotonic function f, let $f^{-1}(n)$ denote the inverse of f; that is, the unique function such that $f^{-1}(f(n)) = n$. We assume familiarity with basic concepts such as deterministic and probabilistic Turing machines. A *uniform* T-time algorithm A is a probabilistic Turing machine that on inputs x of length n runs in time at most $T(n)$. A *non-uniform* T-time algorithm A is specificed by a pair $(A', \{z_n\}_{n\in\mathbb{N}})$ where A' is a probabilistic Turing machine such that given any input x of length n, $A'(x, z_n)$ terminates within $T(n)$ steps; given any input x of lenght n, we refer to the output of $A(x)$ as the output of $A'(x, z_n)$.[10]

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 $\{0,1\}^n$. For any integer $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

2.1 Time-bounded Kolmogorov Complexity

We define the notion of t-time-bounded Kolmogorov complexity that we rely on. We consider some universal Turing machines U that can emulate any Turing machine M with polynomial overhead.

[10] While in our context, hardness w.r.t. non-uniform *probabilistic* T-time algorithms is equivalent to hardness w.r.t. non-uniform *deterministic* T-time algorithms (since we can always fix the best random coins of the attacker), we prefer to explicitly model attackers a non-uniform probabilistic algorithms to clarify why many of our proofs extend to the setting of *uniform* probabilistic algorithms.

The universal Turing machine U receives as input a description/program $\Pi \in \{0,1\}^* = (M, w)$ where M is a Turing machine and $w \in \{0,1\}^*$ is an input to M; we let $U(\Pi(i), 1^{t(|\Pi|)})$ denote the output of $M(w, i)$ when emulated on U for $t(|\Pi|)$ steps.

Definition 2.1. *Let U be a universal Turing machine and $t(\cdot)$ be a polynomial. Define*

$$K^t(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \forall i \in [|x|], U(\Pi(i), 1^{t(|\Pi|)}) = x_i\}.$$

We remark that the notion of time-bounded Kolmogorov complexity has been defined in a lot of different ways [Kol68, Sip83, Tra84, Ko86, ABK+06]; the definition we consider here is the "local compression" version (see e.g., [ABK+06]) where the program Π is required to efficiently output each individual bit x_i of the string x, given i as input. This notion captures the "truthtable" compression problem discussed in the introduction, where we think of x as the truthtable of a function. (A different version of time-bounded Kolmogorov complexity instead considers the size of the shortest program that outputs the *whole* string x. This other notion refers to a "global compression" notion, but is less appealing from the point of view of truthtable compression, as the running-time of the program can never be smaller than the length of the truthtable x. We do not consider the global compression notion in this paper.)

Let $\mathsf{MK}^t\mathsf{P}[s(n)]$ be a language consisting of strings x with K^t-complexity at most $s(|x|)$. We recall the following fact about (time-bounded) Kolmogorov complexity.

Fact 2.1. *There exists a constant c such that for every polynomial $t(n) \geq (1+\varepsilon)n, \varepsilon > 0$, the following holds:*

(1) For every $x \in \{0,1\}^*$, $K^t(x) \leq |x| + c$;

(2) For every integer $n \in \mathbb{N}$, every function $0 < s(n) < n$, $2^{\lfloor s(n) \rfloor - c} \leq |\mathsf{MK}^t\mathsf{P}[s(n)] \cap \{0,1\}^n| \leq 2^{\lfloor s(n) \rfloor + 1}$.

Proof: Let M be a Turing machine such that $M(w, i)$ outputs w_i if $i \leq |w|$, and otherwise outputs 0. The running time of M can be bounded by $t(|w|)$, and M can be encoded in c bits (for some universal constant c). Thus, (1) follows from the fact that any string x has a description $\Pi_x = (M, x)$, which can be encoded in $|x| + c$ bits (see [Sip96] for simple treatment). For (2), since the number of descriptions with size at most $s(n)$ is $2^{\lfloor s(n) \rfloor + 1}$ and a single description could only produce a singe string $\in \{0,1\}^n$, it follows that $|\mathsf{MK}^t\mathsf{P}[s(n)] \cap \{0,1\}^n| \leq 2^{\lfloor s(n) \rfloor + 1}$. To get a lower bound on $|\mathsf{MK}^t\mathsf{P}[s(n)] \cap \{0,1\}^n|$, note that every n-bit string of the form $y||0^{n - \lfloor s(n) \rfloor - c}$ where $y \in \{0,1\}^{\lfloor s(n) \rfloor - c}$ can be described by the program $\Pi_y = (M, y)$ of size $\lfloor s(n) \rfloor$ and the number of such strings is $2^{\lfloor s(n) \rfloor - c}$. Thus, $|\mathsf{MK}^t\mathsf{P}[s(n)] \cap \{0,1\}^n| \geq 2^{\lfloor s(n) \rfloor - c}$. ■

2.2 Average-case Complexity

We recall the definitions of average-case hardness and average-case hardness w.r.t. errorless heuristics. We focus our attention on average-case hardness w.r.t., non-uniform algorithms.

Definition 2.2 (Average-case Hardness)**.** *We say that a language L is* $\alpha(\cdot)$ hard-on-average (α-HoA) for $T(\cdot)$-time heuristics *if for all probabilistic non-uniform $T(\cdot)$-time heuristics $\mathcal{H}$, for all sufficiently large $n \in \mathbb{N}$,*

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

Definition 2.3 (Average-case Hardness w.r.t. Errorless Heuristics)**.** *We say that a language L is* $\alpha(\cdot)$ hard-on-average (α-HoA) for $T(\cdot)$-time errorless heuristics *if for all probabilistic non-uniform $T(\cdot)$-time heuristics $\mathcal{H}$, for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = \{L(x), \bot\}] \neq 1$$

or

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = \bot] > \alpha(n)$$

In other words, there does not exist a T-time heuristic (resp errorless heuristic) that decides L with probability $1 - \alpha(n)$ for infinitely many n, where an errorless heuristic never outputs $\neg L(x)$ on input x (but it can output $\bot$ for "I don't know").

We will sometime also consider average-case hardness with respect to more restrictive classes of heuristics (e.g., uniform heuristics, or uniform and deterministic heuristics), and then explicitly state so.

In this work, we introduce a notion of average-case hardness w.r.t. two-sided error heuristics which is meaningful also for *sparse* languages. Before describing the definition, let us first define the density of a language: We say that a language $L \subset \{0,1\}^*$ is $D(\cdot)$-dense if for all $n \in \mathbb{N}$, $|L_n| = D(n)$, where $L_n = L \cap \{0,1\}^n$. Now we are ready to define the notion of average-case* hardness.

Definition 2.4 (Average-case* Hardness)**.** *We say that a $D(\cdot)$-dense language L is* $\alpha(\cdot)$ hard-on-average* (α-HoA*) for $T(n)$-time heuristics *if for all probabilistic non-uniform $T(n)$-time heuristics $\mathcal{H}$, for all sufficiently large n, there exist $\mu \in \{0,1\}$ such that,*

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = \mu \mid L(x) = \mu] < 1 - \alpha(n^*),$$

where $n^* = \log D(n)$.

In other words, there does not exist a T-time "heuristic*" that decides L with probability $1-\alpha(n^*)$ *conditioned* on YES (and NO) instances. We say that a language L is α-HoA* *for sublinear-time heuristics* if there exists $0 < \delta < 1$ such that L is α-HoA* for n^δ-time heuristics. We say that a language L is *mildly* HoA* (resp HoA, HoA w.r.t. errorless heuristics) for $T(n)$-time heuristics if there exists a strictly increasing polynomial $p(\cdot) > 0$ such that L is $\frac{1}{p(\cdot)}$-HoA* (resp HoA, HoA w.r.t. errorless heuristics) for $T(n)$-time heuristics.

The following two lemmas show that mild average-case* hardness is a notion that lies between mild average-case hardness w.r.t. errorless heuristics and mild average-case hardness.

Lemma 2.2. *If a language L is mildly HoA for $T(n)$-time heuristics, then L is mildly HoA* for $T(n)$-time heuristics.*

Lemma 2.3. *If a language L is mildly HoA* for $T(n)$-time heuristics, then L is mildly HoA w.r.t. $T(n)$-time errorless heuristics.*

We refer the reader to Appendix A for the (straight-forward) proofs of the above two lemmas.

2.3 One-way Functions

We recall the standard definitions of one-way functions (with security w.r.t. non-uniform efficient attackers).

Definition 2.5. *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a* (T, ε)-one-way function *if for any probabilistic non-uniform algorithm $\mathcal{A}$ of running time $T(n)$, for all sufficiently large $n \in \mathbb{N}$,*

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

We say that f is $T(n)$-one-way (or is a T-hard one-way function) if f is $(T(n), 1/T(n))$-one-way. We say that f is $\varepsilon(n)$-weak $T(n)$-one-way if f is $(T(n), 1-\varepsilon(n))$-one-way. If $\varepsilon(n)$ is a (monotonically increasing) polynomial, we say f is *weakly* $T(n)$-one-way. We say that f is simply *one-way* if f is $T(n)$-one-way for all polynomials $T(n)$. When $T(n)$ is a super-polynomial function, we refer to f as being *subexponentially-secure* (resp *quasi-polynomially-secure*) if there exists a constant $c > 0$ such that f is 2^{n^c}-one-way (resp $n^{c \log n}$-one-way).

We recall the hardness amplification lemma [Yao82] which was originally stated for (polynomially-hard) OWFs; we here extend it to work for T-one-way functions. We defer the proof (which is a simple generalization of Yao's proof) to Appendix A.

Lemma 2.4 (Hardness Amplification [Yao82])**.** *Assume that there exists a weakly $T(n)$-one-way function for an arbitrary function $T(\cdot)$. Then, there exists a $(T'(n))$-one-way function where $T'(n) = \sqrt{\frac{T(n^{\Omega(1)})}{n^{O(1)}}} - n^{O(1)}$.*

Finally, we note that when we assume that there exists a OWF f, we can assume without loss of generality that f is length-preserving.

Lemma 2.5 (Length-preserving OWFs from OWFs [HHR06])**.** *Assume that $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is a $(T(n), \varepsilon(n))$-one-way function. Then there exists a length-preserving $(T(n^{\Omega(1)}) - n^{O(1)}, 2\varepsilon(n^{\Omega(1)}))$-one-way function.*

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 $(T(\cdot), \varepsilon(\cdot))$-*indistinguishable*, if for every probabilistic non-uniform $T(\cdot)$-time machine D (the "distinguisher") whose running time is $T(\cdot)$ 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 are $\{A_n\}_{n\in\mathbb{N}}$ and $\{B_n\}_{n\in\mathbb{N}}$ simply indistinguishable *if they are $(T(\cdot), \frac{1}{p(\cdot)})$-indistinguishable for all polynomials $p(\cdot)$, $T(\cdot)$.*

2.5 Pseudorandom Generators and Pseudorandom Functions

We recall the standard definitions of pseuodrandom generators (PRGs) and pseudorandom functions (PRFs).

Definition 2.7. *Let $g : \{0,1\}^n \to \{0,1\}^{m(n)}$ be a polynomial-time computable function. g is said to be a $(T(\cdot), \varepsilon(\cdot))$-pseudorandom generator if for any probabilistic non-uniform $T(\cdot)$-time algorithm $\mathcal{A}$ (whose running time is $T(\cdot)$ in the length of its first input), for all sufficiently large n,*

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

Definition 2.8. *Let $f : \{0,1\}^n \times \{0,1\}^{k(n)} \to \{0,1\}$ be a polynomial-time computable function. f is said to be a $(T(\cdot), \varepsilon(\cdot))$-pseudorandom function if for any probabilistic non-uniform $T(\cdot)$-time algorithm $\mathcal{A}$, for all sufficiently large n,*

$$|\Pr[x \leftarrow \{0,1\}^n : \mathcal{A}^{f(x,\cdot)}(1^n) = 1] - \Pr[f' \leftarrow \mathcal{F} : \mathcal{A}^{f'}(1^n) = 1]| \leq \varepsilon(n)$$

where $\mathcal{F} = \{f' : \{0,1\}^{k(n)} \to \{0,1\}\}$.

In addition, we recall the following classic results on constructing PRGs and PRFs from OWFs.

Theorem 2.9 (1-bit stretching PRGs from Length-preserving OWFs [HILL99, HRV10])**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a $(T(n), \varepsilon(n))$-one-way function where $\varepsilon(n) \leq 1/n^c$ for some constant c such that $\varepsilon(n)$ is polynomial-time computable. There exists an efficient generator g from strings of length $d = d(n) = n^{O(1)}$ to strings of length $d+1$ such that for any polynomial-time computable function $\varepsilon'(n)$, G is a $(T(n) \cdot (\varepsilon'(n)/n)^{O(1)}, \varepsilon'(n) \cdot n^{O(1)})$-PRG.*

Lemma 2.6 (Length-doubling PRGs from 1-bit stretching PRGs [Gol01])**.** *Assume that there exists a $(T(n), \varepsilon(n))$-PRG $g : \{0,1\}^n \to \{0,1\}^{n+1}$. Then there exists a $(T(n), \varepsilon(n) \cdot n)$-PRG $g' : \{0,1\}^n \to \{0,1\}^{2n}$.*

Theorem 2.10 (PRFs from Length-doubling PRGs [GGM84])**.** *Assume that there exists a length-doubling $(T(n), \varepsilon(n))$-PRG g with running time $r_g(n)$. For any time-constructible function $0 < T'(n) < 2^n$, there exists a $(T(n) - O(n), T'(n)n\varepsilon(n))$-PRF $f : \{0,1\}^n \times [T'(n)] \to \{0,1\}$ with running time $r_g(n) \cdot \log T'(n)$.*

2.6 Statistical Distance and Entropy

For any two random variables X and Y defined over some set $\mathcal{V}$, we let $\mathsf{SD}(X,Y) = \max_{T \subseteq U} |\Pr[X \in T] - \Pr[Y \in T]| = \frac{1}{2}\sum_{v \in \mathcal{V}} |\Pr[X = v] - \Pr[Y = v]|$ denote the *statistical distance* between X and Y. It will be helpful to note that the expression is maximixed when the "distinguisher" $T = \{\omega : \Pr[X = \omega] > \Pr[Y = \omega]\}$. 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.

We recall a simple lemma from [LP20] showing that any distribution that is statistically close to random has very high Shannon entropy.

Lemma 2.7 ([LP20])**.** *For every $n \geq 4$, the following holds. Let X be a random variable over $\{0,1\}^n$ such that $\mathsf{SD}(X, \mathcal{U}_n) \leq \frac{1}{n^2}$. Then $H(X_n) \geq n - 2$.*

2.7 Universal Hash Functions

We recall the notion of a universal hash function [CW79].

Definition 2.11. *Let $\mathcal{H}^n_m$ be a family of functions where $m < n$ and each function $h \in \mathcal{H}^n_m$ maps $\{0,1\}^n$ to $\{0,1\}^m$. We say that $\mathcal{H}^n_m$ is a* universal hash family *if (i) the functions $h_\sigma \in \mathcal{H}^n_m$ can be described by a string σ 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 $y, y' \in \{0,1\}^m$*

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

It is well-known that truncation preserves pairwise independence; see e.g., [LP20] for a proof.

Lemma 2.8. *If $\mathcal{H}^n_m$ is a universal hash family and $\ell \leq n$, then $\mathcal{H}'^n_\ell = \{h_\sigma \in \mathcal{H}^n_m : [h_\sigma]_\ell\}$ is also a universal hash family.*

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

Lemma 2.9 ([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 2.10 (Leftover Hash Lemma (LHL) [HILL99]). *For any integers $d < k \le n$, let $\mathcal{H}^n_{k-d}$ be a universal hash family where each $h \in \mathcal{H}^n_{k-d}$ maps $\{0,1\}^n$ to $\{0,1\}^{k-d}$. Then, for any random variable X over $\{0,1\}^n$ such that $H_\infty(X) \ge k$, it holds that*

$$\mathsf{SD}((H^n_{k-d}, H^n_{k-d}(X)), (H^n_{k-d}, \mathcal{U}_{k-d})) \le 2^{-\frac{d}{2}},$$

where H^n_{k-d} denotes a random variable uniformly distributed over $\mathcal{H}^n_{k-d}$.

2.8 "Nice" Function Classes

We consider "nice" classes of function families, where the class of functions $\mathcal{F}$ is said to be "nice" if

- for every function $T \in \mathcal{F}$, T is time-constructible and strictly increasing in the sense that there exists a constant $\nu > 0$ such that for all $n > 1$, $T(n+1) \ge T(n) + \nu$;
- $\mathcal{F}$ is closed under (sublinear) polynomial compositions: for any $T \in \mathcal{F}$, for all $0 < \varepsilon_1, \varepsilon_2 < 1$, $(T(n^{\varepsilon_1}))^{\varepsilon_2} \in \mathcal{F}$.

Given a class of functions, let $\mathcal{F}^{-1}$ denote the class of inverse function: $\mathcal{F}^{-1} = \{f \text{ s.t. } f^{-1} \in \mathcal{F}\}$. Several examples of "nice" classes of super-polynomial functions (and their inverse classes) are (a) $\mathcal{F}_{\mathsf{subexp}} = \{2^{cn^\varepsilon}\}_{c>0, 0<\varepsilon<1}$ and $\mathcal{F}^{-1}_{\mathsf{subexp}} = \{c \log^\beta n\}_{c>0,\beta>1}$, (b) $\mathcal{F}_{\mathsf{qpoly}} = \{n^{c \log n}\}_{c>0}$ and $\mathcal{F}^{-1}_{\mathsf{qpoly}} = \{2^{c\sqrt{\log n}}\}_{c>0}$.

The notion of "nice" function classes has the important property that "polynomial-time" reductions "preserve $\mathcal{F}$-hardness". Roughly speaking, almost all the reductions (considered in this work) are of form "if A is $T(n)$-hard, B is $(T(n^{\Omega(1)})^{\Omega(1)}/n^{O(1)} - n^{O(1)})$-hard". When A is a language, we refer to A as being $T(n)$-hard if A is mildly HoA* for $T(n)$-time heuristics. When A is a cryptographic primitive, we refer to A as being $T(n)$-hard if A is secure against all $T(n)$-time attackers. The following fact shows that such reductions actually prove the following statement: "if there exists $T_1 \in \mathcal{F}$ such that A is $T_1(n)$-hard, then there exists $T_2 \in \mathcal{F}$ such that B is $T_2(n)$-hard".

Fact 2.11. *Let $\mathcal{F}$ be a nice class of super-polynomial functions. For every $T \in \mathcal{F}$, for all $0 < \varepsilon_1, \varepsilon_2 < 1, c_1, c_2 > 1$, there exists a function $T' \in \mathcal{F}$ such that for all sufficiently large n, $T'(n) \le T(n^{\varepsilon_1})^{\varepsilon_2}/n^{c_1} - n^{c_2}$.*

Proof: Let $T_1(n)$ denote $T(n^{\varepsilon_1})^{\varepsilon_2}$. Since $\mathcal{F}$ is closed under polynomial compositions, $T_1(n) \in \mathcal{F}$. Since T_1 is a super-polynomial function, $T_1(n) \ge n^{4(c_1+c_2)}$. It follows that for all large n, $T_1(n)/(2n^{c_1}) \ge n^{c_2}$, so $T_1(n)/n^{c_1} - n^{c_2} \ge T_1(n)/(2n^{c_1})$. Note that $2n^{c_1} \le T_1(n)^{\frac{1}{2}}$ for sufficiently large n, and thus $T_1(n)/(2n^{c_1}) \ge T_1(n)/(T_1(n)^{\frac{1}{2}}) \ge T_1(n)^{\frac{1}{2}}$. Finally, notice that $T_1(n)^{\frac{1}{2}} \in \mathcal{F}$ since $\mathcal{F}$ is closed under (sublinear) polynomial compositions, which concludes the proof. ■

3 Avg-case Hardness Magnification for $\mathsf{MK}^t\mathsf{P}[s]$

We here show a hardness magnification theorem for $\mathsf{MK}^t\mathsf{P}[s]$. Roughly speaking, it shows that if $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA* even just w.r.t. sublinear algorithms, then for every γ, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* w.r.t. much stronger, T-time algorithms, where T grows as a function of the inverse of s. For instance, if $s(n) = \mathsf{poly}\log n$, then $T(n)$ becomes 2^{n^ε} for some $\varepsilon > 0$.

Lemma 3.1 (Hardness Magnification for $\mathsf{MK}^t\mathsf{P}$)**.** *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exist a function $s \in \mathcal{F}^{-1}$ and an integer constant $\tau \geq 0$ such that $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ is mildly HoA* for sublinear-time heuristics. Then, there exists a function $T \in \mathcal{F}$ such that for every integer $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* for $T(n)$-time heuristics.*

Roughly speaking, we will consider a heuristic $\mathcal{H}$ that given a string $z \in \{0,1\}^m$, *truncates* z into a string y of length, roughly, $\gamma s(m)$, and next runs $\mathcal{H}'(y)$. The problem, however, is that $\mathcal{H}'$ may only work for infinitely many input lengths, so we need to make sure that we truncate the the string z into a length on which actually $\mathcal{H}'$ succeeds. The formal proof deals with this issue. We proceed to the formal proof.

Proof: [of Lemma 3.1] Consider some nice class of super-polynomial functions $\mathcal{F}$, and some function $s \in \mathcal{F}^{-1}$ such that $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ is mildly HoA* for sublinear-time heuristics; that is, there exists some $\delta > 0$ and some monotonically increasing polynomial $p'(\cdot)$ such that $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ is $\frac{1}{p'(\cdot)}$-HoA* for m^δ-time heuristics.

We will show that there exists a function $T \in \mathcal{F}$ such that for every integer $\gamma > 1$, there exists some monotonically increasing polynomial $q'(\cdot)$ such that $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is $\frac{1}{q'(\cdot)}$-HoA* for $T(n)$-time heuristics. Pick any function $T \in \mathcal{F}$ such that $T(n) \leq (s^{-1}(n^{0.99}))^\delta$ (guaranteed to exist by Fact 2.11), and consider any $\gamma > 1$. Let $m^* = \log D_1(m)$ where $D_1(m)$ is the density of $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$. By Fact 2.1, $\lfloor s(m) \rfloor + \tau - c \leq m^* \leq \lfloor s(m) \rfloor + \tau + 1$, so there exists an monotonically increasing polynomial $p(\cdot)$ such that $p(\lfloor s(m) \rfloor + \tau) \geq p'(m^*)$ (for all sufficiently large m and m^*). Let $q(n)$ be a polynomial such that $q(n) = 2^{c+3} n p(n)^2$ for all $n \in \mathbb{N}$. Let $n^* = \log D_2(n)$ where $D_2(n)$ is the density of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$. Let $q'(\cdot)$ be a polynomial guaranteed to exist (due to Fact 2.1) such that $q'(n^*) \geq q(\lfloor n/\gamma \rfloor)$ (for all sufficiently large n and n^*).

Assume for contradiction that there exists a $T(n)$-time heuristic $\mathcal{H}'$ for $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ with success probability at least $1 - \frac{1}{q'(n^*)} \geq 1 - \frac{1}{q(\lfloor n/\gamma \rfloor)}$ conditioned on YES and NO instances on infinitely many n. We will construct a m^δ-time heuristic $\mathcal{H}$ for $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ with success probability at least $1 - \frac{1}{p(\lfloor s(m) \rfloor + \tau)} \geq 1 - \frac{1}{p'(m^*)}$ on infinitely many m, which contradicts to the fact that $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ is $\frac{1}{p'(\cdot)}$-HoA* and concludes the proof.

Constructing the heuristic $\mathcal{H}$: Consider a heuristic heuristic algorithm $\mathcal{H}_{a,b}$ (parametrized by integer constants a, b) that on input $z \in \{0,1\}^m$, just returns $\mathcal{H}'(y)$ where $y = [z]_n$ is the n-bit prefix of z where $n = \lfloor s(m) + \tau + a \rfloor \times \gamma + b$. First, note that for any choice of a, b, $\mathcal{H}'$ runs in time $T(n) = s^{-1}((\lfloor s(m) + \tau + a \rfloor \times \gamma + b)^{0.99})^\delta \leq s^{-1}(s(m))^\delta \leq m^\delta$, so $\mathcal{H}_{a,b}$ runs in time m^δ.

Note that $\mathcal{H}'$ only succeeds on infinitely many input lengths; we will carefully pick a, b such that for infinitely many m, $\mathcal{H}'$ succeeds on input lengths $n = \lfloor s(m) + \tau + a \rfloor \times \gamma + b$. In more detail, let $\{n_1, n_2, n_3, \ldots\}$ be an infinite sequence of input lengths on which $\mathcal{H}'$ succeeds. For all $n \in \mathbb{N}$, n must be of the form $n = \lfloor n/\gamma \rfloor \times \gamma + b$ for some integer $0 \leq b < \gamma$. It follows that there exists an integer constant $b \in [0, \gamma)$ such that there exists an infinite sequence $\{k_1, k_2, k_3, \ldots\}$ where for all i, there exists $j \in \mathbb{N}$ such that $n_j = (k_i + \tau) \times \gamma + b$. Let $\nu > 0$ be the constant such that for all n, $s^{-1}(n+1) - s^{-1}(n) \geq \nu$ guaranteed to exist since $s^{-1} \in \mathcal{F}$. For any $k \in \mathbb{N}$, we claim that there exist an integer m' and an integer $0 \leq a \leq \lfloor \frac{1}{\nu} \rfloor$ such that $k = \lfloor s(m') \rfloor + a$. Note that for all n, $s(n+1) - s(n) \leq \lfloor \frac{1}{\nu} \rfloor$. Let m' be the largest integer such that $\lfloor s(m') \rfloor \leq k$. Since $k < s(m'+1)$, it follows that $a = k - \lfloor s(m') \rfloor \leq \lfloor \frac{1}{\nu} \rfloor$. Thus, there exists an integer $0 \leq a \leq \lfloor \frac{1}{\nu} \rfloor$ and an infinite sequence $\{m_1, m_2, \ldots\}$ such that for all $i \in \mathbb{N}$, there exists j such that $k_j = \lfloor s(m_i) \rfloor + a$. Fix such two constants a, b and consider the heuristic $\mathcal{H} = \mathcal{H}_{a,b}$. By construction, we have that there exist infinitely many m (namely, $\{m_1, m_2, \ldots\}$) such that $\mathcal{H}'$ succeeds on the input lengths $n = (k + \tau) \times \gamma + b = \lfloor s(m) + \tau + a \rfloor \times \gamma + b$ (for some $k \in \{k_1, k_2, \ldots\}$). We shall show that $\mathcal{H}$ will succeed on all these input lengths m.

Fix some sufficiently large such m such that $\mathcal{H}'$ succeeds on the input lengths $n = \lfloor s(m) + \tau + a \rfloor \times \gamma + b$, and let

$$\ell = \lfloor s(m) \rfloor + \tau = \lfloor n/\gamma \rfloor - a.$$

We will show that for any such m, $\mathcal{H}$ is a $(1 - \frac{1}{p(\ell)})$-heuristic for $\mathsf{MK}^t\mathsf{P}[s(m) + \tau] \cap \{0,1\}^m$. Since $1 - \frac{1}{p(\ell)} = 1 - \frac{1}{p(\lfloor s(m) \rfloor + \tau)} \geq 1 - \frac{1}{p'(m^*)}$, we have that $\mathcal{H}$ breaks the $\frac{1}{p'(\cdot)}$-HoA* property of $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$.

Analyzing the success probability of $\mathcal{H}$ on YES-instances: We start by showing that on input a random m-bit YES instance $z \in \mathsf{MK}^t\mathsf{P}[s(m)+\tau]$, $\mathcal{H}$ decides $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$ with probability at least $1 - \frac{1}{p(\ell)} = 1 - \frac{1}{p(\lfloor s(m) \rfloor + \tau)}$. Since $\mathcal{H}'$ succeeds in deciding $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ with probability at least $1 - \frac{1}{q(\lfloor n/\gamma \rfloor)} = 1 - \frac{1}{q(\ell + a)} \geq 1 - \frac{1}{q(\ell)}$ conditioned on YES instances, by an averaging argument, except for a $1 - \frac{1}{2p(\ell)}$ fraction of random tapes r for $\mathcal{H}'$, the *deterministic* machine $\mathcal{H}'_r$ (i.e., machine $\mathcal{H}'$ with randomness fixed to r) fails to solve $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ with probability at most $\frac{2p(\ell)}{q(\ell)}$ (conditioned on YES instances). Fix some such "good" random tape r for which $\mathcal{H}'_r$ decides $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ with probability at least $1 - \frac{2p(\ell)}{q(\ell)}$ conditioned on the input being an n-bit YES instance. Let $\mathcal{H}_r$ denote the deterministic heuristic which uses r as the random tape when invoking $\mathcal{H}'$.

Let S be the set of m-bit YES instances of $\mathsf{MK}^t\mathsf{P}[s(m) + \tau]$ on which $\mathcal{H}_r$ outputs 0. By Fact 2.1, there are at least $2^{\lfloor s(m) \rfloor + \tau - c} = 2^{\ell - c}$ m-bit YES instances in $\mathsf{MK}^t\mathsf{P}[s(m) + \tau]$. Thus, $\mathcal{H}_r(z)$ outputs 0 with probability

$$fail_r \leq \frac{|S|}{2^{\ell - c}}$$

when z is a uniform random m-bit YES instance of $\mathsf{MK}^t\mathsf{P}[s(m) + \tau]$. Consider any string $z \in S$ and let $w = K^t(z)$. Since $z \in \mathsf{MK}^t\mathsf{P}[s(m) + \tau]$, it follows that $w \leq \lfloor s(m) \rfloor + \tau = \ell$, and there exists a machine Π with description length w that produces each bit in the string z in $t(w)$ steps. Specifically, for all $i \in [m]$, $\Pi(i)$ outputs z_i within $t(w)$ steps. Thus, the same machine Π will also produce the string y within $t(w)$ steps where $y = [z]_n$ is the n-bit prefix of z. So, it follows that $K^t(y) \leq w \leq \ell = \lfloor n/\gamma \rfloor - a \leq \lfloor n/\gamma \rfloor$ and thus y belongs to $\mathsf{MK}^t\mathsf{P}[n/\gamma]$. Since $\mathcal{H}_r$ fails on z, $\mathcal{H}'_r$ must also fail on y. Note that the number of n-bit YES instances of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is at most $2^{\lfloor n/\gamma \rfloor + 1} = 2^{\ell + a + 1}$ (by Fact 2.1), so the probability that $\mathcal{H}'_r$ fails conditioned on the input being an m-bit YES instance is at least

$$\frac{|S|}{2^{\ell + a + 1}} \geq fail_r \cdot \frac{1}{2^{c + a + 1}}$$

which by the assumption (that $\mathcal{H}'_r$ is a good heuristic) is at most $\frac{2p(\ell)}{q(\ell)}$. We thus conclude that

$$fail_r \leq \frac{2^{c+a+2} p(\ell)}{q(\ell)}.$$

Then, by a union bound, we have that $\mathcal{H}$ (using a uniform random tape) outputs 0 with probability at most

$$\frac{1}{2p(\ell)} + \frac{2^{c+a+2} p(\ell)}{q(\ell)} = \frac{1}{2p(\ell)} + \frac{2^{c+a+2} p(\ell)}{2^{c+3} \ell p(\ell)^2} \leq \frac{1}{p(\ell)} = \frac{1}{p(\lfloor s(m) \rfloor)}$$

conditioned on z being an m-bit YES instance of $\mathsf{MK}^t\mathsf{P}[s(m) + \tau]$ (since $2^a \leq \ell$ when ℓ is sufficiently large).

Analyzing the success probability of $\mathcal{H}$ on NO-instances: We turn to showing that on m-bit random NO-instances, $z \notin \mathsf{MK}^t\mathsf{P}[s(m) + \tau]$, $\mathcal{H}$ decides $\mathsf{MK}^t\mathsf{P}[s(m) + \tau]$ with probability at least $1 - \frac{1}{p(\lfloor s(m) \rfloor + \tau)}$. We proceed by a hybrid argument. Let

- $Z_1 = \{z \leftarrow \{0,1\}^m : z \notin \mathsf{MK}^t\mathsf{P}[s(m)+\tau]\}$ be the uniform distribution over m-bit NO instances of $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$;
- $Z_2 = \{z \leftarrow \{0,1\}^m\}$ be the uniform distribution over m-bit strings;
- $Z_3 = \{z \leftarrow \{0,1\}^n\}$ be the uniform distribution over n-bit strings;
- $Z_4 = \{z \leftarrow \{0,1\}^n : z \notin \mathsf{MK}^t\mathsf{P}[n/\gamma]\}$ be the uniform distribution over n-bit NO instances of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$.

By Fact 2.1, there are at most $2^{\lfloor s(m)\rfloor+\tau+1}$ m-bit strings that are YES-instances of $\mathsf{MK}^t\mathsf{P}[s(m)+\tau]$, thus there are at most $2^{\lfloor s(m)\rfloor+\tau+1}$ points that have higher probability mass in Z_1 than in Z_2, and the difference in probability mass for each such point is exactly 2^{-m}. By the observation noted after the definition of statistical distance[11], it follows that the statistical distance is upper bounded by

$$2^{-m+\lfloor s(m)\rfloor+\tau+1} = 2^{-m+\ell+1}.$$

By the same argument, we have that the statisical distance between Z_3 and Z_4 is upper bounded by

$$2^{-n+\lfloor n/\gamma\rfloor+1} \leq 2^{-(1-1/\gamma)n+1} \leq 2^{-(1-1/\gamma)\ell+1}.$$

(since, recall, $\ell = \lfloor n/\gamma\rfloor - a \leq n$). On the other hand, if z is distributed uniformly over m-bit strings, the distribution over $y = [z]_n$ is also the uniform over n-bit string. Thus, $[Z_2]_n$ is identically distributed to Z_3. Note that since $\mathcal{H}'$ is a good heuristic,

$$\Pr_{z\leftarrow Z_4}[\mathcal{H}'(z)=1] \leq \frac{1}{q(\lfloor n/\gamma\rfloor)} = \frac{1}{q(\ell+a)} \leq \frac{1}{q(\ell)}.$$

Thus,

$$\begin{aligned}
\Pr_{z\leftarrow Z_1}[\mathcal{H}(z)=1] = \Pr_{z\leftarrow Z_1}[\mathcal{H}'([z]_n)=1] &\leq \Pr_{z\leftarrow Z_4}[\mathcal{H}'(z)=1] + \mathsf{SD}(Z_4,Z_3) + \mathsf{SD}([Z_2]_n,[Z_1]_n)\\
&\leq \Pr_{z\leftarrow Z_4}[\mathcal{H}'(z)=1] + \mathsf{SD}(Z_4,Z_3) + \mathsf{SD}(Z_2,Z_1)\\
&\leq \frac{1}{q(\ell)} + 2^{-(1-1/\gamma)\ell+1} + 2^{-m+\ell+1}\\
&\leq \frac{2}{q(\ell)} \leq \frac{1}{p(\ell)}
\end{aligned}$$

■

Remark 3.2 (A note on hardness w.r.t. uniform attackers)**.** *We note that although the lemma is stated w.r.t. non-uniform attackers, the proof directly applies also w.r.t. uniform hardness. This follows from the fact that nowhere in the reduction do we use any extra non-uniform advice—the reduction is completely black box—and we explicitly deal with randomized attackers.*

4 OWFs from Mild Avg-case Hardness of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$

We here show how to construct a OWF assuming mild average-case* hardness of $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ for some $\gamma > 1$. Our construction is essentially identical to the OWF construction in [LP20].

[11] That is, that the optimal distinguisher is $T = \{\omega : \Pr[Z_1 = \omega] > \Pr[Z_2 = \omega]\}$.

[LP20] bases the weak one-wayness of this construction on the assumption that K^t is mildly HoA to compute. We observe that a similar proof can be used to show that the same construction is a weak OWF assuming $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA* for any threshold s—the key difference here is that we need to show how to use a OWF inverter to not only compute K^t over random instances, but also conditioned on YES/NO-instances.

Lemma 4.1 (OWFs from $\mathsf{MK}^t\mathsf{P}$)**.** *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exist functions $T \in \mathcal{F}, s(n) > 0$, and polynomial $t(n) > 0$ such that $\mathsf{MK}^t\mathsf{P}[s(n)]$ is mildly HoA* for $T(n)$-time heuristics. Then, there exists a weakly $T'(n)$-one-way function for some function $T' \in \mathcal{F}$.*

Proof: Consider the function $f : \{0,1\}^{n+\lceil \log(n) \rceil} \to \{0,1\}^*$, which given an input $\ell || \Pi'$ where $|\ell| = \lceil \log(n) \rceil$ and $|\Pi'| = n$, outputs

$$\ell || U(\Pi(1), 1^{t(\ell)}) || U(\Pi(2), 1^{t(\ell)}) || \ldots || U(\Pi(n-1), 1^{t(\ell)}) || U(\Pi(n), 1^{t(\ell)})$$

where Π is the ℓ-bit prefix of Π'. Note that U only has polynomial overhead, so f can be computed in time $d(n)$ (for some increasing polynomial $d(\cdot)$). This function 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 T_1-one-way (over the restricted input lengths, for some function $T_1 \in \mathcal{F}$), then f' will be weakly T'-one-way (over all input lengths, for some function $T' \in \mathcal{F}$): $f'(x')$ simply truncates its input x' (as little as possible) so that the (truncated) input x now becomes of length $n' = n + \lceil \log(n) \rceil$ for some n and outputs $f(x)$. (We can pick any function $T' \in \mathcal{F}$ such that $T'(|x'|) \leq T_1(|x|)$, guaranteed to exist by Fact 2.11.)

Since $\mathsf{MK}^t\mathsf{P}[s(n)]$ is mildly HoA*, let $p(n)$ be the (monotonically increasing) polynomial such that $\mathsf{MK}^t\mathsf{P}[s(n)]$ is $\frac{1}{p(\cdot)}$-HoA*. Let $n'(n) = n + \lceil \log n \rceil$ be the input length of f. Let q be a monotonically increasing polynomial such that $q(n') = 2^{c+2} n' p(n')^2$, and let $T_1 \in \mathcal{F}$ be a function such that $T_1(n') \leq T(n'/2)/n' - d(n')$ (which is guaranteed to exist by Fact 2.11). We will show that f is $(T_1(n'), 1 - \frac{1}{q(n')})$-one way (on input lengths where f is well defined).

Assume for contradiction that f is not $(T_1(n'), 1 - \frac{1}{q(n')})$-one-way. Then, there exists an attacker $\mathcal{A}$ with running time bounded by $T_1(n')$ that inverts f for infinitely many n and $n'(n)$. We will construct a $T(n)$-time heuristic for $\mathsf{MK}^t\mathsf{P}[s(n)]$ with success probability (conditioned on YES and NO instances) at least $1 - \frac{1}{p(n)}$ using $\mathcal{A}$ (which breaks the $\frac{1}{p(\cdot)}$-HoA* property of $\mathsf{MK}^t\mathsf{P}[s(n)]$ since $1 - \frac{1}{p(n)} \geq 1 - \frac{1}{p(n^*)}$ where $n^* = \log D(n) \leq n$ and $D(n)$ is the density of $\mathsf{MK}^t\mathsf{P}[s(n)]$). Fix some n such that $\mathcal{A}$ inverts f with probability at least $1 - \frac{1}{q(n')}$. Our heuristic algorithm $\mathcal{H}(z)$, on input $z \in \{0,1\}^n$, runs $\mathcal{A}(1^{n'}, i||z)$ for every $i \in [n]$ where i is represented as a $\log(n)$ bit string, and outputs 1 if and only if the length of the shortest description Π output by $\mathcal{A}$, which produces each bit in the string z within $t(|\Pi|)$ steps, is at most $s(n)$.

We first analyze the running time of $\mathcal{H}$. $\mathcal{H}$ invokes $\mathcal{A}(1^{n'}, \cdot)$ (of running time $T_1(n')$) for n times, and invokes f (of running time $d(n)$) for n times to check whether the description output by $\mathcal{A}$ indeed produces y, so it runs in time $n(T_1(n') + d(n)) \leq n(T(n'/2)/n' - d(n') + d(n)) \leq n(T(n)/n - d(n) + d(n)) \leq T(n)$ (since $n'/2 \leq n \leq n'$).

We then analyze the correctness of $\mathcal{H}$. Our goal is to show that $\mathcal{H}(x)$ decides $\mathsf{MK}^t\mathsf{P}[s(n)]$ with probability at least $1 - \frac{1}{p(n)}$ conditioned on both $x \in \mathsf{MK}^t\mathsf{P}[s(n)]$ and $x \notin \mathsf{MK}^t\mathsf{P}[s(n)]$. If f is not $\frac{1}{q(n')}$-weak $T_1(n')$-one-way, then the inverter $\mathcal{A}$ will be able to invert f with probability at least $1 - \frac{1}{q(n')} \geq 1 - \frac{1}{q(n)}$ (since $n' \geq n$ and q is monotonically increasing). 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 $1 - \frac{2p(n)}{q(n)}$. When we invoke $\mathcal{A}$ in our heuristic $\mathcal{H}$, we always fix the randomness used in $\mathcal{A}$ to the same string, and let $\mathcal{H}_r$ denote the heuristic when the randomness is r.

We first show that with probability at most $\frac{1}{p(n)}$, $\mathcal{H}_r(z)$ outputs 0 when the input $z \in \{0,1\}^n$ is a random YES instance of $\mathsf{MK}^t\mathsf{P}[s(n)]$. Let S be the set of YES instances in $\mathsf{MK}^t\mathsf{P}[s(n)]$ on which $\mathcal{H}_r$ outputs 0. By Fact 2.1, there are at least $2^{\lfloor s(n) \rfloor - c}$ YES instances in $\mathsf{MK}^t\mathsf{P}[s(n)]$ (restricted to length n). Thus, $\mathcal{H}_r(z)$ outputs 0 with probability

$$fail_r \leq \frac{|S|}{2^{\lfloor s(n) \rfloor - c}}$$

conditioned on z being a YES-instance of $\mathsf{MK}^t\mathsf{P}[s(n)]$. Consider any string $z \in S$ and let $w = K^t(z) \leq \lfloor s(n) \rfloor$ (as z is a YES-instance of $\mathsf{MK}^t\mathsf{P}[s(n)]$). Since there exists a machine Π of length w that produces each bit of z in $t(w)$ steps, that is, for all $i \in [n]$, $\Pi(i)$ outputs z_i within $t(w)$ steps. Thus, there must exist a pre-image of f, denoted by x, such that $f(x) = (w||z)$. Since $\mathcal{H}_r(z)$ outputs 0, $\mathcal{A}_r$ must fail to invert $(w||z)$. But, since $w \leq \lfloor s(n) \rfloor$, the output $(w||z)$ is sampled with probability

$$\frac{1}{n} \cdot \frac{1}{2^{|w|}} \geq \frac{1}{n 2^{\lfloor s(n) \rfloor}}$$

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

$$|S| \cdot \frac{1}{n 2^{\lfloor s(n) \rfloor}} \geq fail_r \cdot \frac{1}{2^c n}$$

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

$$fail_r \leq \frac{2^{c+1} n p(n)}{q(n)}$$

By a union bound, we have that $\mathcal{H}$ (using a uniform random tape r) outputs 0 with probability at most

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

conditioned on z being a YES instance of $\mathsf{MK}^t\mathsf{P}[s(n)]$.

We then show that on input of a random NO instance of $\mathsf{MK}^t\mathsf{P}[s(n)]$, $\mathcal{H}$ will output 0 with probability 1, which (combining with the fact that $\mathcal{H}$ also succeeds conditioned on YES instances) shows that $\mathcal{H}$ is a heuristic for $\mathsf{MK}^t\mathsf{P}[s(n)]$ with success probability at least $1 - \frac{1}{p(\cdot)}$ (conditioned on both) and we reach a contradiction. Note that on a n-bit string z, if $\mathcal{H}$ outputs 1, there must exist a machine Π with description length $\leq s(n)$ such that Π produces each bit of z within $t(|\Pi|)$ steps, so $K^t(z) \leq s(n)$. Thus, $\mathcal{H}$ *never* outputs 1 when the input z satisfies $K^t(z) > s(n)$. ■

We prove the following simple corollary of the above lemma:

Corollary 4.1. *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exist a function $T \in \mathcal{F}$, a polynomial $t(n) > 0$, and a constant $\gamma > 1$ such that $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* for $T(n)$-time heuristics. Then, there exists a function $T' \in \mathcal{F}$ and a T'-one-way function.*

Proof: By Lemma 4.1, there exists a function $T_1 \in \mathcal{F}$ and a weakly T_1-one-way function. Then by Lemma 2.4 and Fact 2.11, the corollary follows. ■

Remark 4.2 (A note on hardness w.r.t. uniform attackers)**.** *We note that although the theorem and corollary are stated w.r.t. non-uniform attackers, the proof directly applies also w.r.t. uniform hardness; as before, the reduction is completely black-box. In more detail, if we assme that $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is average-case hard with respect to uniform $T(n)$-time attackers, we will get a OWF that is secure w.r.t. uniform T'-time attackers.*

5 Avg-case* Hardness of $\mathsf{MK}^t\mathsf{P}[s+O(1)]$ from OWFs

In this section, we establish the average-case* hardness of $\mathsf{MK}^t\mathsf{P}[s+O(1)]$ assuming the existence of OWFs. We first introduce the notion of a (conditionally-secure) entropy-preserving pseudorandom function (cond EP-PRF) and show that the existence of a cond EP-PRF implies the average-case* hardness of $\mathsf{MK}^t\mathsf{P}[s+O(1)]$. We then construct a cond EP-PRF from a (conditionally-secure) entropy-preserving pseudorandom generator (cond EP-PRG) and a standard PRF. Finally, we show that standard PRGs imply cond EP-PRGs, which completes our proof (since standard PRFs and standard PRGs exist assuming the existence of OWFs [HILL99, GGM84]).

5.1 Avg-case* Hardness of $\mathsf{MK}^t\mathsf{P}[s+O(1)]$ from Cond EP-PRF

We start by defining the notion of a *conditionally-secure entropy-preserving pseudorandom function.*

Definition 5.1. *An efficiently computable function* $f : \{0,1\}^n \times \{0,1\}^{k(n)} \to \{0,1\}$ *is a* $(T(\cdot), \varepsilon(\cdot))$-conditionally-secure entropy-preserving pseudorandom function ((T,ε)-cond EP-PRF) *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):** *For every probabilistic non-uniform* $T(n)$*-time attacker* $\mathcal{A}$ *and sufficiently large* $n \in \mathbb{N}$,
$$|\Pr[s \leftarrow \{0,1\}^n; \mathcal{A}^{f(s,\cdot)}(1^n) = 1 \mid E_n] - \Pr[f' \leftarrow \mathcal{F}; \mathcal{A}^{f'(\cdot)}(1^n) = 1]| < \varepsilon(n),$$
where $\mathcal{F} = \{f' : \{0,1\}^{k(n)} \to \{0,1\}\}$*;*
- **(entropy-preserving):** *For all sufficiently large* $n \in \mathbb{N}$, $H(\mathsf{tt}_n(f(\mathcal{U}_n \mid E_n, \cdot))) \geq n - \alpha \log n$*.*

In other word, a cond EP-PRF f is a PRF only when being conditioned on some event E_n, but even if being conditioned, the first n-bit of the truth table still contains high entropy (where the probability is taken over the random choice of the seed).

We say that $f : \{0,1\}^n \times \{0,1\}^{k(n)} \to \{0,1\}$ has rate-1 efficiency if for all $n \in \mathbb{N}, x \in \{0,1\}^n, i \in \{0,1\}^{k(n)}$, $f(x,i)$ runs in $n + O(n^\varepsilon)$ time for some constant $\varepsilon < 1$.

The following lemma shows that $\mathsf{MK}^t\mathsf{P}[s]$ is hard-on-average* assuming the existence of a sufficiently hard (rate-1 efficient) cond EP-PRF.

Lemma 5.1. *Let* $\mathcal{F}$ *be a nice class of super-polynomial functions, and* $\delta > 1$ *be some constant. Assume that there exist functions* $T_1, T_2 \in \mathcal{F}$ *such that* $T_1(n) \geq T_2(n)^\delta$ *and a rate-1 efficient* $(T_1(n), \frac{1}{n^2})$*-cond-EP-PRF* $f : \{0,1\}^n \times [T_2(n)] \to \{0,1\}$*. Then, there exists a constant* $\tau \in \mathbb{N}$ *such that for every constant* $\varepsilon > 0, 0 < \delta' < \delta$*, every polynomial* $t(m) \geq (1+\varepsilon)m$*,* $\mathsf{MK}^t\mathsf{P}[T_2^{-1}(m) + \tau]$ *is HoA* for* $m^{\delta'}$*-time heuristics.*

Proof: Let τ' be a constant such that the function f can be implemented by a Turing machine of description length τ'. Let $\tau = \tau' + 1$. Let $p(\cdot)$ be a polynomial such that $p(n) = 2^{\tau+3}(n+c)^{\alpha+4}$ where α is the entropy-loss constant of f. Consider any $\varepsilon > 0$ and any polynomial $t(n) \geq (1+\varepsilon)n$.

We assume for contradiction that there exists a $m^{\delta'}$-time non-uniform heuristic algorithm $\mathcal{H}$ for $\mathsf{MK}^t\mathsf{P}[T_2^{-1}(m) + \tau]$ with success probability at least $1 - \frac{1}{p'(m^*)}$ on infinitely many m, for some polynomial p' (where $m^* = \log D(m)$ and $D(m)$ is the density of $\mathsf{MK}^t\mathsf{P}[T_2^{-1}(m) + \tau]$). By Fact 2.1, $\lfloor T_2^{-1}(m) \rfloor + \tau - c \leq m^* \leq \lfloor T_2^{-1}(m) \rfloor + \tau + 1$, and it follows that there exists a polynomial $p(\cdot)$ such that $p'(m^*) \leq p(\lceil T_2^{-1}(m) \rceil)$ (for all sufficiently large m). Thus, $\mathcal{H}$ solves $\mathsf{MK}^t\mathsf{P}[T_2^{-1}(m) + \tau]$ with probability at least $1 - \frac{1}{p'(m^*)} \geq 1 - \frac{1}{p(\lceil T_2^{-1}(m) \rceil)}$ on infinitely many m (conditioned on YES/NO instances). We show that we can use $\mathcal{H}$ to break the pseudorandomness of f.

Consider some sufficiently large m where $\mathcal{H}$ succeeds. Let

$$n = \lceil T_2^{-1}(m) \rceil$$

be the smallest integer such that $T_2(n) \geq m$.

Note that f is a function that given a seed of length n, maps an integer $\in [T_2(n)]$ to either '0' or '1'. For any fixed seed $x \in \{0,1\}^n$, let $\mathsf{tt}_m(f(x,\cdot))$ denote the first m bits of the truth table of $f(x,\cdot)$. Note that for any $x \in \{0,1\}^n$, $\mathsf{tt}_m(f(x,\cdot))$ has low K^t-complexity (with probability 1):

$$K^t(\mathsf{tt}_m(f(x,\cdot))) \leq n + \tau'$$

since a Turing machine that contains the code of f (of length τ') and the seed x (of length n) can output each bit i on the truth table in $t(n)$ time (since f is rate-1 efficient). However, a random string of length m has high K^t-complexity with high probability:

$$\Pr_{y \in \{0,1\}^m}[K^t(y) > n + \tau] \geq 1 - \frac{1}{2^{m-n-\tau-1}},$$

since there are at most $2^{n+\tau+1}$ Turing machines with description length smaller than $n+\tau$, and each of them can produce at most a single truth table of length m.

With the above observations, we build an attacker $\mathcal{A}(1^n)$ that distinguishes between $f(x,\cdot)$ (for a random seed x) and a random function. On input length n, let $m(n)$ be an integer such that $T_2(n-1) < m(n) \leq T_2(n)$ and the heuristic $\mathcal{H}$ succeeds on input length $m(n)$, and let the attacker $\mathcal{A}(1^n)$ receive $m = m(n)$ as a non-uniform advice string. (If there's no such m, the attacker $\mathcal{A}$ just aborts. Since $\mathcal{H}$ succeeds on infinitely many m, there will be infinitely many n for which a "good" m exists.) Note that whenever $m(n)$ is defined, it holds that $n = \lceil T_2^{-1}(m(n)) \rceil$ (as T_2, by assumption, is monotonically increasing). Given black-box access to some function $f : [T_2(n)] \to \{0,1\}$, our distinguisher $\mathcal{A}^f(1^n)$ first queries f on every input $i \in [m]$ and gets the first m bits of its truth table, $\mathsf{tt}_m(f)$. Then, the distinguisher feeds $\mathsf{tt}_m(f)$ to $\mathcal{H}$ and outputs 0 if $\mathcal{H}(\mathsf{tt}_m(f))$ returns 0. Note that $\mathcal{A}$ needs to query each of the first m bits on the truth table of f (which takes $O(m(n+n^{\varepsilon}))$ time), so the running time of $\mathcal{A}$ is at most $O(\mathsf{poly}(n)T_2(n)^{\delta'} + m(n+n^{\varepsilon}) + (m)^{\delta'}) < T_2(n)^{\delta} \leq T_1(n)$ (since $\delta > 1$ and $\delta > \delta'$ and the function $T_2(\cdot)$ is super-polynomial). The following two claims will conclude that $\mathcal{A}$ distinguish the cond EP-PRF and a random function with probability at least $\frac{1}{n^2}$.

Claim 1. *$\mathcal{A}^{f_{\mathsf{r}}}(1^n)$ will output 0 with probability at least $1 - \frac{1}{2p(n)}$, where f_{r} is uniformly sampled from $\mathcal{F} = \{f_{\mathsf{r}} : [T_2(n)] \to \{0,1\}\}$.*

Proof: Recall that for a random f_{r}, the probability that $K^t(\mathsf{tt}_m(f_{\mathsf{r}})) > n + \tau$ is at least $1 - \frac{1}{2^{m-n-\tau-1}}$. Conditioned on $K^t(\mathsf{tt}_m(f_{\mathsf{r}})) > n + \tau = \lceil T_2^{-1}(m) \rceil + \tau$, $\mathsf{tt}_m(f_{\mathsf{r}})$ is a random NO instances of $\mathsf{MK^tP}[T_2^{-1}(m) + \tau]$ and $\mathcal{H}$ will output 0 with probability at least $1 - \frac{1}{p(\lceil T_2^{-1}(m) \rceil)} = 1 - \frac{1}{p(n)}$.

$$\begin{aligned}
&\Pr[f_{\mathsf{r}} \leftarrow \mathcal{F}; \mathcal{A}^{f_{\mathsf{r}}}(1^n) = 0] \\
\geq &\Pr[f_{\mathsf{r}} \leftarrow \mathcal{F}; K^t(\mathsf{tt}_m(f_{\mathsf{r}})) > n + \tau \wedge \mathcal{H}(\mathsf{tt}_m(f_{\mathsf{r}})) = 0] \\
= &\Pr[f_{\mathsf{r}} \leftarrow \mathcal{F}; K^t(\mathsf{tt}_m(f_{\mathsf{r}})) > n + \tau] \cdot \Pr[f_{\mathsf{r}} \leftarrow \mathcal{F}; \mathcal{H}(\mathsf{tt}_m(f_{\mathsf{r}})) = 0 \mid K^t(\mathsf{tt}_m(f_{\mathsf{r}})) > n + \tau] \\
\geq &(1 - \frac{1}{2^{m-n-\tau-1}})(1 - \frac{1}{p(n)}) \\
\geq &1 - \frac{2}{p(n)}
\end{aligned}$$

■

Claim 2. *$\mathcal{A}^{f(\mathcal{U}_n|E_n,\cdot)}(1^n)$ will output 0 with probability at most $1-\frac{1}{n}+\frac{2}{n^2}$.*

Proof: By the assumption that $\mathcal{H}$ solves $\mathsf{MK^tP}[T_2^{-1}(m)+\tau]$ on average conditioned on YES instances, given a random string $z \in \{0,1\}^m$ such that $K^t(z) \leq \lfloor T_2^{-1}(m)\rfloor+\tau$ (i.e., an YES instance), $\mathcal{H}$ outputs 0 (i.e., fails on z) with probability at most $\frac{1}{p(\lceil T_2^{-1}(m)\rceil)} = \frac{1}{p(n)}$. By an averaging argument, for at least a $1-\frac{1}{n^2}$ fraction of random tapes r for $\mathcal{H}$, the deterministic machine $\mathcal{H}_r$ outputs 0 with probability at most $\frac{n^2}{p(n)}$ when the input is sampled uniformly among YES instances. Fix some good random tape r such that $\mathcal{H}_r$ succeeds with probability at least $1-\frac{n^2}{p(n)}$ conditioning on input that has K^t-complexity at most $\lfloor T_2^{-1}(m)\rfloor+\tau$.

We then analyze the probability that $\mathcal{A}_r^{f(\mathcal{U}_n|E_n,\cdot),m}$ outputs 0. (When we invoke $\mathcal{H}$ in $\mathcal{A}_r^m$, we fix the randomness used by $\mathcal{H}$ to be r. Recall that r is the good random tape we fixed.) Assume for contradiction that $\mathcal{A}_r^m$ outputs 0 with probability at least

$$1-\frac{1}{n}+\frac{2^{\tau+3}n^{\alpha+2}}{p(n)}$$

when the seed of f is sampled conditioning on E_n. Recall that (1) the entropy of $\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))$ is at least $\mathsf{tt}_n(f(\mathcal{U}_n \mid E_n,\cdot)) \geq n-\alpha\log n$ and (2) the quantity $-\log\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))=z]$ is upper bounded by n for all $z \in \mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))$. By an averaging argument, with probability at least $\frac{1}{n}$, a random $z \in \mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))$ will satisfy

$$-\log\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))=z] \geq (n-\alpha\log n)-1.$$

We refer to a truth table z satisfying the above condition as being "good" and other z's as being "bad". Let $Z=\{z \in \mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) : \mathcal{A}_r^{\mathsf{fn}(z),m}(1^n)=0 \wedge z \text{ is good}\}$, and let $Z'=\{z \in \mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) : \mathcal{A}_r^{\mathsf{fn}(z),m}(1^n)=0 \wedge z \text{ is bad}\}$. Since

$$\Pr[\mathcal{A}^{f(\mathcal{U}_n|E_n,\cdot),m}(1^n)=0]=\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) \in Z]+\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) \in Z'],$$

and $\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) \in Z']$ is at most the probability that $\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))$ is "bad" (which is at most $1-\frac{1}{n}$ by the above argument), we have that

$$\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) \in Z] \geq \left(1-\frac{1}{n}+\frac{2^{\tau+3}n^{\alpha+2}}{p(n)}\right)-\left(1-\frac{1}{n}\right)=\frac{2^{\tau+3}n^{\alpha+2}}{p(n)}.$$

Furthermore, since for every $z \in Z$, $\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot))=z] \leq 2^{-n+\alpha\log n+1}$, we also have

$$\Pr[\mathsf{tt}_m(f(\mathcal{U}_n \mid E_n,\cdot)) \in Z] \leq |Z|2^{-n+\alpha\log n+1}.$$

Thus,

$$|Z| \geq \frac{2^{\tau+3}n^{\alpha+2}\cdot 2^{n-\alpha\log n-1}}{p(n)}.$$

However, for any $x \in \{0,1\}^n$, $K^t(\mathsf{tt}_m(f(x,\cdot))) \leq n+\tau' = \lceil T_2^{-1}(m)\rceil+\tau' \leq \lfloor T_2^{-1}(m)\rfloor+\tau'+1 = \lfloor T_2^{-1}(m)\rfloor+\tau$, so $\mathsf{tt}_m(f(x,\cdot))$ is also a YES instance of $\mathsf{MK^tP}[T_2^{-1}(m)+\tau]$. By Fact 2.1, there are at most $2^{\lfloor T_2^{-1}(m)\rfloor+\tau+1} \leq 2^{n+\tau+1}$ strings that are YES instances of $\mathsf{MK^tP}[T_2^{-1}(m)+\tau]$. It follows that $\mathcal{H}_r$ fails on random YES instances with probability at least

$$\frac{|Z|}{2^{n+\tau+1}} \geq \frac{2^{\tau+3}n^{\alpha+2}\cdot 2^{n-\alpha\log n-1}}{p(n)\cdot 2^{n+\tau+1}}=\frac{2n^2}{p(n)},$$

which contradicts to the fact that $\mathcal{H}_r$ is a good heuristic conditioned on YES instances.

We conclude that for every good randomness r, $\mathcal{A}_r^m$ outputs 0 with probability at most $1 - \frac{1}{n} + \frac{2^{\tau+3}n^{\alpha+2}}{p(n)}$. Finally, by a union bound (and since a random tape is bad with probability $\leq \frac{1}{n^2}$), we have that the probability that $\mathcal{A}^{f(\mathcal{U}_n|E_n,\cdot),m}(1^n)$ (when using a uniform random tape) outputs 0 is at most

$$\frac{1}{n^2} + \left(1 - \frac{1}{n} + \frac{2^{\tau+3}n^{\alpha+2}}{p(n)}\right) \leq 1 - \frac{1}{n} + \frac{2}{n^2}$$

since $p(n) = 2^{\tau+3}n^{\alpha+4}$. ■

With the above two claims (and $p(n) > n^3$), we conclude that $\mathcal{A}^{m(n)}$ distinguishes $f(\mathcal{U}_n \mid E_n, \cdot)$ and a random function f' with probability at least

$$\left(1 - \frac{2}{p(n)}\right) - \left(1 - \frac{1}{n} + \frac{2}{n^2}\right) > \frac{1}{n^2}$$

for infinitely many n.

■

5.2 Cond EP-PRF from Cond EP-PRG

In this section, we build a cond EP-PRF from a cond EP-PRG (and a standard PRF). We first recall the notion of a *conditionally-secure entropy-preserving PRG.*

Definition 5.2. *An efficiently computable function $g : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$ where $0 < \xi < 1$ is a $(T(n), \varepsilon(n))$-*conditionally-secure entropy-preserving pseudorandom generator *((T, ε)-*cond EP-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 $\{\mathcal{U}_{n+n^\xi}\}_{n\in\mathbb{N}}$ are $(T(n), \varepsilon(n))$-indistinguishable;*
- **(entropy-preserving):** *For all sufficiently large $n \in \mathbb{N}$, $H([g(\mathcal{U}_n \mid E_n)]_n) \geq n - \alpha \log n$.*

We remark that the notion of a cond EP-PRG was proposed in [LP20]. However, the above definition of a cond EP-PRG is *stronger* than the definition in [LP20]: they only require a cond EP-PRG with logarithmic stretch but we require a cond EP-PRG with *sublinear stretch*, n^ξ. Note that the recursive composition of a cond-EP-PRG g with logarithmic stretch (i.e., $g'(\cdot) = g(g(\dots g(\cdot)\dots)))$ is *not* necessarily a cond EP-PRG with sublinear stretch since g is only conditionally-secure, and the probability that *on every iteration*, the good event E holds, may become tiny.

The following lemma shows that we can construct a rate-1 efficient cond-EP-PRF from a cond-EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$ and a standard PRF, by first applying $g(\cdot)$ and then applying the standard PRF on the last n^ξ bits. We can next use a padding trick to make our construction rate-1 efficient.

Lemma 5.2. *Consider some constant $c_1 \geq 1$, some negligible function μ and assume there exist a $(T_1(n), \frac{1}{n^{6c_1}})$-cond-EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$ and a $(T_1(n), \mu)$-PRF $h : \{0,1\}^n \times [T_2(n)] \to \{0,1\}$ such that g, h run in time $O(n^{c_1})$. Then, there exist a constant $0 < \theta < 1$ and a rate-1 efficient $(T_1(n^\theta), \frac{1}{n^2})$-cond-EP-PRF $f : \{0,1\}^n \times [T_2(n^\theta)] \to \{0,1\}$.*

Proof: Let $g_0(\cdot), g_1(\cdot)$ denote the n-bit prefix and the n^ξ-bit suffix of g respectively, i.e., $g(\cdot) = g_0(\cdot)||g_1(\cdot)$ and $|g_0(x)| = n, |g_1(x)| = n^\xi$ for all $x \in \{0,1\}^n$. Let $\theta = \xi/(2c_1)$. Roughly speaking, to construct a cond-EP-PRF, we first apply a cond-EP-PRG on the seed x. Then we leave the first part ($g_0(x)$) as it is to keep the entropy, and apply a (standard) PRF on the second part ($g_1(x)$). We will use a padding trick to make our construction rate-1 efficient. Now we proceed to a formal construction. Let $m = n^{2c_1}$. Consider the function $f : \{0,1\}^m \times [T_2(n^\xi)] \to \{0,1\}$ defined as the following:

$$f(x,i) = \begin{cases} g_0([x]_n)_i, & \text{if } i \le n \\ x_i, & \text{if } n < i \le m \\ h(g_1([x]_n), i), & \text{if } i > m \end{cases}$$

where $g_0([x]_n)_i$ (resp x_i) denotes the i-th bit on the string $g_0([x]_n)$ (resp x). In other words, on input a seed x of length m, f uses the first $n = m^{1/(2c_1)}$ bits as the input of $g(\cdot)$. For the rest of the bits in the input, f pastes them into the truth table directly. (This is where the padding trick is.) Then f outputs the first n bits of $g([x]_n)$ directly to keep the entropy, and f applies a PRF $h : \{0,1\}^{n^\xi} \times [T_2(n^\xi)] \to \{0,1\}$ on the rest n^ξ bits of $g([x]_n)$. Note that $T_2(n^\xi) = T_2(m^{\xi/(2c_1)}) = T_2(m^\theta)$. Thus, f indeed maps $\{0,1\}^m \times [T_2(m^\theta)]$ to $\{0,1\}$.

We first show that f is entropy-preserving. Let E_m denote the event $\{x \in \{0,1\}^m : [x]_n \in E'_n\}$ where E'_n is the event associated with g (over input length n). It is sufficient to show that the first m bits (in the truth table of f) contain high enough entropy, which is indeed the case since the first m bits are of the form $g_0(x_{\mathsf{pre}})||x_{\mathsf{suf}}$ where $x = x_{\mathsf{pre}}||x_{\mathsf{suf}}$, and note that $g_0(\cdot) = [g(\cdot)]_n$ is entropy preserving. Formally,

$$H([\mathsf{tt}(f(\mathcal{U}_m \mid E_m, \cdot))]_m) = H(g_0(\mathcal{U}_n \mid E'_n)) + m - n =$$
$$H([g(\mathcal{U}_n \mid E'_n)]_n) + m - n \ge m - \alpha\log(n) \ge m - \alpha\log m,$$

where α is the entropy-loss constant of g.

We then show that f is pseudorandom by a standard hybrid argument. For any $T_1(m^\theta) = T_1(n^\xi)$-time adversary $\mathcal{A}$ and all sufficiently large m, let Real denote the quantity $\Pr[x \leftarrow \{0,1\}^m; \mathcal{A}^{f(x,\cdot)}(1^m) = 1 \mid E_m]$, and let Ideal denote the quantity $\Pr[f' \leftarrow \mathcal{F}; \mathcal{A}^{f'(\cdot)}(1^m) = 1]$ where $\mathcal{F}$ is the family of random functions. Define

$$f''(x,y,i) = \begin{cases} x_i, & \text{if } i \le m \\ h(y,i), & \text{if } i > m \end{cases}$$

where $|x| = m, |y| = n$. Let Hybrid denote the quantity $\Pr[x \leftarrow \{0,1\}^m, y \leftarrow \{0,1\}^n; \mathcal{A}^{f''(x,y,\cdot)}(1^m) = 1]$. The following two claims show that $|\mathsf{Real} - \mathsf{Ideal}| < \frac{1}{m^2}$ and thus f satisfies the pseudorandom property.

Claim 3. $|\mathsf{Real} - \mathsf{Hybrid}| < \frac{1}{m^3}$.

Proof: This claim follows from the fact that $\{g(\mathcal{U}_n \mid E_n)\}_{n\in\mathbb{N}}$ and $\{\mathcal{U}_{n+n^\xi}\}_{n\in\mathbb{N}}$ are $(T_1(n), \frac{1}{n^{6c_1}})$-indistinguishable and note that by our choice of parameters, $\frac{1}{n^{6c_1}} = \frac{1}{(n^{2c_1})^3} = \frac{1}{m^3}$ and $\mathcal{A}$ runs in time $T_1(m^\theta) = T_1(n^\xi) \le T_1(n)$. ■

Claim 4. $|\mathsf{Hybrid} - \mathsf{Ideal}| < \frac{1}{m^3}$.

Proof: This claim follows immediately from the fact that h is a $(T_1(\ell), \mu)$-pseudorandom function (where ℓ is the input length of h) and note that by our choice of parameters, $\ell = n^\xi$. So $\mathcal{A}$ runs in time $T_1(m^\theta) = T_1(n^\xi) \le T_1(\ell)$, and $\mu(\ell) = \mu(n^\xi) = \mu(m^\theta) \le \frac{1}{m^3}$ (since m is sufficiently large). ■

Finally, we show that f has running time $m + O(\log m)$ which (together with the above two proofs) shows that f is a rate-1 efficient cond-EP-PRF. Recall that both g and h run in time $O(n^{c_1})$, but the running time of f depends on i: If $i \le n$, f runs in time $O(n^{c_1}) = O(\sqrt{m})$. If $n < i \le m$, f

will output the i-th in the seed, which takes time $m + O(\log m)$. If $i > m$, the running time of f is bounded by $O(n^{c_1}) + O(n^{\xi c_1}) = O(\sqrt{m})$. Thus, f runs in time $m + O(\log m)$. ■

The next corollary shows that we can construct a rate-1 efficient cond EP-PRF from a cond-EP-PRG and a T-hard OWF.

Corollary 5.3. *Let $\mathcal{F}$ be a nice class of super-polynomial functions and let $c_2 \geq 1$ be some constant. Assume that for every $\beta > 0$, there exist a $(T_3, \frac{1}{n^\beta})$-cond-EP-PRG with running time $O(n^{c_2})$ (for some function $T_3 \in \mathcal{F}$) and a T_3-hard OWF. Then, for any constant $\delta > 1$, there exist functions $T_1, T_2 \in \mathcal{F}$ such that $T_1(n) \geq T_2(n)^\delta$ and a rate-1 efficient $(T_1(n), \frac{1}{n^2})$-cond-EP-PRF $f : \{0,1\}^n \times [T_2(n)] \to \{0,1\}$.*

Proof: We first show that the existence of a T_3-hard OWF will imply that there exist a negligible function μ, a constant c_3, functions $T_4, T_5, T_6, T_7, T_8 \in \mathcal{F}$ such that $T_3 \geq T_4 \geq T_5 \geq T_6 \geq T_7 \geq T_8$, and a $(T_7(n), \mu)$-PRF $h : \{0,1\}^n \times [T_8(n)] \to \{0,1\}$ with running time $O(n^{c_3})$ by the following steps:

- There exists a length-preserving T_4-one-way function (by Lemma 2.5 and Fact 2.11).
- There exists a $(T_5, 1/T_5)$-PRG (by Theorem 2.9 and Fact 2.11).
- There exists a length-doubling $(T_6, 1/T_6)$-PRG (by Lemma 2.6 and Fact 2.11) with running time $r_g(n)$ (for some polynomial $r_g(\cdot)$).
- Let $T_7(n) = T_6(n)^{1/2}$ and $T_8(n) = T_6(n)^{1/(2\delta)}$. By Theorem 2.10, there exists a $(T_7(n), \frac{nT_8(n)}{T_6(n)})$-PRF $h : \{0,1\}^n \times [T_8(n)] \to \{0,1\}$ with running time $r_g(n) \cdot \log T_8(n) \leq r_g(n) \cdot n$ (since $T_8(n) \leq 2^n$). Let c_3 be the constant such that $O(n^{c_3}) \geq r_g(n) \cdot n$, and note that $\frac{nT_8(n)}{T_6(n)} \leq \frac{n}{T_6(n)^{1-1/(2\delta)}}$ which is a negligible function. Thus, there exists a negligible function μ such that h is a (T_7, μ)-PRF with running time bounded by $O(n^{c_3})$.

Let $c_1 = \max\{c_2, c_3\}$, and let $\beta = 6c_1$. Since $T_3 \geq T_7$, it follows that there exist a $(T_7, \frac{1}{n^{6c_1}})$-cond-EP-PRG g and a (T_7, μ)-PRF $h : \{0,1\}^n \times [T_8(n)] \to \{0,1\}$ such that both g and h run in $O(n^{c_1})$ time.

Thus, by Lemma 5.2, there exist a constant $0 < \theta < 1$ and a rate-1 efficient $(T_7(n^\theta), \frac{1}{n^2})$-cond-EP-PRF $f : \{0,1\}^n \times [T_8(n^\theta)] \to \{0,1\}$. Let $T_1(n) = T_7(n^\theta)$ and $T_2(n) = T_8(n^\theta)$. Note that $T_8(n)^\delta = (T_6(n)^{1/(2\delta)})^\delta = T_6(n)^{1/2} = T_7(n)$, so it holds that $T_2(n)^\delta \leq T_1(n)$. ■

5.3 Cond EP-PRG from OWFs

In this section, we show how to construct a cond EP-PRG with sublinear stretch from OWFs. We refer the reader to the introduction for an overview of this construction, and here directly jump into the formal proof.

Lemma 5.3. *Let $\mathcal{F}$ be a nice class of super-polynomial functions, and T be a function $\in \mathcal{F}$. For any polynomial-time computable function $g : \{0,1\}^n \to \{0,1\}^{2n}$, there exist a polynomial $t_0(\cdot)$ and constants c_0, ξ such that for every $\beta > 0$, if g is a $(T(n), \frac{1}{n^{\beta \cdot c_0}})$-PRG, then there exists a $(T'(m), \frac{1}{m^\beta})$-cond EP-PRG $g_\beta : \{0,1\}^m \to \{0,1\}^{m+m^\xi}$ with running time $t_0(m)$ for some function $T'(m) \in \mathcal{F}$ satisfying $T'(m) \leq T(m)$.*

Proof: By Lemma 2.9 and Lemma 2.8, there exist a 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 $n, k \leq n$, $\mathcal{H}_k^n = \{h'_\sigma : \sigma \in \{0,1\}^{n^c}\}$ is a universal hash family, where $h'_\sigma = [h_\sigma]_k$ and $h_\sigma(x) = H(x, \sigma)$. For all $\beta > 0$, let $\beta'(\beta) =$

$2c(\beta+2)$, $c_0 = 2c$. Consider the function $g_\beta : \{0,1\}^{\log n + n + n^c + (3n)^c} \to \{0,1\}^{n^c + (3n)^c + \frac{3n}{2}}$ defined as the followings:

$$g_\beta(i, x, \sigma_1, \sigma_2) = \sigma_1 \Big|\Big| \sigma_2 \Big|\Big| \left[h_{\sigma_2} \left(g(x) \Big|\Big| [h_{\sigma_1}(x)]_{i - \beta' \log n} \Big|\Big| 0^{n-i+\beta' \log n} \right) \right]_{\frac{3n}{2}}$$

where $|i| = \log n, |x| = n, |\sigma_1| = n^c, |\sigma_2| = (3n)^c$, the input length of $h_{\sigma_1}(\cdot)$ is n, and the input length of $h_{\sigma_2}(\cdot)$ is $3n$. Let $m = m(n) = \log n + n + n^c + (3n)^c$ denote the input length of g_β. Note that g_β expands m bits to $\frac{3n}{2} + n^c + (3n)^c = m + (\frac{n}{2} - \log n) = m + m^\xi$ bits for some constant ξ. Let $\{E_{m(n)}\}$ be a sequence of events where

$$E_{m(n)} = \{i, x, \sigma_1, \sigma_2 : i = D_g(g(x)), x \in \{0,1\}^n, \sigma_1 \in \{0,1\}^{n^c}, \sigma_2 \in \{0,1\}^{(3n)^c}\}$$

and the degeneracy of g on point $g(x)$ is defined as $D_g(g(x)) = \lfloor \log |\{x' \in \{0,1\}^n : g(x') = g(x)\}| \rfloor$. We note that β' is chosen such that $\frac{1}{n^{\beta'}} \le \frac{1}{m^{\beta+2}}$ and c_0 is a constant such that $\frac{1}{8m^\beta} > \frac{1}{n^{c_0 \beta}}$ (for sufficiently large n). The following two claims show that g_β satisfies both the entropy-preserving property and the pseudorandomness property (if the function g is a sufficiently hard PRG).

Claim 5. *$H([g_\beta(\mathcal{U}_m \mid E_m)]_m) \ge m - (2\beta' + 1)\log n - 2$.*

Proof: Let $b = (2\beta'+1)\log n$. We claim that $[g_\beta(\mathcal{U}_m \mid E_m)]_{m-b}$ is $\frac{3}{n^{\beta'}} \le \frac{1}{m^2}$-close to $\mathcal{U}_{m-b}$ in statistical distance. If this is true, by Lemma 2.7, $H([g_\beta(\mathcal{U}_m \mid E_m)]_{m-b}) \ge m - b - 2$. Thus,

$$H([g_\beta(\mathcal{U}_m \mid E_m)]_m) \ge H([g_\beta(\mathcal{U}_m \mid E_m)]_{m-b}) \ge m - b - 2 \ge m - (2\beta'+1)\log n - 2.$$

We turn to proving that $[g_\beta(\mathcal{U}_m \mid E_m)]_{m-b}$ is $\frac{3}{n^{\beta'}}$-close to $\mathcal{U}_{m-b}$ by a standard hybrid argument. Let X, R_1, R_2 be random variables uniformly distributed over $\{0,1\}^n$, $\{0,1\}^{n^c}$, and $\{0,1\}^{(3n)^c}$, respectively, and $I' = D_g(g(X)) - \beta' \log n$. Let

$$\mathsf{Real} = R_1 \Big|\Big| R_2 \Big|\Big| \left[h_{R_2} \left(g(X) \Big|\Big| [h_{R_1}(X)]_{I'} \Big|\Big| 0^{n-I'} \right) \right]_{n - 2\beta' \log n}$$

and

$$\mathsf{Hybrid} = R_1 \Big|\Big| R_2 \Big|\Big| \left[h_{R_2} \left(g(X) \Big|\Big| \mathcal{U}_{I'} \Big|\Big| 0^{n-I'} \right) \right]_{n - 2\beta' \log n}$$

Our proof proceeds as the following:

- Note that Real is identically distributed to $[g_\beta(D_g(g(X)), X, R_1, R_2)]_{m-b}$ (since $|R_1| + |R_2| + n - 2\beta' \log n = m - (2\beta'+1)\log n = m - b$).
- We first show that $\mathsf{SD}(\mathsf{Real}, \mathsf{Hybrid}) \le \frac{1}{n^{\beta'}}$. We define a "post-processing" function:

$$g_{\mathsf{post}}(\sigma_1, \sigma_2, z) = \sigma_1 || \sigma_2 || \, [h_{\sigma_2}(z)]_{n - 2\beta' \log n}$$

Note that we can view the distribution Real and the distribution Hybrid as

$$\mathsf{Real} = g_{\mathsf{post}}(R_1, R_2, g(X) || [h_{R_1}(X)]_{I'} || 0^{n-I'})$$

and

$$\mathsf{Hybrid} = g_{\mathsf{post}}(R_1, R_2, g(X) || \mathcal{U}_{I'} || 0^{n-I'})$$

Since the statistical distance between any two distributions after applying the same (post-processing) function is at most the distance between the two original distributions, it follows that $\mathsf{SD}(\mathsf{Real}, \mathsf{Hybrid})$ is at most

$$\begin{aligned}
&\mathsf{SD}(g_{\mathsf{post}}(R_1, R_2, g(X)||[h_{R_1}(X)]_{I'}||0^{n-I'}), g_{\mathsf{post}}(R_1, R_2, g(X)||\mathcal{U}_{I'}||0^{n-I'})\\
\leq&\mathsf{SD}(R_1||R_2||g(X)||[h_{R_1}(X)]_{I'}||0^{n-I'}, R_1||R_2||g(X)||\mathcal{U}_{I'}||0^{n-I'})\\
\leq&\mathsf{SD}(R_1||g(X)||[h_{R_1}(X)]_{I'}, R_1||g(X)||\mathcal{U}_{I'})\\
=&\mathsf{E}_{y\leftarrow g(\mathcal{U}_n)}[\mathsf{SD}(R_1||y||[h_{R_1}(X \mid g(X)=y)]_{I'}, R_1||y||\mathcal{U}_{I'})]\\
=&\mathsf{E}_{y\leftarrow g(\mathcal{U}_n)}[\mathsf{SD}(R_1||y||[h_{R_1}(X \mid g(X)=y)]_{D_g(y)-\beta'\log n}, R_1||y||\mathcal{U}_{D_g(y)-\beta'\log n})]\\
\leq&\mathsf{E}_{y\leftarrow g(\mathcal{U}_n)}\left[\frac{1}{n^{\beta'}}\right] = \frac{1}{n^{\beta'}}
\end{aligned}$$

which follows from the Leftover Hash Lemma (i.e., Lemma 2.10) since $H_\infty(X \mid g(X)=y) \geq D_g(y)$.

- We then show that $\mathsf{SD}(\mathsf{Hybrid}, R_1||R_2||\mathcal{U}_{n-2\beta'\log n}) \leq \frac{2}{n^{\beta'}}$ also by the Leftover Hash Lemma. Note that $H_\infty(g(X)||\mathcal{U}_{I'}||0^{n-I'}) = H_\infty(g(X)||\mathcal{U}_{I'})$ and $H_\infty(g(X)||\mathcal{U}_{I'})$ is at least

$$\begin{aligned}
&\min_{y\in g(\mathcal{U}_n), z\in\{0,1\}^{I'}} -\log\Pr[g(X)=y \wedge \mathcal{U}_{I'}=z]\\
=&\min_{y\in g(\mathcal{U}_n), z\in\{0,1\}^{D_g(y)-\beta'\log n}} -\log\Pr[g(X)=y \wedge \mathcal{U}_{D_g(y)-\beta'\log n}=z]\\
=&\min_{y\in g(\mathcal{U}_n), z\in\{0,1\}^{D_g(y)-\beta'\log n}} -\log(\Pr[g(X)=y] \times \Pr[\mathcal{U}_{D_g(y)-\beta'\log n}=z])\\
=&\min_{y\in g(\mathcal{U}_n)} -\log\Pr[g(X)=y] + D_g(y) - \beta'\log n\\
\geq&\min_{y\in g(\mathcal{U}_n)} -\log\frac{2^{D_g(y)+1}}{2^n} + D_g(y) - \beta'\log n\\
=&\min_{y\in g(\mathcal{U}_n)} n - D_g(y) - 1 + D_g(y) - \beta'\log n\\
=&n - 1 - \beta'\log n.
\end{aligned}$$

Thus, $H_\infty(g(X)||\mathcal{U}_{I'}||0^{n-I'}) \geq n - 1 - \beta'\log n$. And again by the Leftover Hash Lemma, it follows that Hybrid is $\frac{2}{n^{\beta'}}$-close to $R_1||R_2||\mathcal{U}_{n-2\beta'\log n}$.

- Finally, $R_1||R_2||\mathcal{U}_{n-2\beta'\log n}$ is identically distributed to $\mathcal{U}_{m-b}$, which concludes the proof.

■

Claim 6. *If g is a $(T(n), \frac{1}{n^{\beta\cdot c_0}})$-PRG, then the ensembles $\{g_\beta(\mathcal{U}_{m(n)} \mid E_{m(n)})\}_{n\in\mathbb{N}}$ and $\{\mathcal{U}_{m(n)+m(n)^\xi}\}_{n\in\mathbb{N}}$ are $(T'(m), \frac{1}{m^\beta})$-indistinguishable where $T'(m) = \frac{T(n)}{16n^2m^{2\beta}}$.*

Proof: For the sake of contradiction, we can assume without loss of generality that there exists a $T'(m)$-time distinguisher $\mathcal{A}$ such that

$$\Pr[\mathcal{A}(g_\beta(\mathcal{U}_m \mid E_m)) = 1] - \Pr[\mathcal{A}(\mathcal{U}_{m+m^\xi}) = 1] \geq \frac{1}{m^\beta}$$

for infinitely many n. Fix some n, m where $\mathcal{A}$ succeeds. Consider the following experiments:

1. $\mathsf{P}_1 = \Pr[\mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(g(x)||[h_{\sigma_1}(x)]_{i'}||0^{n-i'})]_{\frac{3n}{2}}) = 1]$ where $x \leftarrow \{0,1\}^n, \sigma_1 \leftarrow \{0,1\}^{n^c}, \sigma_2 \leftarrow \{0,1\}^{(3n)^c}, i' = D_g(g(x)) - \beta' \log n$;
2. $\mathsf{P}_2 = \Pr[\mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(g(x)||r_1||0^{n-i'})]_{\frac{3n}{2}}) = 1]$ where $x \leftarrow \{0,1\}^n, \sigma_1 \leftarrow \{0,1\}^{n^c}, \sigma_2 \leftarrow \{0,1\}^{(3n)^c}, i' = D_g(g(x)) - \beta' \log n, r_1 \leftarrow \{0,1\}^{i'}$;
3. $\mathsf{P}_3 = \Pr[\mathcal{A}(\sigma_1||\sigma_2||r_2) = 1]$ where $\sigma_1 \leftarrow \{0,1\}^{n^c}, \sigma_2 \leftarrow \{0,1\}^{(3n)^c}, r_2 \leftarrow \{0,1\}^{\frac{3n}{2}}$.

Observe that $\mathsf{P}_1 = \Pr[\mathcal{A}(g_\beta(\mathcal{U}_m \mid E_m)) = 1]$ and $\mathsf{P}_3 = \Pr[\mathcal{A}(\mathcal{U}_{m+m^\xi}) = 1]$, so $\mathsf{P}_1 - \mathsf{P}_3 \geq \frac{1}{m^\beta}$. Notice that in experiment 1, for any $y \in g(\mathcal{U}_n)$, the min-entropy of the distribution $\{x \leftarrow \{0,1\}^n \mid g(x) = y\}$ is at least $D_g(y) \geq i' + \beta' \log n$. Thus, by the Leftover Hash Lemma, the distribution

$$\{x \leftarrow \{0,1\}^n \mid g(x) = y, \sigma_1 \leftarrow \{0,1\}^{n^c} : [h_{\sigma_1}(x)]_{i'}||\sigma_1\}$$

is $\frac{1}{n^{\beta'}}$-close to the distribution

$$\{r_1 \leftarrow \{0,1\}^{i'}, \sigma_1 \leftarrow \{0,1\}^{n^c} : r_1||\sigma_1\}.$$

It follows that $\mathsf{P}_2 \geq \mathsf{P}_1 - \frac{1}{n^{\beta'}} \geq \mathsf{P}_1 - \frac{1}{4m^\beta} \geq \mathsf{P}_3 + \frac{3}{4m^\beta}$. Given this observation, we construct a distinguisher $\mathcal{A}'$ to break the PRG g.

On input $z \in \{0,1\}^{2n}$, $\mathcal{A}'$ enumerates all possible $j \in [n]$. For each $j \in [n]$, the following procedure is repeated for $16nm^{2\beta}$ times. In the k-th iteration, $\mathcal{A}'$ samples $r_1 \in \{0,1\}^j$, $\sigma_1 \in \{0,1\}^{n^c}$, and $\sigma_2 \in \{0,1\}^{(3n)^c}$, and computes

$$p_{z,j,k} = \mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(z||r_1||0^{n-j})]_{\frac{3n}{2}}).$$

Let $p_{z,j}$ denote the mean of $p_{z,j,k}$'s; that is,

$$p_{z,j} = \frac{1}{16nm^{2\beta}} \sum_{k=1}^{16nm^{2\beta}} p_{z,j,k}.$$

We can consider the value $p_{z,j}$ as an empirical estimation of

$$q_{z,j} = \Pr_{r_1,\sigma_1,\sigma_2}[\mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(z||r_1||0^{n-j})]_{\frac{3n}{2}}) = 1].$$

Let j^* be the index j such that $p_{z,j}$ is maximized; that is, $j^* = \arg\max_j p_{z,j}$. Finally, $\mathcal{A}'$ samples $r_1^* \in \{0,1\}^{j^*}$, $\sigma_1^* \in \{0,1\}^{n^c}$, $\sigma_2^* \in \{0,1\}^{(3n)^c}$, and returns

$$\mathcal{A}(\sigma_1^*||\sigma_2^*||[h_{\sigma_2^*}(z||r_1^*||0^{n-j^*})]_{\frac{3n}{2}}).$$

Note that $\mathcal{A}'$ runs in time $n \cdot 16nm^{2\beta} \cdot T'(m) \leq T(n)$.

We first claim that when the input z is a random string of length $2n$, $\Pr_z[\mathcal{A}'(z) = 1]$ is roughly at most P_3. Let Z be a random variable uniformly distributed over $\{0,1\}^{2n}$, J^* be a random variable describing $\mathcal{A}'(Z)$'s choice of the index, and $R_1^*, R_{\sigma_1^*}, R_{\sigma_2^*}$ be three random variables uniformly distributed over $\{0,1\}^{J^*}, \{0,1\}^{n^c}, \{0,1\}^{(3n)^c}$, respectively. It follows that

$$\Pr_{z \leftarrow \{0,1\}^{2n}}[\mathcal{A}'(z) = 1] = \Pr[\mathcal{A}(R_{\sigma_1^*}||R_{\sigma_2^*}||[h_{R_{\sigma_2^*}}(Z||R_1^*||0^{n-J^*})]_{\frac{3n}{2}}) = 1].$$

Notice that $H_\infty(Z||R_1^*||0^{n-J^*}) \geq H_\infty(Z) \geq 2n$. It follows by the Leftover Hash Lemma that the distribution

$$R_{\sigma_1^*}||R_{\sigma_2^*}||[h_{R_{\sigma_2^*}}(Z||R_1^*||0^{n-J^*})]_{\frac{3n}{2}}$$

is $2^{-n/2}$-close to the distribution $R_{\sigma_1^*}||R_{\sigma_2^*}||\mathcal{U}_{\frac{3n}{2}}$ (which is distributed uniformly) in statistical distance. Thus,

$$\Pr_{z\leftarrow\{0,1\}^{2n}}[\mathcal{A}'(z)=1] \leq \Pr[\mathcal{A}(R_{\sigma_1^*}||R_{\sigma_2^*}||\mathcal{U}_{\frac{3n}{2}})=1] + \frac{1}{2^{n/2}} \leq \mathsf{P}_3 + \frac{1}{2^{n/2}}.$$

We then show that on input $z \leftarrow g(\mathcal{U}_n)$, $\Pr[\mathcal{A}'(z)=1]$ is, roughly, at least P_2. Recall that in $\mathcal{A}'(z)$, $p_{z,j}$ is computed as an empirical estimation of $q_{z,j}$. By the Chernoff bound,

$$\Pr[|q_{z,j}-p_{z,j}| > \frac{1}{4m^{\beta}}] \leq 2e^{-\frac{1}{16m^{2\beta}}\cdot 16nm^{2\beta}} \leq 2e^{-n}.$$

Let E' be the event that for all $j \in [n]$, $|q_{z,j}-p_{z,j}| \leq \frac{1}{4m^{\beta}}$, which by a union bound happens with probability at least $1-2ne^{-n}$. Since j^* is the index such that p_{z,j^*} is maximized, $p_{z,j^*} \geq p_{z,i'}$ where $i' = D_g(z) - \beta' \log n$ is defined in experiment 2. Conditioned on the event E', $q_{z,j^*} \geq p_{z,j^*} - \frac{1}{4m^{\beta}} \geq p_{z,i'} - \frac{1}{4m^{\beta}} \geq q_{z,i'} - \frac{1}{2m^{\beta}}$. Finally, note that the probability that $\mathcal{A}'(z)$ outputs 1 is exactly q_{z,j^*}, and thus

$$\begin{aligned}
&\Pr[\mathcal{A}'(g(\mathcal{U}_n))=1]\\
=&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[\Pr_{j^*\leftarrow J^*, r_1\leftarrow\{0,1\}^{j^*},\sigma_1,\sigma_2}[\mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(z||r_1||0^{n-j^*})])=1]\right]\\
=&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[\Pr[E']\cdot\mathsf{E}_{j^*\leftarrow J^*}[q_{z,j^*}\mid E'] + \Pr[\neg E']\cdot\mathsf{E}_{j^*\in J^*}[q_{z,j^*}\mid\neg E']\right]\\
\geq&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[\Pr[E']\cdot\mathsf{E}_{j^*\leftarrow J^*}[q_{z,j^*}\mid E']\right]\\
\geq&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[\Pr[E']\cdot(q_{z,i'}-\frac{1}{2m^{\beta}})\right]\\
\geq&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[q_{z,i'}-\frac{1}{2m^{\beta}}-2ne^{-n}\right]\\
=&\mathsf{E}_{z\leftarrow g(\mathcal{U}_n)}\left[\Pr_{r_1\in\{0,1\}^{i'},\sigma_1,\sigma_2}[\mathcal{A}(\sigma_1||\sigma_2||[h_{\sigma_2}(z||r_1||0^{n-i'})])=1]\right]-\frac{1}{2m^{\beta}}-2ne^{-n}\\
=&\mathsf{P}_2-\frac{1}{2m^{\beta}}-2ne^{-n}.
\end{aligned}$$

Combining the above two proofs, we conclude that $\Pr[\mathcal{A}'(g(\mathcal{U}_n))=1] - \Pr[\mathcal{A}'(\mathcal{U}_{2n})=1] \geq \mathsf{P}_2 - \frac{1}{2m^{\beta}} - 2ne^{-n} - \mathsf{P}_3 - \frac{1}{2^{n/2}} \geq \frac{1}{8m^{\beta}} \geq \frac{1}{n^{c_0\cdot\beta}}$, which is a contradiction. ■

Note that in the above claim, if $T \in \mathcal{F}$, then there exists a function $T'' \in \mathcal{F}$ such that $T''(m) \leq T'(m) = \frac{T(n)}{16n^2m^{2\beta}}$ by Fact 2.11. Thus, the two distributions are $(T''(m), \frac{1}{m^{\beta}})$-indistinguishable.

Although we have shown that g_{β} satisfies both the entropy-preserving property and pseudorandomness property, g_{β} is only defined over some input lengths $m = m(n)$. We then specify the behavior of g_{β} over those undefined input lengths. On input a string x' of an arbitrary length, denoted by m', $g_{\beta}(x')$ finds a prefix x of x' as long as possible such that $|x|$ is of form $m(n) = \log n + n + n^c + (3n)^c$ for some n, rewrites $x' = x||y$, and outputs $g_{\beta}(x)||y$. It follows that g_{β} still keeps the the entropy-preserving property and pseudorandomness property, and there exists a constant ξ such that g_{β} expands m bits to at least $m + m^{\xi}$ bits for every m.

We finally show that there exists a polynomial $t_0(\cdot)$ such that for every $\beta > 0$, g_{β} runs in time $t_0(m)$ on inputs of length m. Note that the function g used in the construction can be assumed to have some fixed polynomial running time. And the hash function h_{σ_1} (resp h_{σ_2}) always take n^c bits (resp $(3n)^c$ bits) as input, so we can assume that both the hash functions run in a fixed polynomial time. So for any $\beta > 0$, the running time of g_{β} will always be upper-bounded by some polynomial $t_0(\cdot)$. ■

The following corollary shows that we can construct a cond EP-PRG from a T-hard one-way function.

Corollary 5.4. *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exists a function $T_1 \in \mathcal{F}$ such that T_1-one-way functions exist. Then, there exist a polynomial $t_0(\cdot)$ and a constant ξ such that for every $\beta > 0$, there exists a $(T_2(n), \frac{1}{n^\beta})$-cond EP-PRG $g_\beta : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$ with running time $t_0(m)$ for some function $T_2(n) \in \mathcal{F}$ satisfying $T_2(n) \leq T_1(n)$.*

Proof: Since there exists a T_1-one-way function for some function $T_1 \in \mathcal{F}$, the proof of Corollary 5.3 shows that there exists a function $T_3 \in \mathcal{F}$ such that there exists a length-doubling $(T_3, 1/T_3)$-PRG g. The corollary follows from Lemma 5.3 since for all constants $\beta, c_0 > 0$, $\frac{1}{T_3(n)} \leq \frac{1}{n^{\beta \cdot c_0}}$ for all sufficiently large n, g satisfies the requirement in the statement of the lemma. ■

6 The Main Theorem

In this section, we recall the statement of our main theorem and formally prove it (relying on all the earlier proved results).

Theorem 6.1 (restatement of Theorem 1.1)**.** *Let $\mathcal{F}$ be a nice class of super-polynomial functions, and $t(n)$ be a polynomial such that $t(n) \geq (1+\varepsilon)n, \varepsilon > 0$. The following are equivalent:*

(a) There exists a function $T \in \mathcal{F}$ such that (non-uniformly secure) $T(n)$-one-way functions exist.

(b) There exist an integer constant $\tau \geq 0$ and a function $T \in \mathcal{F}$ such that $\mathsf{MK}^t\mathsf{P}[T^{-1}(n) + \tau]$ is mildly HoA for sublinear-time non-uniform heuristics.*

(c) There exists a function $T \in \mathcal{F}$ such that for any integer $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA for $T(n)$-time non-uniform heuristics.*

At a high level, almost all earlier proved results are polynomial-time reductions (in the input length and the running-times), and the above theorem follows from the fact that such reductions "preserve" $\mathcal{F}$-hardness (as stated in section 2.8).

Proof: [Proof of Theorem 6.1]

(b) ⇒ (c): This follows directly from Lemma 3.1.

(c) ⇒ (a): This follows directly from Corollary 4.1.

(a) ⇒ (b): This follows from Corollary 5.4, Corollary 5.3, and Lemma 5.1. ■

Remark 6.1 (A note on hardness w.r.t. uniform attackers)**.** *We note that although the theorem is stated w.r.t. non-uniform attackers, the implications that (b) implies (c) implies (a) also hold w.r.t. uniform attackers; see Remarks 3.2 and 4.2.*

6.1 Corollaries

We here explicitly state some corollaries of Theorem 6.1 when considering specific function families $\mathcal{F}$.

Corollary 6.2 (Characterizing Subexponentiall-secure OWFs)**.** *Let $\varepsilon > 0$, and let $t(n)$ be a polynomial $t(n) \geq (1+\varepsilon)n$, the following are equivalent:*

(a) Subexponentially-secure (non-uniformly) one-way functions exist.

(b) *There exist constants $\beta > 1, \tau \geq 0$ such that $\mathsf{MK}^t\mathsf{P}[\log^{\beta} n + \tau]$ is mildly HoA* for sublinear-time non-uniform heuristics.*

(c) *There exists a constant $\varepsilon > 0$ such that for any integer $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* for $2^{n^{\varepsilon}}$-time non-uniform heuristics.*

Proof: Let $\mathcal{F}_{\mathsf{subexp}} = \{2^{cn^{\varepsilon}}\}_{c>0,0<\varepsilon<1}$ be the class of subexponential functions. Note that (1) for all $c > 0, 0 < \varepsilon < 1$, the function $f(n) = 2^{cn^{\varepsilon}}$ is time-constructible and for all $n > 1$, $f(n+1) - f(n) = 2^{c((n+1)^{\varepsilon} - n^{\varepsilon})} = \Omega(1)$, (2) for all $0 < \varepsilon_1, \varepsilon_2 < 1$, $f(n^{\varepsilon_1})^{\varepsilon_2} = 2^{c\varepsilon_2 n^{\varepsilon \cdot \varepsilon_1}} \in \mathcal{F}$, and (3) $f(n)$ is super-polynomial. Thus, $\mathcal{F}_{\mathsf{subexp}}$ is a nice class of super-polynomial functions. Let $\mathcal{F}^{-1}_{\mathsf{subexp}} = \{c \log^{\beta} n\}_{c>0,\beta>1}$. Notice that for all $c > 0, 0 < \varepsilon < 1$, the inverse function of $f(n) = 2^{cn^{\varepsilon}}$ is $f^{-1}(n) = \frac{1}{c^{\varepsilon^{-1}}} \log^{\varepsilon^{-1}} n$ since $f(f^{-1}(n)) = 2^{(c(\frac{1}{c^{\varepsilon^{-1}}} \log^{\varepsilon^{-1}} n)^{\varepsilon})} = 2^{(c(\frac{1}{c} \log n))} = 2^{\log n} = n$ and $f^{-1}(n) \in \mathcal{F}^{-1}_{\mathsf{subexp}}$. So, $\mathcal{F}^{-1}_{\mathsf{subexp}}$ is indeed the inverse class of $\mathcal{F}_{\mathsf{subexp}}$. Finally, the corollary follows from Theorem 6.1 with $\mathcal{F} = \mathcal{F}_{\mathsf{subexp}}$ (and the fact that for any $c > 0, 0 < \varepsilon < 1$, there exist constants $0 < \varepsilon_1, \varepsilon_2 < 1$ such that $2^{n^{\varepsilon_1}} \leq 2^{cn^{\varepsilon}} \leq 2^{n^{\varepsilon_2}}$, and for any $c > 0, \beta > 1$, there exist constants $\beta_1, \beta_2 > 0$ such that $\log^{\beta_1} n \leq c \log^{\beta} n \leq \log^{\beta_2} n$). ■

Corollary 6.3 (Characterizing Quasi polynomially-secure OWFs)**.** *Let $\varepsilon > 0$, and let $t(n)$ be a polynomial $t(n) \geq (1+\varepsilon)n$, the following are equivalent:*

(a) *Quasi polynomially-secure (non-uniformly) one-way functions exist.*

(b) *There exists constants $c > 0, \tau \geq 0$ such that $\mathsf{MK}^t\mathsf{P}[2^{c\sqrt{\log n}} + \tau]$ is mildly HoA* for sublinear-time non-uniform heuristics.*

(c) *There exists a constant $c > 0$ such that for any integer $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA* for $n^{c \log n}$-time non-uniform heuristics.*

Proof: Let $\mathcal{F}_{\mathsf{qpoly}} = \{n^{c \log n}\}_{c>0}$ be the class of quasi-polynomial functions. Note that (1) for all $c > 0$, the function $f(n) = 2^{c \log n}$ is time-constructible and for all $n > 1$, $f(n+1) - f(n) = \Omega(1)$, (2) for all $0 < \varepsilon_1, \varepsilon_2 < 1$, $f(n^{\varepsilon_1})^{\varepsilon_2} = (n^{\varepsilon_1})^{c\varepsilon_2 \log(n^{\varepsilon_1})} = n^{c\varepsilon_1^2 \varepsilon_2 \log n} \in \mathcal{F}$, and (3) $f(n)$ is super-polynomial. Thus, $\mathcal{F}_{\mathsf{qpoly}}$ is a nice class of super-polynomial functions. Let $\mathcal{F}^{-1}_{\mathsf{qpoly}} = \{2^{c\sqrt{\log n}}\}_{c>0}$. Notice that for all $c > 0$, the inverse function of $f(n) = n^{c \log n}$ is $f^{-1}(n) = 2^{\frac{1}{\sqrt{c}}\sqrt{\log n}}$ since $f(f^{-1}(n)) = (f^{-1}(n))^{c \log f^{-1}(n)} = 2^{c(\log f^{-1}(n))^2} = 2^{c(\frac{1}{\sqrt{c}}\sqrt{\log n})^2} = 2^{\log n} = n$ and $f^{-1}(n) \in \mathcal{F}^{-1}_{\mathsf{qpoly}}$. So $\mathcal{F}^{-1}_{\mathsf{qpoly}}$ is indeed the inverse class of $\mathcal{F}_{\mathsf{qpoly}}$. Finally, the corollary follows from Theorem 6.1 with $\mathcal{F} = \mathcal{F}_{\mathsf{qpoly}}$. ■

6.2 A Characterization of Polynomially-secure OWF

We finally note that our treatment can also be used to characterize "standard" polynomially-hard OWFs using sublinear hardness of $\mathsf{MK}^t\mathsf{P}[s]$. Although this formally does not follow as a corollary to Theorem 6.1, the same proof directly applies as all reductions used to show Theorem 6.1 are polynomial-time reductions (in the input length and in the running-times).

Theorem 6.4. *Let $\varepsilon > 0$, and let $t(n)$ be a polynomial $t(n) \geq (1+\varepsilon)n$, the following are equivalent:*

(a) *(Non-uniformly) OWFs exist.*

(b) *There exists a constant $\delta > 0$ such that for all $0 < \varepsilon < 1$, $\mathsf{MK}^t\mathsf{P}[n^{\varepsilon}]$ is mildly HoA* for n^{δ}-time non-uniform heuristics.*

(c) For any integer $\gamma > 1$, $\mathsf{MK}^t\mathsf{P}[n/\gamma]$ is mildly HoA for all polynomial-time non-uniform heuristics.*

We note, in contrast to the characterizations in Section 6.1, this characterization is less appealing in that we are not able to identify a single problem whose sublinear hardness characterizes OWFs; rather, OWFs are characterized through the sublinear hardness of $\mathsf{MK}^t\mathsf{P}[n^\varepsilon]$ for *every* $0 < \epsilon < 1$.

7 Unconditional Lowerbounds for $\mathsf{MK}^t\mathsf{P}[s(n)]$

As noted in Remark 6.1, to deduce T-hard OWFs where security holds w.r.t. *uniform* T-time probabilistic attackers (i.e., uniformly-secure OWFs), it suffices to assume sublinear time hardness of $\mathsf{MK}^t\mathsf{P}$ w.r.t. *uniform* sublinear-time attackers.

We now complement this result by establishing lower bounds that come surprisingly close to what is required to *unconditionally* deduce the existence of subexponentially-hard (uniformly-secure) OWFs (and thus that $\mathsf{NP} \notin \mathsf{BPTIME}(2^{n^\alpha})$ for some $\alpha > 0$). Our first lower bound demonstrates *worst-case hardness* of $\mathsf{MK}^t\mathsf{P}[s]$ with respect to sublinear uniform probabilistic algorithms, even for very small thresholds $s(\cdot)$ (recall that for Theorem 6.1, we require mild *average-case* hardness for the same problem).

Theorem 7.1. *For any function $t(n) \geq n$, for every $0 < \delta < 1$, $\omega(1) < s(n) < n - n^\delta - 2$, $\mathsf{MK}^t\mathsf{P}[s(n)] \notin \mathsf{BPTIME}(n^\delta)$.*

Proof: Let L denote the language $\mathsf{MK}^t\mathsf{P}[s(n)]$. Suppose for contradiction that there exists a probabilistic n^δ-time Turing machine $\mathcal{H}$ such that for every $x \in \{0,1\}^n$, $\Pr[\mathcal{H}(x) = L(x)] \geq \frac{2}{3}$. Consider some n-bit string $y \in \mathsf{MK}^t\mathsf{P}[s(n)]$, and it follows that $\Pr[\mathcal{H}(y) = 1] \geq \frac{2}{3}$. Note that the heuristic $\mathcal{H}$ is of running time n^δ and it can only access to the first n^δ bits of y. Let z be a string $\in \{0,1\}^n$ whose first n^δ bits are identical to the first n^δ bits of y. It's impossible for $\mathcal{H}$ to tell apart z from y, so $\mathcal{H}(y) = \mathcal{H}(z)$. The following claim shows that there exist a large number of such z's that have high enough K^t complexity.

Claim 7. *There are $2^{n-n^\delta-1}$ strings $z \in \{0,1\}^n$ such that $[z]_{n^\delta} = [y]_{n^\delta}$ and $K^t(z) > s(n)$.*

Proof: This claim follows from a counting argument. Let $Z' = \{z' \in \{0,1\}^n : [z']_{n^\delta} = [y]_{n^\delta}\}$ be a set of strings that have the same prefix with y. It follows that $|Z'| \geq 2^{n-n^\delta}$. By Fact 2.1, we know that there are at most $2^{s(n)+1}$ strings of length n that have K^t-complexity no more than $s(n)$. Let $Z = Z'/\mathsf{MK}^t\mathsf{P}[s(n)]$, and it follows that $|Z| \geq 2^{n-n^\delta} - 2^{s(n)+1} \geq 2^{n-n^\delta-1}$. ■

Thus, for every string $z \in Z$, $\Pr[\mathcal{H}(z) = 1] = \Pr[\mathcal{H}(y) = 1] \geq \frac{2}{3}$. However, $z \notin L$, which is a contradiction. ■

Our second lower bound demonstrates that when the threshold s is large, $s(n) = n - \log n$, then $\mathsf{MK}^t\mathsf{P}[s]$ is mildly HoA (and thus also mildly HoA*) with respect to not only sublinear uniform algorithms, but even for algorithms that run in time $t(n)/n^3$. This theorem extends a recent lower bound by Hirahara [Hir20] that establishes average-case hardness of $\mathsf{MK}^t\mathsf{P}[n-1]$ w.r.t. *errorless* deterministic heuristics where $t = n^{\omega(1)}$. As far as we know, our result is the first lower bound demonstrating *two-sided error* average-case hardness for time-bounded Kolmogorov complexity.

Theorem 7.2. *For any function t, any constant $\beta > 2$, $0 < \alpha \leq \beta - 2$, $\mathsf{MK}^t\mathsf{P}[n - \alpha \log n]$ is $\frac{1}{n^\beta}$-HoA* for deterministic uniform $t(n)/(n^{\alpha+1} \log^3 n)$-time heuristics.*

Proof: Consider some function t and any constants $\beta > 2$, $0 < \alpha \leq \beta - 2$. Let L denote the language $\mathsf{MK}^t\mathsf{P}[n - \alpha \log n]$. Let $m(n) = (\alpha + 1)\log n + 3\log\log n$, and assume for contradiction that there exists a

$$t(n)/2^{m(n)} = t(n)/(n^{\alpha+1}\log^3 n)$$

time *deterministic* heuristic $\mathcal{H}$ with success probability at least $1 - \frac{1}{n^\beta}$ for infinitely many n. Fix some n where $\mathcal{H}$ succeeds (i.e., its failure probability is at most $\frac{1}{n^\beta}$).

We start by showing that the space of n-bit strings can be covered by "balls" of $2^{m(n)}$ strings, where each ball is specified by an $n - m(n)$-bit prefix y, such that there exists many balls on which $\mathcal{H}$ perfectly decides the language. More precisely, given an $n - m(n)$-bit prefix y, let B_y denote the set of n-bit strings that have y as a prefix. We refer to a ball B_y as being "good" if $\mathcal{H}(x) = L(x)$ for all $x \in B_y$ (note that the definition of B_y being good relies on $\mathcal{H}$ being deterministic). We now have the following claim:

Claim 8. *For at least a fraction 1/2 of $n - m(n)$ bits strings y, we have B_y is good.*

Proof: By an averaging argument, for at most a $\frac{1}{2}$ fraction of $(n - m(n))$-bit strings y, the probability that $\mathcal{H}(x)$ fails over random strings $x \in B_y$ is at most $\frac{2}{n^\beta}$. Note, however that B_y contains $2^{m(n)} = n^{\alpha+1}\log^3 n < n^{\alpha+2}/2 \leq n^\beta/2$ strings, so $\mathcal{H}$ cannot fail on any single one of them. ■

We will next show that there must exist some good ball that contains a string z that is not in L (i.e., has high K^t complexity). Intuitively, this will yield a contradiction as we can compress this string z by specifying the ball (through its $(n - m(n))$-bit prefix y), and next searching of the string z by enumarating all string in B_y and using $\mathcal{H}$ to determine its K^t-complexity.

Claim 9. *There exists some good ball B_y such that B_y contains a string $x \notin L$.*

Proof: By Fact 2.1, only a $\frac{1}{n^\alpha} < \frac{1}{2}$ fraction of n-bit strings are in $\mathsf{MK}^t\mathsf{P}[n - \alpha \log n]$. Since by Claim 8, there are at most a $\frac{1}{2}$ fraction of strings contained in "bad" balls (as all balls are disjoint), there must exist at least one string $x \notin L$ that is contained in a good ball. ■

Fix a string y guaranteed to exists by Claim 9, and let $z \in \{0,1\}^n$ be the lexicographically smallest string in B_y such that $z \notin L$ and thus, $K^t(z) > n - \alpha \log n$. We contradict this by presenting a machine $\mathcal{A}_y$ with a short description that generates z. On input $i \in [n]$, $\mathcal{A}_y(i)$ tries all $a \in \{0,1\}^{m(n)}$, runs $\mathcal{H}(y||a)$, and outputs the i-th bit of the lexicographically smallest string $y||a$ such that $\mathcal{H}(y||a)$ returns 0. Since by assumption $y \in B_y$, we have that $\mathcal{H}(y||a) = L(y||a)$ and thus $\mathcal{A}_y$ generates z as desired.

Note that $\mathcal{A}_y$ can be described by $n \in \mathbb{N}, y \in \{0,1\}^{n-m(n)}$, and the code of $\mathcal{H}$ (which is constant); thus, the length of the description of $\mathcal{A}_y$ is

$$(\log n + 2\log\log n) + (n - m(n)) + O(1) \leq n - \alpha \log n.$$

Furthemore, the running time of $\mathcal{A}_y$ is at most $2^{m(n)} \times \frac{t(n)}{2^{m(n)}} = t(n)$, so we conclude that $K^t(z) \leq n - \alpha \log n$, which contradicts the fact that $z \notin L$. ■

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.

[AK10] Eric Allender and Michal Kouckỳ. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM (JACM)*, 57(3):1–36, 2010.

[All01] Eric Allender. When worlds collide: Derandomization, lower bounds, and kolmogorov complexity. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 1–15. Springer, 2001.

[BABB19] Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. The average-case complexity of counting cliques in erdos-rényi hypergraphs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1256–1280. IEEE, 2019.

[BRSV17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 483–496, 2017.

[BRSV18] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of work from worst-case assumptions. In *Annual International Cryptology Conference*, pages 789–819. Springer, 2018.

[CHO$^+$20] Lijie Chen, Shuichi Hirahara, Igor C Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[CJW19] Lijie Chen, Ce Jin, and R Ryan Williams. Hardness magnification for all sparse np languages. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1240–1255. IEEE, 2019.

[CMMW19] Lijie Chen, Dylan M McKay, Cody D Murray, and R Ryan Williams. Relations and equivalences between circuit lower bounds and karp-lipton theorems. In *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[CT19] Lijie Chen and Roei Tell. Bootstrapping results for threshold circuits "just beyond" known lower bounds. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 34–41, 2019.

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

[DLW20] Mina Dalirrooyfard, Andrea Lincoln, and Virginia Vassilevska Williams. New techniques for proving fine-grained average-case hardness. *arXiv preprint arXiv:2008.06591*, 2020.

[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.

[GR18] Oded Goldreich and Guy Rothblum. Counting t-cliques: Worst-case to average-case reductions and direct interactive proof systems. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 77–88. IEEE, 2018.

[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.

[Hir20] Shuichi Hirahara. Unexpected hardness results for kolmogorov complexity under uniform reductions. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1038–1051, 2020.

[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.

[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.

[LLW19] Rio LaVigne, Andrea Lincoln, and Virginia Vassilevska Williams. Public-key cryptography in the fine-grained setting. In *Annual International Cryptology Conference*, pages 605–635. Springer, 2019.

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, (052), 2020.

[LW13] Richard J Lipton and Ryan Williams. Amplifying circuit lower bounds against polynomial time, with applications. *computational complexity*, 22(2):311–343, 2013.

[MMW19] Dylan M McKay, Cody D Murray, and R Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1215–1225, 2019.

[MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2):102735, 2020.

[Oli19] Igor Carboni Oliveira. Randomness and intractability in kolmogorov complexity. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[OPS19] Igor Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. 2019.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 65–76. IEEE, 2018.

[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.

[Sri03] Aravind Srinivasan. On the approximability of clique and related maximization problems. *Journal of Computer and System Sciences*, 67(3):633–651, 2003.

[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.

[Yab59a] Sergey Yablonski. The algorithmic difficulties of synthesizing minimal switching circuits. *Problemy Kibernetiki*, 2(1):75–121, 1959.

[Yab59b] Sergey V Yablonski. On the impossibility of eliminating perebor in solving some problems of circuit theory. *Doklady Akademii Nauk SSSR*, 124(1):44–47, 1959.

[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.

A Proofs for some Lemmas

In this section, we provide formal proofs for Lemma 2.2, 2.3 and 2.4.

Proof of Lemma 2.2 We first claim that L must be a $D(n)$-dense language where $D(n) \geq 2^{n-\log^2 n}$. Suppose for contradiction that $D(n) \leq \frac{2^n}{\log^2 n}$. Then a heuristic that always outputs 0 will decide L with probability at least $1 - \frac{1}{2^{\log^2 n}}$, which is a contradiction.

We now show that L is mildly HoA* (for $T(n)$-time heuristics), which concludes the proof. Let $n^* = \log D(n)$; it follows that $n^* \geq n - \log^2 n \geq n/2$. Assume for contradiction that there exist a monotonically increasing polynomial $p(\cdot)$ and a heuristic $\mathcal{H}$ such that for infinitely many n, for all $\mu \in \{0,1\}$,

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

It follows that

$$\begin{aligned}
&\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}(x) = L(x)] \\
=&\Pr[x \in L(x)] \Pr[\mathcal{H}(x) = 1 \mid L(x) = 1] + \Pr[x \notin L(x)] \Pr[\mathcal{H}(x) = 0 \mid L(x) = 0] \\
\geq&1 - \frac{1}{p(n^*)} \\
\geq&1 - \frac{1}{p(n/2)}
\end{aligned}$$

which is a contradiction.

Proof of Lemma 2.3 Suppose for the sake of contradiction that there exist a $T(n)$-time errorless heuristic $\mathcal{H}'$ and a monotonically increasing polynomial $p(\cdot)$ such that for infinitely many n,

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}'(x) = \bot] \le \frac{1}{p(n)},$$

and

$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{H}'(x) \in \{L(x), \bot\}] = 1.$$

There exists a bit $\mu \in \{0,1\}$ such that for infinitely many n on which $\mathcal{H}'$ succeeds, $|\{x \in \{0,1\}^n : L(x) = \mu\}| \ge 2^{n-1}$. Fix some such n. Our heuristic $\mathcal{H}$, on input $x \in \{0,1\}^n$, outputs $\mathcal{H}'(x)$ if $\mathcal{H}'(x) \neq \bot$ and otherwise outputs $\mu \oplus 1$. It follows that

$$\Pr[\mathcal{H}(x) \neq \mu \mid L(x) = \mu] = \Pr[\mathcal{H}'(x) = \bot \mid L(x) = \mu] \le \frac{\Pr[\mathcal{H}'(x) = \bot]}{\Pr[L(x) = \mu]} \le \frac{2}{p(n)} \le \frac{2}{p(n^*)}$$

and

$$\Pr[\mathcal{H}(x) \neq \mu \oplus 1 \mid L(x) = \mu \oplus 1] = \Pr[\mathcal{H}'(x) = \mu \mid L(x) = \mu \oplus 1] = 0 \le \frac{2}{p(n^*)}$$

where $n^* = \log D(n)$ and $D(n)$ is the density of L. Thus, $\mathcal{H}'$ is a good "heuristic*" for L that succeeds with probability at least $1 - \frac{2}{p(n^*)}$ over infinitely many n, which is a contradiction.

Proof of Lemma 2.4 Let f be a $(T(n), 1-\frac{1}{q(n)})$-one-way function. Consider the dot product function f' defined as $f'(x_1, \dots, x_m) = (f(x_1), \dots, f(x_m))$ where $m = 2nq(n)$. Let us denote the input length of f' by n' where $n' = nm$. We claim that f' is a $(T'(n'))$-one-way where $T'(n') = \sqrt{\frac{T(n)}{2nm^2} - \mathsf{poly}(n)}$.

Assume for contradiction that there exists a $(T'(n'))$-time algorithm $\mathcal{A}'$ that inverts f' with probability $1/T'(n')$. We construct an adversary $\mathcal{A}$ inverting f. On input $(1^n, y)$, $\mathcal{A}$ samples $j \in [m]$ and let $y_j \leftarrow y$. For $i \neq j$, $\mathcal{A}$ samples $y_i \leftarrow f(\mathcal{U}_n)$. Then $\mathcal{A}$ runs $\mathcal{A}'$ to invert f' on $(y_1, \dots, y_m)$, and returns the pre-image of y if $\mathcal{A}'$ succeeds on the j-th component. To amplify the success probability, $\mathcal{A}$ will repeat the above procedure for $2nm^2T'(n')$ times. Note that $\mathcal{A}'$ runs in $T'(n')$ time, and checking whether $\mathcal{A}'$ succeeds on the j-th component takes time $\mathsf{poly}(n)$, so the running time of $\mathcal{A}$ can be bounded by $2nm^2T'(n') \cdot (T'(n') + \mathsf{poly}(n)) \le T(n)$.

To analyze the success probability of $\mathcal{A}$, let $w(y)$ denote the probability that $\mathcal{A}$ receives a correct pre-image of y from $\mathcal{A}'$ in a single attempt (of invoking $\mathcal{A}'$). We refer to a string x as being "good" if $w(f(x)) \ge \frac{1}{2m^2T'(n')}$. It follows that on a good x, $\mathcal{A}$ fails to invert $f(x)$ with probability at most

$$\left(1 - \frac{1}{2m^2T'(n')}\right)^{2nm^2T'(n')} \approx e^{-n}.$$

We then claim the probability that x is good is at least $1 - \frac{1}{2q(n)}$ when x is sampled uniformly. If this is true, it follows that $\mathcal{A}$ will invert f with probability at least $1 - \frac{1}{q(n)}$, which will conclude the proof.

To reach a contradiction, assume x is good with probability $< 1 - \frac{1}{2q(n)}$. We show that $\mathcal{A}'$ inverts f' with probability $< 1/T'(n')$ (which is a contradiction). It holds that the probability that all x_i's sampled by $\mathcal{A}'$ are good is at most $(1 - \frac{1}{2q(n)})^{2nq(n)} \approx e^{-n} < \frac{1}{2T'(n')}$, so we only have to show that, the probability that $\mathcal{A}'$ inverts f' and some x_i is bad is at most $\frac{1}{2T'(n')}$. Supposed not. It follows (by taking a union bound) that there exists an index $j \in [m]$ such that conditioned on x_j being bad,

$\mathcal{A}'$ inverts f' with probability at least $\frac{1}{2mT'(n')}$. And by an averaging argument, there exists a bad x such that $\mathcal{A}'$ inverts f' with probability at least $\frac{1}{2mT'(n')}$ when $x_j = x$. If so, on the bad string x, with probability $\frac{1}{m}$, $\mathcal{A}$ will assign $f(x)$ to y_j (in a single attempt of invoking $\mathcal{A}'$), and thus $\mathcal{A}$ will succeeds with probability at least $\frac{1}{2m^2T'(n')}$, which contradicts the fact that x is bad.