On One-way Functions and the Worst-case Hardness of Time-Bounded Kolmogorov Complexity

Yanyi Liu*
Cornell Tech
yl2866@cornell.edu

Rafael Pass†
Tel-Aviv University & Cornell Tech
rafaelp@tau.ac.il

July 12, 2023

Abstract

Whether one-way functions (OWF) exist is arguably the most important problem in Cryptography, and beyond. While lots of candidate constructions of one-way functions are known, and recently also problems whose average-case hardness characterize the existence of OWFs have been demonstrated, the question of whether there exists some *worst-case hard problem* that characterizes the existence of one-way functions has remained open since their introduction in 1976.

In this work, we present the first "OWF-complete" promise problem—a promise problem whose worst-case hardness w.r.t. BPP (resp. $\mathsf{P/poly}$) is *equivalent* to the existence of OWFs secure against PPT (resp. nuPPT) algorithms. The problem is a variant of the Minimum Time-bounded Kolmogorov Complexity problem ($\mathsf{MK}^t\mathsf{P}[s]$ with a threshold s), where we condition on instances having small "computational depth".

We furthermore show that depending on the choice of the threshold s, this problem characterizes either "standard" (polynomially-hard) OWFs, or quasi polynomially- or subexponentially-hard OWFs. Additionally, when the threshold is sufficiently small (e.g., $2^{O(\sqrt{n})}$ or $\mathsf{poly}\log n$) then *sublinear* hardness of this problem suffices to characterize quasi-polynomial/sub-exponential OWFs.

While our constructions are black-box, our analysis is *non-black box*; we additionally demonstrate that fully black-box constructions of OWF from the worst-case hardness of this problem are impossible. We finally show that, under Rudich's conjecture, and standard derandomization assumptions, our problem is not inside coAM; as such, it yields the first candidate problem believed to be outside of $\mathsf{AM} \cap \mathsf{coAM}$, or even $\mathbf{SZK}$, whose worst case hardness implies the existence of OWFs.

*Supported by a JP Morgan Fellowship. Part of work done while visiting the Simons Institute.

†Part of work done while visiting the Simons Institute. Supported in part by NSF Award CNS 2149305, NSF Award CNS-2128519, NSF Award RI-1703846, AFOSR Award FA9550-18-1-0267, FA9550-23-1-0312, a JP Morgan Faculty Award, the Algorand Centres of Excellence programme managed by Algorand Foundation, and 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, DARPA or the Algorand Foundation.

1 Introduction

A *one-way function* [DH76] (OWF) is a function f that can be efficiently computed (in polynomial time), yet no probabilistic polynomial-time (PPT) algorithm can invert f with inverse polynomial probability for infinitely many input lengths n. Whether one-way functions exist is unequivocally the most important open problem in Cryptography (and arguably the most importantly open problem in the theory of computation, see e.g., [Lev03]): OWFs are both necessary [IL89] and sufficient for many of the most central cryptographic primitives and protocols (e.g., pseudorandom generators [BM84, HILL99], pseudorandom functions [GGM84], private-key encryption [GM84], digital signatures [Rom90], commitment schemes [Nao91], identification protocols [FS90], coin-flipping protocols [Blu82], and more). These primitives and protocols are often referred to as *private-key primitives*, or "Minicrypt" primitives [Imp95] as they exclude the notable task of public-key encryption [DH76, RSA83]. Additionally, as observed by Impagliazzo [Gur89, Imp95], the existence of a OWF is equivalent to the existence of polynomial-time method for sampling hard *solved* instances for an NP language (i.e., hard instances together with their witnesses).

Cryptography from Worst-case Hardness and the "coAM Barrier": A long standing question, dating back to the original work by Diffie and Hellman [DH76], is whether OWFs can be based on the *worst-case hardness* of some NP problem; ideally, this problem should be NP-complete which would yield the existence of OWFs based on the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (which trivially is implied by the existence of OWFs). This question is usually referred to as the "holy grail" of Cryptography, and is still wide open.

Following the breakthrough result of Ajtai in 1996 [Ajt96], there has been an explosion of cryptography based on the worst-case hardness of lattice problems (see e.g., [AD97, Reg04]); these problems, however, are all in $\mathsf{AM} \cap \mathsf{coAM}$ [GG00] and are thus unlikely to be NP-complete. Indeed, starting in the early 1980's, works by Brassard [Bra83], Bogdanov and Trevisan [BT03] and Akavia, Goldreich, Goldwasser and Moshkovitz [AGGM06] show that such containment in $\mathsf{AM} \cap \mathsf{coAM}$ may be necessary—at least w.r.t. *black-box reductions*[1]. The work by Akavia et al, however, explicitly mention the possibility that non-black box techniques, although "uncommon" in complexity theory, may be useful in overcoming these barriers:

Can OWFs be based on the worst-case hardness of some promise problem $\Pi \not\subseteq \mathsf{coAM}$*?*

OWF-complete problems More generally, we may ask whether some NP problem can be used to *characterize* the existence of OWFs—namely, do "OWF-complete" problems exist?

As we will explain in more detail shortly, the above coAM black-box barriers also extend to OWF-completeness, and as such, we will here focus on defining a notion of OWF-completeness w.r.t. *non black-box* reductions—in fact, for generality, we will allow even non explicit reductions (although the actual reduction presented in this paper will be explicit).

Define the class OWF of promise problems Π having the property that there exists some polynomial-time computable function f such that if there exists some "efficient attacker" that can invert f with probability (say $1/2$) for infinitely many n, then Π can be decided on infinitely many input lengths by some "efficient attacker". Additionally, we refer to a problem Π as being OWF-hard if it holds that if Π can be decided (in the worst-case) for infinitely many input lengths by "efficient attackers", then all poly-time functions can be inverted with probability $1/2$ by "efficient attackers". Finally, Π is OWF-complete if $\Pi \in \mathsf{OWF}$ and Π is OWF-hard.

[1] We highlight that these results actually do not manage to fully rule out all black-box reduction; they either apply to so-called *non-adaptive* black-box reductions, or only to restricted types of one-way functions.

To prevent artificial complete problems (e.g., $L = SAT$ if polynomially-secure OWF exists and empty otherwise[2]), and to capture intuitions similar to those of black-box reductions (which also prevent artificial complete problems), we require the above to *simultaneously* hold any "natural" class $\mathcal{C}$ of "efficient adversaries".[3] For simplicity of notation, we here focus on $\mathcal{C} = \{\mathsf{PPT}, \mathsf{nuPPT}\}$, but we our result directly extend also to any uniform (resp. non-uniform) class of adversaries whose running time is closed under polynomial composition (e.g., poly-time, quasi-polynomial-time or subexponential-time). With the above concrete choice of "efficient attacker" (i.e., the above class $\mathcal{C}$), we have that a promise problem Π is OWF-complete if it holds that $\Pi \notin \mathsf{ioBPP}$ (resp. $\Pi \notin \mathsf{ioP/poly}$) if and only if OWFs secure against PPT (resp. secure against non-uniform polynomial-time algorithms) exist.

Given this notion of completeness, an natural questions is whether there is some promise problem that characterizes the existence of OWFs:

> *Does there exist some promise problem in* NP *that is* OWF *complete? That is,* $\Pi \notin \mathsf{ioBPP}$ *(resp.* $\Pi \notin \mathsf{ioP/poly}$*) if and only if OWFs secure against* PPT *(resp.* nuPPT*) exist?*

Black-box Barriers to OWF-complete problems As alluded to above, the barriers established by Bogdanov and Trevisan [BT03] and Akavia, Goldreich, Goldwasser and Moshkovitz [AGGM06] also yield limitations of OWF-complete problem. These works demonstrate that *non-adaptive black-box reductions* can only be used to reduce OWFs to the worst-case hardness of languages in $\mathsf{AM} \cap \mathsf{coAM}$, which under standard derandomization assumptions equals $\mathsf{NP} \cap \mathsf{coNP}$. In other words, under standard derandomization assumptions, only languages in $\mathsf{NP} \cap \mathsf{coNP}$ would exist in OWF in case we only considered Karp, or even non-adaptive black-box, reductions when defining the class OWF. But it was shown already by Blum-Impagliazzo [BI87] and Rudich [Rud88] in the 1980s that there are no so-called "fully black-box" constructions of a hard language in $\mathsf{NP} \cap \mathsf{coNP}$ based on the existence of OWFs; in fact, recently, Bitansky, Degwekar and Vaikuntanathan [BDV17] strengthened this result to show impossibility of fully black-box constructions of a hard problem in $\mathsf{NP} \cap \mathsf{coNP}$ from a host of standard cryptographic primitives, including injective OWFs and indistinguishability obfuscation [BDV17].

Of course, these results do not show that that obtaining a OWF-complete problem is impossible—only that it will require using either adaptive black-box techniques, or to use non black-box techniques, but either of these are rare, at least for the analysis of the most basic cryptographic building blocks.

1.1 Our Results

In this paper, we demonstrate the existence of a OWF-complete problem. Our problem will be natural variant of the standard time-bounded Kolmogorov complexity problem and will be based on a recent thread of literature demonstrating the existence of natural problems whose *average-case* hardness characterize the existence of OWFs. As we shall see, we will show how to use *non-black box techniques* to extend these results to also work in the worst-case regime.

In a bit more detail, we will present a promise version of the time-bounded Kolmogorov complexity problem, parametrized by a threshold s. When the threshold is large, worst-case hardness of this problem will characterize "plain" OWFs, when it is "intermediate", it will characterize quasi-polynomially secure OWFs, and when it is "small", it will characterize subexponentially secure

[2] We thank an anonymous reviewer for pointing out this "trivial" complete problem.

[3] That is, the problem is "uniform" w.r.t. the attack class. This is needed to prevent the "complete" problem from simply encoding that OWF exists w.r.t. to a specific attack model; uniformity/obliviousness w.r.t. the attack class ensures that the problem captures the "essence" of the notion of one-wayness.

OWFs. In other words, we identify not only a OWF-complete problem, but the same problem, with a different threshold is also complete for quasi-polynomial/subexponential OWFs. Additionally, as we shall see, in the regime of quasi-polynomial/subexponential OWFs, it will suffice to assume that the promise problem is (worst-case) hard w.r.t. sublinear time algorithms.

Before turning to the formal statement of our results, let us first review the recent literature connecting OWFs and Kolmogorov complexity.

On OWFs and Kolmogorov Complexity: The $\mathsf{MK}^t\mathsf{P}$ problem 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 the 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.[4]

Given a threshold, $s(\cdot)$, and a polynomial time-bound, $t(\cdot)$, let $\mathsf{MK}^t\mathsf{P}[s]$ denote the language consisting 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—indeed, this is a long-standing open problem. In [LP20], Liu and Pass recently showed that when the threshold $s(\cdot)$ is "large" (more precisely, when $s(n) = n - c\log n$, for some constant c), then mild *average-case hardness* of this language w.r.t., the uniform distribution of instances is equivalent to the existence of one-way functions (OWF).[5]

Even more recently, a different work by Liu and Pass [LP21a] demonstrated that when the threshold is smaller, and if we consider a notion of mild average-case* hardness (which roughly speaking requires average-case hardness conditioned on both YES and NO instances), then this problem characterizes also quasi-polynomial or sub-exponential one-way functions. In particular, quasi-polynomially secure and subexponentially-secure OWFs are characterized by mild average-case* hardness of $\mathsf{MK}^t\mathsf{P}[s]$ where the threshold are $s(n) = 2^{O(\sqrt{\log n})}$ and $s(n) = \mathsf{poly}\log n$ respectively. Intriguingly, their result—following a literature on so-called *hardness magnification* [OS18, MMW19, CT19, OPS19, CMMW19, Oli19, CJW19, CHO+20]—shows that it suffices to assume *sublinear* hardness of these problems to provide those characterizations (when the threshold is sublinear). We mention one caveat in these results—whereas the original result of [LP20] characterizing standard OWFs applies both in the uniform and non-uniform regime, the small threshold characterization (which only require sublinear hardness) only applies in the non-uniform regime (i.e., they characterize hardness of $\mathsf{MK}^t\mathsf{P}[s]$ with respect to non-uniform algorithms through OWFs secure against non-uniform algorithms).

Roughly speaking, our main results will show that if we consider a promise-problem variant of the $\mathsf{MK}^t\mathsf{P}[s]$ problem, then we can demonstrate the above characterization but in terms of simply

[4]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.

[5]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.

worst-case hardness of the problem. Additionally, our characterizations simultanously holds in both the non-uniform and uniform regime, as required by our definition of OWF-completness.[6]

Computational Depth and "Natural" Instances To state our results, let us first (abuse of notation) and let $\mathsf{MK}^t\mathsf{P}[s]$ denote the promise problem where:

- **YES**-instances consist of strings x such that $K^t(x) \leq s(|x|)$;
- **NO**-instances consist of string x such that $K^t(x) \geq n - 1$;

Note that the only difference between the $\mathsf{MK}^t\mathsf{P}[s]$ language defined above and this promise problem is we restrict the NO-instances to have very high K^t-complexity (i.e., they are "K^t-random"). Ideally, we would like to show that worst-case hardness of this standard problem characterizes OWFs. We will, however, need to consider a somewhat stronger hardness assumption. Roughly speaking, we will require this problem to be hard even when we restrict the inputs to be of a certain "natural" form (i.e., we require that every algorithm fails on some natural input), where naturality will be defined in a precise mathematical way.

Given a promise problem $\Pi = (\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$, and an event $Q \subseteq \{0,1\}^*$, we define the "conditioned" promise problem $\Pi|_Q \stackrel{\text{def}}{=} (\Pi_{\mathsf{YES}} \cap Q, \Pi_{\mathsf{NO}} \cap Q)$. To define naturality, we will consider the notion of *computational depth* [AFvMV06]. Recall that the computational depth of a string x is defined as $CD^t(x) = K^t(x) - K(x)$. For every function t and constant β, define the following event

$$Q^t_\beta = \{x \in \{0,1\}^* : K^t(x) - K(x) \leq \beta \log K(x)\}$$

That is, the event that the computational depth is "small" relative to $K(x)$. Intuitively, the notion of computational depth is thought of a measure of "unnaturality" of strings: arguably, only "unnatural" strings have a large gap between how much they can be efficiently and non efficiently compressed. Thus, by conditioning on strings with (relatively) small computational depth, it means that we require the problem to be hard on "natural" inputs. In other words, hardness of a promise problem conditioned on the event Q^t_β requires every algorithm to fail on some "natural" string.

We remark that the event Q^t_β is not *computable* as $K(x)$ is not computable. We mention, however, that all the result of this paper would still remain valid if replacing $K(x)$ by $K^{\mathsf{EXP}}(x) = K^{t'}(x)$ where $t'(n) = 2^{\mathsf{poly}(n)}$.

The work of Antunes and Fortnow [AF09] We highlight that Antunes and Fortnow [AF09] elegantly used computational depth to connect worst-case hardness of a problem when restricting attention to elements with small computational depth and average-case hardness on sampleable distributions. We will rely on some of the same intuitions, but emphasize a cruicial difference. [AF09] only connects the notion of *errorless* average-case hardness (i.e., average-case hardness w.r.t. algorithms that never make mistake—they either give the right answer or output $\perp$) and worst-case hardness, and their proof techniques are tailored to this notion. And for the particular problem that we consider (i.e., $\mathsf{MK}^t\mathsf{P}[s]$), it is already known [Hir18, LP21b], that worst-case hardness directly (without considering computational depth) implies errorless average-case hardness with respect to the uniform distribution, so it would seem that computational depth is not helpful. Nevertheless, as we shall see, we will be able to essentially connect worst-case hardness conditioned on instances with small computational depth and also *two-sided* error average-case hardness (and thus be able to rely on results like [LP20, LP21a]).

We are now ready to state our main theorems.

[6] In fact, it would seem that our techniques could also be applied to the average-case setting and show that the results in [LP21a] actually also work in the uniform regime.

Characterizing OWFs Our first result demonstrates the first OWF-complete problem, thus providing a positive answer to the second question in the introduction:

Theorem 1.1 (Characterizing OWFs)**.** *For every polynomial $t(n) \geq 2n$, all constant $\beta > 0, \delta > 0$, and any threshold function $s(\cdot)$, $n^\delta \leq s(n) < n-1$, the following are equivalent:*

- *OWF (resp. non-uniformly secure OWF) exists;*
- $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta} \notin \mathsf{ioBPP}$ *(resp.* $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$*)*
- $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioBPP}$ *(resp.* $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$*).*

As mentioned above, the above problem is robust in the sense that completeness holds also when considering more general classes of "efficient adversaries" such as probabilistic/non-uniform quasi-polynomal, or probabilistic/non-uniform subexponential, attackers.

Computational Depth and Average-case hardness As mentioned above, the work of Antunes and Fortnow [AF09] demonstrates that worst-case hardness of a language L conditioned on instances with small computational depth implies *errorless* average-case hardness on sampleable distributions.

One may, however, wonder whether a similar result can be shown also for two-sided error average-case hardness—which for the particular $\mathsf{MK}^t\mathsf{P}$ problem has been shown to be equivalent to OWFs (when considering average-case hardness w.r.t. the uniform distribution) [LP20]. We do not know of proof of this for general languages L (and thus the proof of Theorem 1.1 relies on a different approach), but note that as a direct corollary of Theorem 1.1 and the main results of of [LP20], we have that worst-case hardness of $\mathsf{MK}^t\mathsf{P}[n - O(\log n)]$ conditioned on instances with small computational depth is equivalent to average-case hardness of $\mathsf{MK}^t\mathsf{P}[n - O(\log n)]$ w.r.t. the *uniform distribution* (since by our results, the former is equivalent to OWFs, and by [LP20] the latter is equivalent to OWFs). In fact, under standard derandomization assumptions, we can also get average-case hardness under samplable distributions (as long as t is sufficiently bigger than the running time of the sampler)—this follows since [LP22a] recently showed (under derandomization assumptions) the equivalence between OWFs and average-case hardness of $\mathsf{MK}^t\mathsf{P}[n - O(\log n)]$ under sampleable distributions (when t is sufficiently big).

The Complexity of $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta}$ and going beyond the coAM barrier An interesting consequence of the Theorem 1.1 is the equivalence of bullet 2 and 3—that is, the hardness of the problem remains the same when the thresholds is anywhere from $n^{\Omega(1)}$ to $n-2$; as we shall see shortly, this will (likely) not be the case when the threshold is significantly smaller (as the same problem with characterize quasi-polynomial/sub-exponential OWFs).

We additionally remark that under reasonable assumptions—in particular, under Rudich's assumption [Rud97] regarding the existence of cryptographic PRGs secure against coNP algorithms, and standard derandomization assumptions (i.e., the same one used to traditionally argue that $\mathsf{MK}^t\mathsf{P}[s]$ is not inside coAM [ABK+06, Hir18]) —$\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ is not inside coAM when t, β are sufficiently big; as such, Theorem 1.1 yields the first problem believed to be outside of $\mathsf{AM} \cap \mathsf{coAM}$ whose worst-case hardness (even just) *implies* the existence of OWFs, providing a positive answer to question 1 in the introduction (under computational assumptions). In more detail, we will show that the problem (at least in some parameter regime that suffices to characterize OWFs) is not in io-coNP/poly, which contains coAM.

Theorem 1.2 (informally stated)**.** *There exists some $\beta > 0$ such that under Rudich's conjecture and standard derandomization assumption, it holds that for all sufficiently large polynomials $t(n)$,*

$$\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta} \notin \mathsf{io\text{-}coNP/poly}$$

Impossibility of Fully Black-Box Constructions As mentioned above, the proof of our main theorems rely on *non-black box techniques*. In particular, in contrast to the construction of OWFs from the average-case hardness of $\mathsf{MK}^t\mathsf{P}[s]$ of [LP20] which is *fully black-box*, we make use of the code of the attacker in analyzing the security of the OWFs. We show that this non-black box usage is needed, by demonstrating the impossibility of fully black-box constructions of OWF from the hardness of $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ when β is sufficiently big. Roughly speaking, we here refer to a reduction to $\mathsf{MK}^t\mathsf{P}$ as being *fully black-box* if both the construction and the reduction treats the universal Turing machine in the definition of K^t in a black-box way, and additionally the reduction only gets black-box access to the attacker. We note that we here also rule out *adaptive* black reductions (c.f. the results of [BT03, AGGM06] that only deal with non-adaptive ones).

Theorem 1.3. *For all sufficiently large $\beta > 0$, for all polynomials $t(n) \geq 2n$, there does not exists a fully black-box construction of OWFs from $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$.*

We highlight that perhaps surprisingly, the proof of the black-box impossibility result heavily relies on the proof techniques developed to show Theorem 1.1 (i.e., our main characterization).

Characterizing Qpoly/Subexponential OWFs We next show that the $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ problem becomes "easier" when the threshold is smaller by showing that its hardness characterizes quasi-polynomially/sub-exponentially secure OWFs when the threshold is smaller. We here additionally show that *sublinear* hardness of the same problem also characterizes the same primitive.

To simplify notation, we state these results for the setting of uniform security, but emphasize that these results (just as Theorem 1.1 where we did it explicitly) also work in the setting on non-uniform security. We highlight that this is not immediate since we are employing non-black box techniques.

Theorem 1.4 (Characterizing Quasi-polynomially Secure OWFs)**.** *For every polynomial $t(n) \geq 2n$, every constant $\beta > 0, \delta > 0$, the following are equivalent,*

- *Quasi-polynomially secure OWFs exist;*
- $\mathsf{MK}^t\mathsf{P}[2^{O(\sqrt{\log n})}]|_{Q^t_\beta} \notin \mathsf{ioBPTIME}[n^\delta]$

Theorem 1.5 (Characterizing Subexponentially Secure OWFs)**.** *For every polynomial $t(n) \geq 2n$, every constant $\beta > 0, \delta > 0$, the following are equivalent,*

- *Subexponentially secure OWFs exists;*
- $\mathsf{MK}^t\mathsf{P}[\mathsf{poly}\log n]|_{Q^t_\beta} \notin \mathsf{ioBPTIME}[n^\delta]$

Theorems 1.4 and 1.5 follow from a more general theorem characterizing T-secure OWF through the worst-case hardness of $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ where s is polynomially related to T^{-1} (see Theorem 3.3 in Section 3).

1.2 Perspective

Taken together, our results demonstrate that worst-case hardness of the *same* natural problem—that is, i.e., $\mathsf{MK}^t\mathsf{P}[s]$ conditioned on inputs being "natural" (i.e., of small computational depth) characterizes all of OWFs, quasi-polynomially secure OWFs and subexponentionally secure OWFs, depending on how the threshold is set. There are several interesting consequences one can draw from this:

- **Characterizing the "holy grail"**: As mentioned above, the holy grail of Cryptography is basing OWFs on the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (or more precisely $\mathsf{NP} \not\subseteq \mathsf{ioBPP}$). By our results, solving this problem is *equivalent* to showing that our promise problem is NP complete (perhaps with a non-black box reduction). As far as we know, there are no barriers to this since as argued above, the problem is unlikely to be inside coAM. There has been lots of recent progress (see e.g. [Ila20, ILO20, Ila21, Ila22, LP22b, Hir22]) on showing that $\mathsf{MK}^t\mathsf{P}[s]$ may be NP complete (for various variants of the problems), so there is hope that this can be done.

- **Characterizing Hardness Magnification for OWFs**: Could it be that plain OWFs imply quasi-polynomially secure (or even subexponentially secure OWFs)? Our results demonstrate that this is equivalent to demonstrating a reduction—in the *worst-case regime*—from the low threshold case to the high threshold case for our promise problem.

- **Towards $\mathsf{NP} \neq \mathsf{P}$, or even $\mathsf{NP} \not\subseteq \mathsf{BPTIME}(2^{n^\epsilon})$** As far as we know, Theorem 1.4 and 1.5 yield the first problem whose *sublinear* worst-case hardness implies that $\mathsf{NP} \neq \mathsf{P}$ (and in fact, they yield even the stronger consequence that NP cannot be solved in quasi-polynomial or subexponential time). Worst-case hardness w.r.t. sublinear time algorithms is typically easy to show for natural problem (e.g., [LP21a] even showed it for a different variant of the $\mathsf{MK}^t\mathsf{P}[s]$ problem) so there is hope that our results yield a new path toward solving the NP v.s. P problem.

- **Beyond OWFs:** Our work introduces a new non black-box technique to analyze protocols based on the hardness of Kolmogorov complexity problems. We believe these techniques will be useful also outside the realm of just OWFs. Indeed, a very recent paper [BLMP23] which follows up on ours, demonstrates how to use these techniques (and in particular how restricting attention to an appropriate analog of computational depth) can be used to get a characterization of *key-exchange agreement* [DH76] using the worst-case hardness of a Kolmogorov complexity-style problem.

Concurrent and Independent Work: A concurrent and independent elegant work by Hirahara and Nanashima [HN23] also provides a worst-case characterization of OWFs.[7] There are some conceptual similarities: both works consider worst-case hardness of a language/promise problem conditioned on instances with small computational depth.

There are also some significant differences:

- [HN23] does not actually characterize OWF but rather only so-called *infinitely-often* OWFs (which are less relevant for cryptography). Their proof technique seemingly does not extend to deal with "standard" OWFs. (In contrast, ours directly extends to also characterize infinitely-often OWFs.)

- For our problem, changing the threshold enables performing "hardness magnification" (i.e., characterizing stronger OWFs and enabling using simply sublinear worst-case hardness in the small threshold case.) Their problem/proof approach is seemingly not amenable to this.

- The actual problem they consider is less standard/more complicated ("estimating the probability that a random program outputs a certain string") than the one we consider (i.e., the standard $\mathsf{MK}^t\mathsf{P}$ problem). Additionally, they also do not rely on the standard notion of computational depth but a variant of it related to the above problem.

[7] Both papers were submitted to FOCS'23. Theirs was accepted, ours not.

As a consequence, whereas our problem is (trivially) in NP, theirs is only shown to be in AM (and even this requires some work).

- The problem in [HN23] is not proven to be OWF-complete according to our notion of completeness as the characterization only holds in the *uniform* setting (since a non black-box proof is used, security in the uniform setting does not imply security in the non-uniform setting). In contrast, we show equivalence in both the uniform and the non-uniform setting. (Conceivably, however, our new proof technique for dealing with non-uniformity may also be applicable to their problem.)

- Finally, [HN23] do not show that their problem is not contained coAM; conceivably, however, our proof technique may be applicable to show that theirs also is not in coAM.

(Of course, [HN23] also contains other intriguing results, but we are here simply comparing the characterization of OWFs.)

Despite all these differences, the results of [HN23] indicate that a conceptually different type of a OWF-complete problem may be within reach, and consequently that the class of OWF-complete promise problems may contain conceptually different types of problem (similar to NP-complete problems)—we interpret this as exciting evidence of the richness of the OWF class.

1.3 Proof Overview

We here provide a proof overview of Theorem 1.1, 1.4 and 1.5.

The Key Idea In a Nutshell To explain our approach, let us start by a simple but powerful observation. Let Π denote some decidable promise problem and let $\mathbf{KR}_c$ denote the event that $K(x) \geq n - c \log n$ (i.e, x is asymptotically "Kolmogorov Random"). Then, for every $c > 1$, worst-case hardness of the conditional promise problem $\Pi|_{\mathbf{KR}_c}$ implies mild average-case hardness of Π with respect to the uniform distribution. To see this, assume for contradiction that some *uniform* polynomial-time attacker A manages to solve Π on average with probability $1 - 1/n^{c'}$ for some sufficiently big $c' > c$. In other words, there are at most $3 \times 2^n/n^{c'}$ instances on which A fails with probability $\geq 1/3$ over its randomness. But then all those instances must have Kolmogorov complexity bounded by $\log(2^n/n^{c'}) + O(\log n)$ (to index the instance among the list of elements on which A fails with probability $\geq 1/3$, plus the additional $O(\log n)$ to describe n as well as to provide the *constant-size* description of A). But for a sufficiently large c', this is strictly smaller than $n - c \log n$, so A can *only* fail on instances outside of the promise $\mathbf{KR}_c$. Note that the above argument is *non-black box*: we rely on the fact that we have a short description of the attacker A. In fact, the above argument seemingly only shows average-case hardness w.r.t *uniform* algorithms A. However, by an additional trick we can extend it to work also for non-uniform algorithms (assuming worst-case hardness of $\Pi|_{\mathbf{KR}_c}$ with respect to non-uniform polynomial-time algorithms). Assume for contradiction that there exists some non-uniform polynomial-time algorithm with size/time bounded by n^d that breaks the average-case hardness of Π. Then, given d (which can be described in $O(1)$ bits), we can simply enumerate all possible non-uniform attackers of size up to n^d and pick the one who solves the promise problem with the highest probability, and do the rest of the argument with respect to this attacker (which now can be described using $O(1)$ bits). Note that we here need to rely on the decidability of the promise problem.

Characterizing OWF through KR: A Warm-Up We note that although the above observation is simple, it is quite powerful. It can already by used, combined with the results of [LP20], to

demonstrate that worst-case hardness of $\mathsf{MK}^t\mathsf{P}[n - c\log n]|_{\mathbf{KR}_{2c}}$ characterizes the existence of OWF for every sufficiently large c. Roughly speaking, this follows from the fact that [LP20] showed that mild average-case hardness of $\mathsf{MK}^t\mathsf{P}[n - c\log n]$ is equivalent to the existence of OWF (for every sufficiently large c); in fact, by observing the proof of [LP20], it turns out that for all sufficiently big c, the equivalence holds w.r.t. $\mathsf{MK}^t\mathsf{P}[n - c\log n]$ being $1 - 1/n^{c+3}$-average-case hard (i.e., no PPT attacker can solve the problem with probability better than $1-1/n^{c+3}$), which is implied by the worst-case hardness of $\mathsf{MK}^t\mathsf{P}[n - c\log n]|_{\mathbf{KR}_{2c}}$ by the above argument (since c is sufficiently big). In other words, worst-case hardness of $\mathsf{MK}^t\mathsf{P}[n - c\log n]|_{\mathbf{KR}_{2c}}$ implies the existence of OWFs. Furthermore, by [LP20], the existence of OWFs imply that $\mathsf{MK}^t\mathsf{P}[n - c\log n]$ is mildly hard on average, which by a standard averaging argument implies worst-case hardness of $\mathsf{MK}^t\mathsf{P}[n - c\log n]|_{\mathbf{KR}_{2c}}$ since the probability of $\mathbf{KR}_{2c}$ is $1 - O(1/n^{2c})$.

The reader may wonder why we need to through the result of [LP20] at all here: the "key-observation" shows that worst-case hardness conditioned on $\mathbf{KR}_{2c}$ yields average-case hardness w.r.t. the uniform distribution. And as noted in the previous sentence, average-case hardness w.r.t. the uniform distribution implies worst-case hardness conditioned on $\mathbf{KR}_{2c}$, so it would seem that two-sided error average-case and worst-case hardness conditioned on $\mathbf{KR}_2c$ are equivalent! (and then we can just rely [LP20] in a black-box way to get a OWF-complete problem). There is an important issue with this approach: worst-case hardness conditioned on $\mathbf{KR}_{2c}$ only implies and weak form of average-case hardness, but in the other direction we require a quantititively stronger form of average-case hardness to get back worst-case hardness conditioned on $\mathbf{KR}_{2c}$. Going through [LP20], and its cryptographic machinery, enables doing this amplification. (So, at the end of the day, with respect to the particular $\mathsf{MK}^t\mathsf{P}$ problem, it is the case that two-sided error average-case and worst-case hardness conditoned on high K-complexity are equivalent, but proving so relies on going through OWFs and cryptographic machinery).

Characterizing OWFs through CD While the above yields a simple characterization of OWFs, it requires tightly calibrating the constant c in the definition of $\mathbf{KR}_c$ to the threshold of the $\mathsf{MK}^t\mathsf{P}$ problem, so it makes for a somewhat brittle characterization. Furthermore, the simple argument above only works to considering hardness of $\mathsf{MK}^t\mathsf{P}[s]$ where $s = n - O(\log n)$, and as such will not be helpful when wanting to characterize quasi-polynomially/subexponentially secure OWFs.

It turns out that instead conditioning on strings having small *computational depth* [AFvMV06, AF09] enables us to deal with these issues and provides for a clean characterization where we can simply condition on the *same* event (namely Q^t_β for any $\beta > 0$), and consider $\mathsf{MK}^t\mathsf{P}[s]$ with respect to any threshold s.

The forward direction of our proof, follows similar intuition to the above, but requires going deeper into the constructions and proofs in [LP20, LP21a], and combining the high-level ideas in those proofs with intuitions similar to the ones use above. We can then show that for any constant β, worst-case hardness of $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ implies OWF or even quasi-polynomially/subexponentially secure OWF when the threshold is sufficiently small, and additionally, in the small threshold case, it suffices to just require *sublinear* time (worst-case) hardness.

For the backward direction, we may again rely on the proofs in [LP20, LP21a] combined the with the above observation to show that OWFs (resp T-secure OWFs) imply worst-case hardness of $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$. While the high-level ideas here are similar to [LP20, LP21a], we are required to provide a tighter analysis to deal with the fact that we here consider hardness of the *promise* problem $\mathsf{MK}^t\mathsf{P}[s]$, where for NO-instances, x, requires $K^t(x) \geq n - 1$. As such, the actual technical details here are somewhat different than those in [LP20, LP21a]. Additionally, [LP21a] (which considered the small threshold case in the average-case setting) unfortunately only works for non-uniform attackers. To deal with uniform attacker, we develop a new proof technique (that also ought to work in the average-

case setting and may be of independent interest). Without getting too deep in the details, the key obstacle is that [LP21a] relies on the security of a primitive (called an conditionally-secure entropy-preserving PRF (cond EP-PRF) for which it is (seemingly) hard to check if an attacker manages to break its security, and non-uniformity was used to provide the input lengths on which the attacker succeeds. We here show how to also deal with this obstacle without non-uniform advice.

Roughly speaking, the key idea is to leverage the fact that in [LP21a], a cond EP-PRF was constructed based on the existence of a PRG and a PRF (primitives for which we efficiently check whether an attacker succeeds) using an explicit (efficient) *black-box* reduction having the property that any attacker that breaks the cond EP-PRF on some specific inputs length n can be used to break the PRG (or the PRF) on some specific (and efficiently computable) input length n'. We can next use this reduction to efficiently find the input lengths on which the attacker succeeds in breaking the cond EP-PRF.

2 Preliminaries

For any string $x \in \{0,1\}^*$, we let $[x]_n$ denote the first n-bit prefix of x. For any functions $s(\cdot)$, we refer to it as a *threshold function* if s is time-constructible and strictly increasing.

Sublinear-time Algorithms If an algorithm M runs in time n^δ for some $\delta < 1$, we refer to M as a sublinear-time algorithm. Notice that sublinear-time algorithms cannot read the whole input. In this work, we assume that a (uniform) sublinear-time algorithm, when running on some input, will be additionally provided with the length of the input.

2.1 Promise Problems and "Conditioned" Problems

In this work, we focus on promise problems $\Pi = (\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$, and algorithms that decides Π on infinitely many input lengths. We say that an algorithm M decides Π infinitely often if there exists infinitely many $n \in \mathbb{N}$ such that $(\Pi_{\mathsf{YES}} \cup \Pi_{\mathsf{NO}}) \cap \{0,1\}^n \neq \emptyset$ and M decides Π on input length n.

We consider the promise variant of standard infinitely often complexity classes. Let ioBPP (resp $\mathsf{ioP/poly}, \mathsf{io\text{-}coNP/poly}$) denote the class of promise problems where for any promise problem Π, $\Pi \in \mathsf{ioBPP}$ (resp $\mathsf{ioP/poly}, \mathsf{io\text{-}coNP/poly}$) if and only if there exists a probabilistic (resp non-uniform, non-uniform co-non deterministic) polynomial time algorithm M that decides Π.

Let us introduce what it means by *"conditioned" promise problems*. For any promise problem Π, and any event $Q \subseteq \{0,1\}^*$, we define the promise problem

$$\Pi|_Q \stackrel{\text{def}}{=} (\Pi_{\mathsf{YES}} \cap Q, \Pi_{\mathsf{NO}} \cap Q)$$

Note that for any Q, Q', $Q \subseteq Q'$, we have that $\Pi|_Q \subseteq \Pi|_{Q'}$. (And therefore, $\Pi|_{Q'}$ is "harder" than $\Pi|_Q$. Namely, if $\Pi|_Q \notin \mathsf{ioBPP}$, $\Pi|_{Q'} \notin \mathsf{ioBPP}$.)

2.2 One-way Functions

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

Definition 2.1. *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 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 non-uniformly secure *if the above holds for all non-uniform algorithm $\mathcal{A}$.*

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 *weak* $T(n)$-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.

Lemma 2.2 (Hardness Amplification [Yao82])**.** *Assume that there exists a weak $T(n)$-one-way function for an arbitrary function $T(\cdot)$. Then, there exists a $(T'(n))$-one-way function where $T'(n) = \sqrt{\frac{T(n^{\Omega(1)})}{n^{O(1)}}} - n^{O(1)}$.*

We refer the read to [LP21a] for a proof of the above Lemma.

2.3 The OWF Class

We turn to defining, OWF, the class of promise problems Π whose worst-case hardness imply the existence of OWFs. Formally, $\Pi \in \mathsf{OWF}$ if and only if there exists a efficiently computable function f such that the following holds: If there exists a PPT (resp. a nuPPT algorithm $\mathcal{A}$ such that $\mathcal{A}$ inverts f infinitely often—that is, for infinitely many $n \in \mathbb{N}$,

$$\Pr[x \leftarrow \{0,1\}^n, y = f(x) : \mathcal{A}(1^n, y) \in f^{-1}(f(x))] \geq \frac{1}{2}$$

then, $\Pi \in \mathsf{ioBPP}$ (resp. $\Pi \in \mathsf{ioP/poly}$).

We highlight that containment in OWF requires the problem to be "as hard as" the efficient function f (is to invert) *simultanously* w.r.t. uniform PPT as well as non-uniform polynomial-time algorithm. This uniformity is a "proxy" for the uniformity imposed by standard definitions of black-box reductions which also provide this guarantee. (One could also extend this uniformity to hold with respect to attackers with larger, e.g., subexponential running time; while this indeed is the case for our results, we simply stick to uniform/non-uniform PPT, for simplicity of notation.)

Let us turn to define OWF-hardness. We say that Π is OWF-hard if the following holds: If $\Pi \in \mathsf{ioBPP}$ (resp. $\mathsf{ioP/poly}$), then any efficient function f can be inverted w.p. $1/2$ for infinitely many input length in probabilistic polynomial time (resp. non-uniform polynomial time).

Finally, we say that Π is OWF-complete if $\Pi \in \mathsf{OWF}$ and Π is OWF-hard. In other words, Π being OWF-complete means that $\Pi \in \mathsf{ioBPP}$ (resp. $\Pi \in \mathsf{ioP/poly}$) iff all efficient functions can be inverted for infinitely many input lengths by PPT algorithms (resp. non uniform polynomial-time algorithms).

2.4 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. 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 2.3. *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, LP21a]) where the program Π is required to efficiently output each individual bit x_i of the string x, given i as input.

A basic computational problem regarding t-time-bounded Kolmogorov complexity is the minimum K^t-complexity problem $\mathsf{MK}^t\mathsf{P}$. In this work, we consider its decisional version, which is parameterized by a threshold $s(\cdot)$, and the goal is to distinguish strings x with small K^t-complexity ($\leq s(|x|)$) from those with large K^t-complexity $\geq n-1$.

Definition 2.4 ($\mathsf{MK}^t\mathsf{P}$)**.** *Let $\mathsf{MK}^t\mathsf{P}[s]$ denote the following promise problem:*

- *YES: $x \in \{0,1\}^*$, $K^t(x) \leq s(|x|)$.*
- *NO: $x \in \{0,1\}^*$, $K^t(x) \geq n-1$.*

Computational Depth We will focus our attention on $\mathsf{MK}^t\mathsf{P}$ with instances having small computational depth [AFvMV06]. Roughly speaking, the computational depth of a string x is the difference between its K^t-complexity and its (time-unbounded) K-complexity. Recall that for any string $x \in \{0,1\}^*$, its (time-unbounded) K-complexity, $K(x)$, is defined to be the length of the shortest program that produces x. Formally,

$$K(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \exists t \in \mathbb{N}, U(\Pi, 1^t) = x\}$$

And we refer to $K^t(x) - K(x)$ as the computational depth of x. Throughout this work, for any polynomial t, any constant $\beta > 0$, we define

$$Q^t_\beta \stackrel{\text{def}}{=} \{x \in \{0,1\}^* : K^t(x) - K(x) \leq \beta \log K(x)\}$$

be the set of strings with computational depth logarithmic in $K(x)$. (And recall that $\mathsf{MK}^t\mathsf{P}|_{Q^t_\beta}$ is the promise variant of $\mathsf{MK}^t\mathsf{P}$ where we condition on instances $\in Q^t_\beta$.) As argued in the introduction, Q^t_β is the set of "natural" instances. Observe that for any polynomial $t_0, t_1, t_1(n) \geq t_0(n)$, for any constant $\beta_1 \geq \beta_0 > 0$, we have that

$$Q^{t_0}_{\beta_0} \subseteq Q^{t_1}_{\beta_0}, Q^{t_0}_{\beta_0} \subseteq Q^{t_0}_{\beta_1}$$

We recall the following fact about (time-bounded) Kolmogorov complexity (and we refer to [LP21a] for a short proof).

Fact 2.5. *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 $0 < s(n) < n$, $2^{\lfloor s(n) \rfloor - c} \leq |\mathsf{MK}^t\mathsf{P}[s(n)] \cap \{0,1\}^n| \leq 2^{\lfloor s(n) \rfloor + 1}$.

2.5 Distributions, Random Variables, and Entropy

Let $\mathcal{D}$ be a distribution. We let $\mathsf{supp}(\mathcal{D})$ denote the support of $\mathcal{D}$. For any $x \in \mathsf{supp}(\mathcal{D})$, we let $\mathcal{D}(x)$ denote $\Pr[\mathcal{D} = x]$.

For a random variable X, let $H(X) = \mathsf{E}[\log \frac{1}{\Pr[X=x]}]$ denote the (Shannon) entropy of X. The following lemma will be useful for us.

Lemma 2.6 (Implicit in [LP20, IRS22]). *Let X be a random variable distributed over $S \subseteq \{0,1\}^n$, E be an set $\subseteq S$. It holds that*

$$\Pr[x \leftarrow X : x \in E] \leq \frac{\log|S| + 1 - H(X)}{\log|S| - \log|E|}$$

Proof: Let flag be a binary random variable (jointly distributed with $x \sim X$) such that $\mathsf{flag} = 1$ if $x \in E$, and $\mathsf{flag} = 0$ if $x \notin E$. Let α denote the value of $\Pr_{x\sim X}[x \in E]$, and assume for contradiction that $\alpha > \frac{\log|S|+1-H(X)}{\log|S|-\log|E|}$. Note that by the chain rule of entropy:

$$H(X) \leq H(X, \mathsf{flag}) = H(\mathsf{flag}) + \alpha H(X|x \in E) + (1-\alpha)H(X|x \notin E)$$

Note that on the RHS, $H(\mathsf{flag}) \leq 1$ since flag is binary. $H(X|x \in E) \leq \log|E|$, and $H(X|x \notin E) \leq \log|S|$ (since X is distributed over S). So the RHS is at most

$$\begin{aligned}
\text{RHS} \leq & 1 + \alpha \log|E| + (1-\alpha)\log|S| \\
= & 1 + \log|S| - \alpha(\log|S| - \log|E|) \\
< & 1 + \log|S| - \frac{\log|S| + 1 - H(X)}{\log|S| - \log|E|}(\log|S| - \log|E|) \\
= & H(X)
\end{aligned}$$

which is a contradiction. ■

2.6 "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.
- $\mathcal{F}$ 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 promise problem, we refer to A as being $T(n)$-hard if A is hard for $T(n)$-time algorithms. 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 2.7. *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, $T'(n) \leq T(n^{\varepsilon_1})^{\varepsilon_2}/n^{c_1} - n^{c_2}$.*

We refer the reader to [LP21a] for a proof of this fact.

3 Our Results

Our first result is an equivalence between OWFs and the hardness of $\mathsf{MK}^t\mathsf{P}[s]$ on the natural instances, where the threshold s is moderately large and t is a polynomial.

Theorem 3.1 (Characterizing OWFs)**.** *For any threshold function $s(\cdot)$, $n^\varepsilon \leq s(n) < n-1, \varepsilon > 0$, any polynomial $t(n) \geq 2n$, any constant $\beta > 0$, the following are equivalent:*

(a) $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioBPP}$ *(resp.* $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$*).*

(b) One-way functions (resp. non-uniformly secure one-way functions) exist.

Proof: (b) $\Rightarrow$ (a) follows from Theorem 4.1 (stated and proved in Section 4) and Lemma 2.2. The non-uniform version of the implication (a) $\Rightarrow$ (b) follows from Theorem 5.2, and Lemma 5.4 (stated and proved in Section 5). The uniform version of the implication follows from Theorem 5.2, Proposition 5.5, and Lemma 5.7 (stated and proved in Section 5). ■

We remark that Theorem 1.1 (stated in the introduction) follows from Theorem 3.1 by taking $s = n^{\Omega(1)}$ or $s = n - 2$. In addition, this yields the first OWF-complete problem.

Corollary 3.2. *Let s, t, β as in Theorem 3.1.* $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$ *is* OWF*-complete.*

Our second result demonstrates that the hardness of the same problem, $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta}$, with respect to polynomial time (or even sublinear-time) algorithms, will characterize quasi-polynomially or subexponentially secure OWF when the threshold s is small. We rely on the notion of "nice" classes of functions, which captures classes of "polynomially-related" functions. We refer the reader to Section 2.6 for more on "nice" classes and we here proceed to our theorem statement.

Theorem 3.3 (Characterizing T-hard OWFs)**.** *Let $\mathcal{F}$ be a "nice" class of super-polynomial functions. For any polynomial $t(n) \geq 2n$, any constant $\beta > 0, \delta > 0$, the following are equivalent:*

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

(b) There exists a function $s \in \mathcal{F}^{-1}$ such that $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioBPTIME}[n^\delta]$ *(resp.* $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioSIZE}[n^\delta]$*).*

Proof: (b) $\Rightarrow$ (a) follows from Theorem 4.2 (stated and proved in Section 4) and Lemma 2.2. The non-uniform version of the implication (a) $\Rightarrow$ (b) follows from Theorem 5.2, and Lemma 5.3 (stated and proved in Section 5). Its uniform version follows from Theorem 5.2, Proposition 5.5, and Lemma 5.6 (stated and proved in Section 5). ■

Taking $\mathcal{F} = \{n^{c \log n}\}_{c>0}$ to be the class of quasi-polynomial functions, we obtain Theorem 1.4 (stated in the introduction), and taking $\mathcal{F} = \{2^{cn^\varepsilon}\}_{c>0,0<\varepsilon<1}$ to be the class of subexponential functions, we obtain Theorem 1.5 (stated in the introduction). Furthermore, we can take $\mathcal{F}$ to be any nice class of super-polynomial functions (such as $\mathcal{F} = \{n^{c \log \log n}\}_{c>0}$) and we obtain equivalence between $\mathcal{F}$-hard OWFs and the hardness of $\mathsf{MK}^t\mathsf{P}[\mathcal{F}^{-1}]$ on natural instances.

4 OWFs from Worst-case Hardness of $\mathsf{MK}^t\mathsf{P}|_Q$

We start by proving that if there exist a polynomial t and a constant $\beta > 0$ such that $\mathsf{MK}^t\mathsf{P}|_{Q^t_\beta}$ is hard, then standard (weak) OWFs exist.

Theorem 4.1. *If there exist a constant $\beta > 0$, a threshold function s, $0 < s(n) < n-1$, and a polynomial t such that $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioBPP}$ (resp. $\mathsf{ioP/poly}$), then weak one-way (resp. weak non-uniformly secure one-way) functions exist.*

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

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

where Π is the ℓ-bit prefix of Π'. Note that U only has polynomial overhead, so f can be computed in polynomial time.

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 weak one-way (over the restricted input lengths), then f' will be weak one-way (over all input lengths): $f'(x')$ simply truncates its input x' (as little as possible) so that the (truncated) input x now becomes of length $n' = n + \lceil \log(n) \rceil$ for some n and outputs $f(x)$.

Assume for contradiction that f is not $\frac{1}{q(n)}$-weak one-way (resp non-uniformly $\frac{1}{q(n)}$-weak one-way) where $q(n) = n^{\beta+4}$. There exists a polynomial $p(\cdot)$ and a p-time attacker $\mathcal{A}$ such that the attacker $\mathcal{A}$ inverts the function f with probability at least $\frac{1}{q(n)}$ for infinitely many n. We can assume without loss of generality that there exists a constant γ such that for all sufficiently large n, $\mathcal{A}$ (on input length n) can be described using γ bits given n: If $\mathcal{A}$ is a uniform attacker, we can let γ be the length of the code of $\mathcal{A}$; if $\mathcal{A} = \{A_n\}_{n \in \mathbb{N}}$ is a non-uniform attacker, on input length n, we can consider A_n as being the lexicographically smallest $p(n)$-time non-uniform attacker such that A_n inverts f with probability at least $\frac{1}{q(n)}$. (If there is no such attacker on input length n, we let A_n be simply an outputting $\perp$ attacker.) Note that A_n can be described using the code of f, the polynomial $p(\cdot)$, and the polynomial $q(\cdot)$, so the attacker $\mathcal{A}$ can be described in constant bits. Fix some n such that $\mathcal{A}$ succeeds with probability at least $\frac{1}{q(n)}$ on input length n.

We turn to constructing a polynomial-time uniform (resp non-uniform) algorithm M to decide $\mathsf{MK}^t\mathsf{P}[s]$ on inputs $z \in \{0,1\}^n \cap Q^t_\beta$. Our algorithm M, on input z, runs $\mathcal{A}(i||z)$ for every $i \in [n]$ where i is represented as a $\lceil \log(n) \rceil$-bit string, and outputs 1 if and only if the length of the shortest program Π output by $\mathcal{A}$, which produces each bit in the string z within $t(|\Pi|)$ steps, is at most $s(n)$. Since $\mathcal{A}$ runs in polynomial time, our algorithm will also terminate in polynomial time.

We next show that our algorithm will output 1 with probability at least $2/3$ on input z if $K^t(z) \leq s(n)$ and $z \in Q^t_\beta$. Assume for contradiction that M outputs 1 with probability $< 2/3$. Let $w = K^t(z)$. Consider any string y such that $K^t(y) = w$, and L_w be the set of "bad" strings such that

$$L_w \stackrel{\text{def}}{=} \{y \in \{0,1\}^n : K^t(y) = w, \Pr[M(y) = 1] < 2/3\}$$

It follows that $z \in L_w$. We rely on the following claim on the Kolmogorov complexity of strings in L_w.

Claim 1. *For all $y \in L_w$, $K(y) < w - \beta \log n$.*

Proof: We first argue that

$$|L_w| \leq 3 \cdot 2^{w - (\beta+3)\log n}$$

Consider any $y \in L_w$. Note that $K^t(y) = w$, there must exist a program Π of size w such that Π outputs each bit of y in time $t(|\Pi|)$. And (w, Π) will be sampled with probability

$$\frac{1}{n} 2^w$$

in the one-way function experiment. However, since $\Pr[M(y) = 1] < 2/3$, $\mathcal{A}$ must fail to invert f on input $(w||y)$ with probability at least $1/3$. So $\mathcal{A}$ will fail to invert f in the one-way function experiment with probability at least

$$1/3|L_w|\frac{1}{n}2^w$$

which is at most $\frac{1}{q(n)} = \frac{1}{n^{\beta+4}}$ since $\mathcal{A}$ is a good inverter. We conclude that

$$|L_w| \leq 3 \cdot 2^{w+\log(n)-(\beta+4)\log n} \leq 3 \cdot 2^{w-(\beta+3)\log n}.$$

We turn to showing how to obtain a short description for each string $\in L_w$. For any $y \in L_w$, consider the following program with n, w, t, the code of M (which as shown before, is of constant length), and the location of y (in L_w) hardwired in it. The program first generate the set L_w by enumerating all strings in $\{0,1\}^n$, and writing down the string if its K^t-complexity is w and the probability that $M(z) = 1$ is $< 2/3$. The program can be described using $2\log n + O(\log\log n) + \log|L|$ bits. So it follows that for any $y \in L_w$,

$$K(y) \leq 2\log n + O(\log\log n) + w - (\beta+3)\log n + O(1) < w - \beta\log n.$$

■

Therefore, $K(z) < w - \beta\log n$. However, recall that $K^t(z) = w$ and $z \in Q^t_\beta$, it holds that $K(z) \geq w - \beta\log K(z) > w - \beta\log n$, which is a contradiction.

We finally prove that if $K^t(z) > n-1$, $M(z)$ will never output 1. Note that $M(z)$ will output 1 only when it finds a K^t-witness of length no more than s, and there is no such witness if $K^t(z) > n-1$. It follows that $M(z)$ will never output 1. ■

We turn to showing that the smaller the threshold in Theorem 4.1 is, the stronger the OWF we deduce. And we only require sublinear-time hardness.

Theorem 4.2. *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exist a function $s \in \mathcal{F}^{-1}$, constants $\beta, \delta > 0$, and a polynomial $t > 0$ such that $\mathsf{MK^tP}[s]|_{Q^t_\beta} \notin \mathsf{ioBPTIME}[n^\delta]$ (resp $\mathsf{ioSIZE}[n^\delta]$). Then, there exist $T \in \mathcal{F}$ and a weak T-one-way (resp weak non-uniform T-one-way) function.*

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

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

where Π is the ℓ-bit prefix of Π'. Note that U only has polynomial overhead, so f can be computed in polynomial time.

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 weak T-one-way (over the restricted input lengths, for some function $T \in \mathcal{F}$), then f' will be weak 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 $T'(|x'|) \leq T(n)$, guaranteed to exist by Fact 2.7.)

We will show that f is $(T, \frac{1}{q(n)})$-one-way (resp non-uniformly $(T, \frac{1}{q(n)})$-one-way), over input lengths on which f is well defined, where $q(n) = n^{\beta+4}$, and T is picked as follows. Recall that s is a function $\in \mathcal{F}^{-1}$ such that by our assumption, $\mathsf{MK^tP}[s]|_{Q^t_\beta}$ is hard. Let $d(\cdot)$ be a polynomial such that f runs

in time $d(n)$. Pick any function $T \in \mathcal{F}$ such that $T(n) \leq (s^{-1}(n))^{\delta}/n - d(n)$ (guaranteed to exist by Fact 2.7).

Assume for contradiction that there exist a $T(n)$-time attacker $\mathcal{A}$ such that the attacker $\mathcal{A}$ inverts the function f with probability at least $\frac{1}{q(n)}$ for infinitely many n. We can assume without loss of generality that there exists a constant γ such that for all $n \in \mathbb{N}$, $\mathcal{A}$ on input length n can be described using γ bits given n: If $\mathcal{A}$ is a uniform attacker, we can let γ be the description length of $\mathcal{A}$; if $\mathcal{A} = \{A_n\}_{n\in\mathbb{N}}$ is a non-uniform attacker, on input length n, we can consider A_n as being the lexicographically smallest $T(n)$-time non-uniform attacker such that A_n inverts f with probability at least $\frac{1}{q(n)}$. (If there is no such attacker on input length n, we let A_n be simply an outputting $\perp$ attacker.) Note that A_n can be described using the code of f, the function $T(\cdot)$, and the polynomial $q(\cdot)$, so the attacker $\mathcal{A}$ can be described in constant bits (given n).

We turn to constructing a m^{δ}-time uniform (resp non-uniform) algorithm M to decide $\mathsf{MK}^t\mathsf{P}[s]$ on inputs $z \in \{0,1\}^m \cap Q^t_\beta$. Our algorithm $M(z)$ computes $n = s(m)$, and truncates z to its $(2n)$-bit prefix y. M then runs $\mathcal{A}(i||y)$ for every $i \in [n]$ where i is represented as a $\lceil \log(n) \rceil$-bit string, and outputs 1 if and only if the length of the shortest program Π output by $\mathcal{A}$, which produces each bit in the string y within $t(|\Pi|)$ steps, is at most $s(m)$. Since $\mathcal{A}$ runs in time $T(n)$, our algorithm will also terminate in time $n(T(n) + d(n)) \leq n((s^{-1}(n))^{\delta}/n - d(n) + d(n)) = (s^{-1}(n))^{\delta} \leq m^{\delta}$.

Fix some n such that $\mathcal{A}$ succeeds with probability at least $\frac{1}{q(n)}$ on input length n. We will show that M succeeds on input length m where m is the smallest $m \in \mathbb{N}$ such that $\lfloor s(m) \rfloor = n$. Since $\mathcal{A}$ succeeds on infinitely many n, it follows that M succeeds on infinitely many m.

We next show that our algorithm will output 1 with probability at least $2/3$ on input z if $K^t(z) \leq s(m)$ and $z \in Q^t_\beta$. Assume for contradiction that M outputs 1 with probability $< 2/3$. Let $w = K^t(z)$. Consider any string y such that $K^t(y) = w$, and L_w be the set of "bad" strings such that

$$L_w \stackrel{\text{def}}{=} \{y \in \{0,1\}^m : K^t(y) = w, \Pr[M(y) = 1] < 2/3\}$$

It follows that $y \in L_w$. We rely on the following claim on the Kolmogorov complexity of strings in L_w.

Claim 2. *For all $z \in L_w$, $K(y) < w - \beta \log n$.*

Proof: We first argue that

$$|L_w| \leq 3 \cdot 2^{w-(\beta+3)\log n}$$

Consider any $y \in L_w$. Note that $K^t(y) = w$, there must exist a program Π of size w such that Π outputs each bit of y in time $t(|\Pi|)$. It follows that the program Π will also output each bit of the $(2n)$-bit prefix of z in time $t(|\Pi|)$. In addition, (w, Π) will be sampled with probability

$$\frac{1}{n}2^w$$

in the one-way function experiment. However, since $\Pr[M(z) = 1] < 2/3$, $\mathcal{A}$ must fail to invert f on input $(w||z')$ (where z' denotes the $(2n)$-bit prefix of z) with probability at least $1/3$. So $\mathcal{A}$ will fail to invert f in the one-way function experiment with probability at least

$$1/3|L_w|\frac{1}{n}2^w$$

which is at most $\frac{1}{q(n)} = \frac{1}{n^{\beta+4}}$ since $\mathcal{A}$ is a good inverter. We conclude that

$$|L_w| \leq 3 \cdot 2^{w+\log(n)-(\beta+4)\log n} \leq 3 \cdot 2^{w-(\beta+3)\log n}$$

We turn to showing how to obtain a short description for each string $\in L_w$. For any $y \in L_w$, consider the following program with n, w, t, s, the code of M (which as shown before, is of constant length), and the location of y (in L_w) hardwired in it. The program computes m to be the smallest integer such that $n = \lfloor s(m) \rfloor$ and generates the set L_w by enumerating all strings in $\{0,1\}^m$, and writing down the string if its K^t-complexity is w and the probability that $M(z) = 1$ is $< 2/3$. The program can be described using $2\log n + O(\log\log n) + \log|L|$ bits. So it follows that for any $y \in L_w$,

$$K(y) \leq 2\log n + O(\log\log n) + w - (\beta+3)\log n + O(1) < w - \beta\log n.$$

■

Therefore, $K(z) < w - \beta\log n$. However, recall that $K^t(z) = w \leq n$ and $z \in Q^t_\beta$, it holds that $K(z) \geq w - \beta\log K(z) > w - \beta\log n$, which is a contradiction.

We finally prove that if $K^t(z) > m - 1$, $M(z)$ will never output 1. Note that $M(z)$ will output 1 only when it finds a K^t-witness of length no more than n for the first $(2n)$-bit prefix y. If such witness exists, we can compress z by using this witness to compute the first $2n$ bits and hardwiring the last $m - 2n$ bits of z, which will conclude that $K^t(z) \leq m - n + O(\log n)$. But $K^t(z) > m - 1$. It follows that $M(z)$ will never output 1. ■

5 Worst-case Hardness of $\mathsf{MK}^t\mathsf{P}|_Q$ from OWFs

In this section, we prove that the existence of OWFs implies worst-case hardness of $\mathsf{MK}^t\mathsf{P}|_Q$. To simplifcy the presentation, we first prove this in the non-uniform setting where the reduction has access to some non-uniform advice. We will later remove the non-uniform advice and prove it in the uniform setting.

5.1 Conditionally-Secure Entropy-Preserving Pseudorandom Functions

We start by recalling the notion of a non-uniform conditionally-secure entropy-preserving pseudorandom function [LP21a], which will be an important tool in our proof.

Definition 5.1. *An efficiently computable function $f : \{0,1\}^n \times \{0,1\}^{k(n)} \to \{0,1\}$ is a* non-uniform $(T(\cdot), \varepsilon(\cdot))$-conditionally-secure α-entropy-preserving pseudorandom function ((T,ε)-cond α-EP-PRF) *if there exist a sequence of events $= \{E_n\}_{n\in\mathbb{N}}$ such that the following conditions hold:*

- **(pseudorandomness):** *For every non-uniform $T(n)$-time attacker $\mathcal{A}$ and sufficiently large $n \in \mathbb{N}$,*

$$|\Pr[s \leftarrow \{0,1\}^n; \mathcal{A}^{f(s,\cdot)}(1^n) = 1 | E_n] - \Pr[f' \leftarrow \mathcal{F}; \mathcal{A}^{f'(\cdot)}(1^n) = 1]| < \varepsilon(n), \quad (1)$$

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

- **(entropy-preserving):** *For all sufficiently large $n \in \mathbb{N}$, $H(\mathsf{tt}_n(f(\mathcal{U}_n|E_n, \cdot))) \geq n - \alpha\log n$, where $\mathsf{tt}_n(\cdot)$ denote the n-bit prefix of the truthtable of the function.*

We refer to the constant α as the entropy-loss constant. *We say that f is a* cond EP-PRF *(without mentioning "non-uniform") if the pseudorandomness condition holds just w.r.t. all probabilistic T-time attackers. We refer to f as a (non-uniform) ε-cond α-EP-PRF if f is secure w.r.t. all (non-uniform)* PPT *attackers.*

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 $\varepsilon < 1$. Recall that a rate-1 efficient cond EP-PRF can be constructed from OWFs [LP21a]. We notice that if the OWF we start with is T-hard, then we obtain a T-hard cond EP-PRF. If the OWF is of (plain) polynomial security, we obtain a cond EP-PRF that is polynomially secure.

Theorem 5.2 (Cond EP-PRF from OWFs [LP21a])**.** *The following statement holds.*

- *(T-hard cond EP-PRF) Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exists $T \in \mathcal{F}$ and a T-hard (resp non-uniform T-hard) OWF. Then, for any constant $\alpha > 0, \delta \geq 1$, there exist $T_1 \in F$ and a rate-1 efficient $(T_1^\delta, 0.1)$-cond α-EP-PRF (resp non-uniformly secure cond EP-PRF) $f : \{0,1\}^n \times [T_1(n)] \to \{0,1\}$.*
- *(Polynomially hard cond EP-PRF) Assume that there exists a OWF (resp non-uniform OWF). Then, for any constant $\alpha > 0$ and any polynomial $d(n) \geq n$, there exists a rate-1 efficient 0.1-cond α-EP PRF (resp non-uniformly secure cond EP-PRF) $f : \{0,1\}^n \times [d(n)] \to \{0,1\}$.*

[LP21a] only proved a weaker version of the above theorem (in which they showed the existence of a cond α-EP-PRF for some constant α). A standard padding argument will be needed to prove the stronger version stated above, and we include a proof for Theorem 5.2 in the Appendix A (see Theorem A.5).

5.2 (Non-uniform) Worst-case Hardness of $\mathsf{MK}^t\mathsf{P}|_Q$ from (Non-uniform) OWFs

We turn to showing that the existence of cond EP-PRFs implies hardness of $\mathsf{MK}^t\mathsf{P}$, even when conditioned on the event Q_β^t. We will first present the proof in the non-uniform setting.

Lemma 5.3 (Hardness of $\mathsf{MK}^t\mathsf{P}[T^{-1}]$ from T-hard Cond EP-PRF)**.** *Let $\mathcal{F}$ be a nice class of super-polynomial functions, $\delta > 1, \beta > 0$ be some constants. Assume that there exist $T_1 \in \mathcal{F}$ and a non-uniformly secure rate-1 efficient $(T_1^\delta, 0.1)$-cond $(\beta/10)$-EP-PRF $h : \{0,1\}^n \times [T_1(n)] \to \{0,1\}$. Then, for every constant $\varepsilon' > 0, 0 < \delta' < \delta$, every polynomial $t(n) \geq 2n$, every $T_2 \in \mathcal{F}$ satisfying $T_2(n) \leq T_1(n/2)$, $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q_\beta^t} \notin \mathsf{ioSIZE}[n^{\delta'}]$.*

Proof: Consider any polynomial $t(n) \geq 2n$, and any constant $0 < \delta' < \delta$. Let $\varepsilon = 0.1$. We will show that for any $T_2 \in \mathcal{F}, T_2(n) \leq T_1(n/2)$, $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q_\beta^t} \notin \mathsf{ioSIZE}[n^{\delta'}]$.

Note that the truthtable of the PRF h is of length $T_1(n)$ (for seeds of length n). For any function $T_2 \in \mathcal{F}$ satisfying $T_2(n) \leq T_1(n/2)$, we will truncate the cond EP-PRF h to another cond EP-PRF f that is easier to work with (so that the truthtable of f is of length roughly $T_2(n)$). Note that both h and T_2 can be computed by uniform algorithms, let γ be a (sufficiently large) constant such that h together with T_2 can be described within $\gamma/4$ bits. Let $f : \{0,1\}^n \times [T_2(n+\gamma)] \to \{0,1\}$ be the function obtained by truncating h to the first $T_2(n+\gamma)$ entries. Note that $T_2(n+\gamma) < T_2(2n) \leq T_1(n)$ (due to our choice of T_2 and that T_2 is strictly increasing), so the truncation is always possible. Also notice that f is still a rate-1 efficient $(T_1^\delta, \varepsilon)$-cond $(\beta/10)$-EP-PRF (as h is). In addition, the code of f can be described in $\gamma/2$ bits.

We assume for contradiction that there exists some $m^{\delta'}$-time non-uniform algorithm that decides each instance of $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]$ in Q_β^t with probability $\frac{2}{3}$. By a Chernoff-type argument, we can show that there exists an algorithm M that succeeds with probability 0.99 and runs in $O(m^{\delta'})$ time (by using constant-fold parallel repetition and taking a majority vote). We will use the algorithm M to build a non-uniform attacker $\mathcal{A}(1^n)$ that breaks the cond EP-PRF f.

Note that f is a function that given a seed of length n, maps an integer $\in [T_2(n+\gamma)]$ 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 $f(x,\cdot)$. Consider any integer $m \leq T_2(n+\gamma)$. Note that for any $x \in \{0,1\}^n$, $\mathsf{tt}_m(f(x,\cdot))$ has low K^t-complexity (with probability 1):

$$K^t(\mathsf{tt}_m(f(x,\cdot))) \leq n+\gamma-1$$

since a Turing machine that contains the code of f (of length $\gamma/2$, as argued above) 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:

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

since there are at most 2^{m-1} Turing machines with description length no longer than $m-2$, and each of them can produce at most a single truth table of length m.

With the above observations, we are ready to construct $\mathcal{A}$ (which breaks f). On input length n, let m be an integer such that $m \in [T_2(n+\gamma-1), T_2(n+\gamma)-1]$ and the algorithm M succeeds in deciding $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q^t_\beta}$ on input length m. We will provide our attacker $\mathcal{A}$ with the integer m as non-uniform advice. (Since M only succeeds infinite often, $\mathcal{A}$ simply aborts if such m doesn't exist.) With the advice m, the attacker (denoted by $\mathcal{A}_m$) proceeds as follows. Given black-box access to a function $f' : [T_2(n+\gamma)] \to \{0,1\}$, $\mathcal{A}_m(1^n)$ first queries f' on every input $i \in [m]$ and obtains the first m bits of the truth table of f', $\mathsf{tt}_m(f')$. Then $\mathcal{A}_m(1^n)$ feeds $\mathsf{tt}_m(f')$ to the algorithm M and outputs $M(\mathsf{tt}_m(f'))$. Note that the attacker $\mathcal{A}_m(1^n)$ runs in time $O(m)+m^{\delta'} < T_1(n)^\delta$.

Since M decides $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]$ on each instances $\in Q^t_\beta$ with probability 0.99 on infinitely many input lengths m, we will show that the attacker $\mathcal{A}$ succeeds in distinguishing the cond EP-PRF f from random functions on infinitely many input lengths (which is a contradiction). Fix some input length m on which M succeeds. Let n be the integer such that

$$T_2(n+\gamma-1) \leq m < T_2(n+\gamma)$$

(guaranteed to exist since T_2 is strictly increasing). The following two claims will show that $\mathcal{A}_m$ will distinguish f from random with probability at least 2ε, which conclude the proof.

Claim 3. *$\mathcal{A}_m(1^n)$ will output 0 with probability at least $\frac{1}{2}-0.02$ when given access to f_r, where f_r is uniformly sampled from $\mathcal{F} = \{f_\mathsf{r} : [T_2(n+\gamma)] \to \{0,1\}\}$.*

Proof: Note that for a random f_r, the probability that $K^t(\mathsf{tt}_m(f_\mathsf{r})) \geq m-1$ is at least $\frac{1}{2}$. Note that $K(\mathsf{tt}_m(f_\mathsf{r}))$ is at least $m-\beta\log m$ with probability 0.01, so it follows that $\mathsf{tt}_m(f_\mathsf{r}) \in Q^t_\beta$ if $K^t(\mathsf{tt}_m(f_\mathsf{r})) \geq m-1$. Since M decides $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q^t_\beta}$ with probability 0.99 on input length m, by a Union bound, $\mathcal{A}_m(1^n)$ will output 0 with probability $\frac{1}{2}-0.02$. ∎

Claim 4. *$\mathcal{A}_m(1^n)$ will output 0 with probability at most $0.2+0.01$ when given access to $f \leftarrow f(\mathcal{U}_n|E_n,\cdot)$, where E_n is the event associated with f.*

Proof: Recall that M decides each instance of $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]$ in Q^t_β with probability 0.99 on input length m. Let $s = \lfloor T_2^{-1}(m) \rfloor$ and notice that

$$s = n+\gamma-1$$

(by the choice of n). Let

$$X = \mathsf{tt}_m(f(\mathcal{U}_n|E_n,\cdot))$$

be the random variable of the first m bits of the truth table of f. Recall that any string x in the support of X will have K^t-complexity at most $n+\gamma-1=s$. Since f is entropy preserving, the entropy of X is at least

$$H(X) \geq n - 0.1\beta \log n$$

Let

$$S = \mathsf{supp}(X)$$

be the set of pseudorandom truth tables. It follows that $2^{H(X)} \leq |S| \leq 2^n$. Let

$$Z = \{z \in S : z \notin Q^t_\beta\}$$

be the set of pseudorandom truth tables that are outside of Q^t_β. Recall that the algorithm M is only guaranteed to work if the input is in Q^t_β, so we can think of Z as the set of "bad" strings. For any $z \in Z$, since $z \in S$ and $z \notin Q^t_\beta$, it holds that $K(z) < K^t(z) - \beta \log K(z) \leq s - \beta \log s$. By a standard counting argument w.r.t. K-complexity, it follows that $|Z| \leq 2^{s-\beta \log s+1}$. Recall that X is a random variable distributed over S, and $Z \subseteq S$. By Lemma 2.6, it follows that the probability that $X \in Z$ is at most

$$\Pr[X \in Z] \leq \frac{\log|S| + 1 - H(X)}{\log|S| - \log|Z|} \leq \frac{n+1-(n-0.1\beta\log n)}{n-0.1\beta\log n-(s-\beta\log s)} \leq 0.2$$

when n is sufficiently large. Therefore, the probability that $M(X)$ outputs 0 is at most

$$\begin{aligned}
\Pr[M(X)=0] =& \Pr[X \in Z]\Pr[M(X)=0|X \in Z] + \Pr[X \notin Z]\Pr[M(X)=0|X \notin Z] \\
\leq& \Pr[X \in Z] + \Pr[M(X)=0|X \notin Z] \\
\leq& 0.2 + \Pr[M(X)=0|X \in Q^t_\beta] \\
\leq& 0.2 + 0.01
\end{aligned}$$

where the last step follows from the correctness of M. ■

■

While in the above theorems, we assume super polynomial hardness, the reduction runs in polynomial time. So the statement will also hold in the polynomial hardness region.

Lemma 5.4 (Hardness of $\mathsf{MK^tP}[n^{\Omega(1)}]$ from poly-hard Cond EP-PRF)**.** *Let $s(\cdot)$ be a threshold function, $n^\varepsilon \leq s(n) \leq n-2$, $\varepsilon > 0$. Let $d(\cdot)$ be a polynomial such that $s^{-1}(n) \leq d(n/2)$, and $\beta > 0$ be a constant. Assume that there exists a rate-1 efficient non-uniformly secure $(\mathsf{poly}, 0.1)$-cond $(\beta/10)$-EP-PRF $f : \{0,1\}^n \times [d(n)] \to \{0,1\}$. Then, for every constant $\varepsilon' > 0$, every polynomial $t(n) \geq (1+\varepsilon')n$, $\mathsf{MK^tP}[s]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$.*

Proof: This lemma follows from the proof of Lemma 5.3 by considering T_1 being d and T_2 being s^{-1}. ■

5.3 Eliminating the Non-uniform Advice

The key observation we rely on in this section is that the security of our cond EP-PRF is established through *black-box* reductions to standard cryptographic primitives. Let us introduce the notion of black-box reductions we rely on.

We say that a (T, ε)-cond EP-PRF f has a *polynomial-time black-box security reduction* to a $(T_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-PRG f_{prg} and a $(T_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}})$-PRF f_{prf} if there exist functions $l_{\mathsf{prg}}, l_{\mathsf{prf}}$ (referred to as *input*

length functions) [8] , polynomials $p_{\mathsf{prg}}, p_{\mathsf{prf}}$ (referred to as *security loss functions*), and oracle machines $R_{\mathsf{prg}}, R_{\mathsf{prf}}$ (referred to as *reductions*) such that the following are satisfied:

- $l_{\mathsf{prg}}, l_{\mathsf{prf}}$ are time-constructible and increasing.
- For any T-time probabilistic adversary $\mathcal{A}$, and any input length n (for f), let n_{prg} (resp n_{prf}) be the input length for PRG (resp PRF) such that

$$l_{\mathsf{prg}}(n_{\mathsf{prg}}) \leq n < l_{\mathsf{prg}}(n_{\mathsf{prg}}+1), l_{\mathsf{prf}}(n_{\mathsf{prf}}) \leq n < l_{\mathsf{prf}}(n_{\mathsf{prf}}+1)$$

  (Note that such $n_{\mathsf{prg}}, n_{\mathsf{prf}}$ always exist since $l_{\mathsf{prg}}, l_{\mathsf{prf}}$ are increasing.) If $\mathcal{A}(1^n)$ distinguishes the cond EP-PRF f from random functions on input length n with advantage $\varepsilon(n)$, then

  (a) either $R^{\mathcal{A}}_{\mathsf{prg}}(1^{n_{\mathsf{prg}}})$ distinguishes f_{prg} from random with advantage $\frac{1}{p_{\mathsf{prg}}(n_{\mathsf{prg}}, 1/\varepsilon(n))} \geq 4\varepsilon_{\mathsf{prg}}(n_{\mathsf{prg}})$ on input length n_{prg} in time $T_{\mathsf{prg}}(n_{\mathsf{prg}})$;

  (b) or $R^{\mathcal{A}}_{\mathsf{prf}}(1^{n_{\mathsf{prf}}})$ distinguishes f_{prf} from random functions with advantage $\frac{1}{p_{\mathsf{prf}}(n_{\mathsf{prf}}, 1/\varepsilon(n))} \geq 4\varepsilon_{\mathsf{prf}}(n_{\mathsf{prf}})$ on input length n_{prf} in time $T_{\mathsf{prf}}(n_{\mathsf{prf}})$.

  In other words, if $\mathcal{A}$ breaks f on input length n, either we break f_{prg} on input length n_{prg}, or we break f_{prf} on input length n_{prf}.

If such a black-box reduction exists, we can prove that f is indeed a (T, ε)-cond EP-PRF if f_{prg} is a $(T_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-PRG and f_{prf} a $(T_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}})$-PRF. Note that the security parameters $T_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}}$ (and $T_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}}$) for the PRG (and the PRF) will usually be implicit in (but can be inferred from) the reduction itself, and we sometimes simply omit them if they will be clear from the reduction.

As mentioned before, we observe that the cond EP-PRF we obtain in Theorem 5.2 has a black-box security reduction.

Proposition 5.5. *Let f be the cond EP-PRF in Theorem 5.2. f has a black-box security reduction to a PRG and a PRF.*

(In Appendix A, we will formally state and prove that f indeed has a black-box reduction. See Theorem A.5.)

We proceed to proving the uniform version of Lemma 5.3.

Lemma 5.6. *Let $\mathcal{F}$ be a nice class of super-polynomial functions, $\delta > 2, \beta > 0$ be any constants. Assume that there exist $T_1 \in \mathcal{F}$ and a rate-1 efficient $(T_1^{\delta}, 0.1)$-cond $(\beta/10)$-EP-PRF $h : \{0,1\}^n \times [T_1(n)] \to \{0,1\}$ with a poly-time black-box security reduction to a $(T_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-PRG f_{prg} and a $(T_{\mathsf{prf}}, \varepsilon_{\mathsf{prf}})$-PRF f_{prf}. Then, for every constant $0 < \delta' < \delta - 1$, every polynomial $t(n) \geq 2n$, every $T_2 \in \mathcal{F}$ satisfying $T_2(n) \leq T_1(n/2)$, $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q^t_\beta} \notin \mathsf{ioBPP}$.*

Proof: This proof relies heavily on the proof of Lemma 5.3, which we refer the reader to for notations used in this proof. In the proof of Lemma 5.3, we showed that the attacker $\mathcal{A}_m$ will break the cond EP-PRF f on input length n if $m \in [T_2(n+\gamma-1), T_2(n+\gamma)-1]$ and the algorithm M succeeds in deciding $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]|_{Q^t_\beta}$ on input length m. The issue in the proof of Lemma 5.3 is that we do not know which m is "good" and we provide the attacker a good m using non-uniform advice. In this proof, we show that the attacker can compute the value of m on its own by appealing to the black-box security reduction. We now explain how to compute the value m.

[8] Note that we consider reductions that establish almost-everywhere security, so it is important for the reduction to specify on which input lengths it works.

On input length n, our uniform attacker $\mathcal{A}'$ emulates all $m \in [T_2(n+\gamma-1), T_2(n+\gamma)-1]$. It needs to decide if $\mathcal{A}_m$ is a good attacker for the cond EP-PRF f. Since f has a black-box security reduction to f_{prg} and f_{prf}, the attacker $\mathcal{A}'$ will use $\mathcal{A}_m$ together with the reductions to try breaking f_{prg} and f_{prf}. Let us first focus on breaking f_{prg} and let $l_{\mathsf{prg}}, p_{\mathsf{prg}}, R_{\mathsf{prg}}$ be the input length function, security loss function, and the reduction associated with the black-box reduction to the PRG. We first compute n_{prg} (to be the input length of f_{prg}) such that

$$l_{\mathsf{prg}}(n_{\mathsf{prg}}) \leq n < l_{\mathsf{prg}}(n_{\mathsf{prg}}+1)$$

Then the attacker $\mathcal{A}'$ will empirically estimate the distinguishing advantage of $R_{\mathsf{prg}}^{\mathcal{A}_m}$ on input length n_{prg} in the distinguishing game for f_{prg}. This can be done by sampling from the f_{prg} distribution (or the uniform distribution) for sufficiently many times, simulating $R_{\mathsf{prg}}^{\mathcal{A}_m}$ on the outcome of the sampler, and finally taking the average over all results. Let us denote

- the distinguishing advantage of $R_{\mathsf{prg}}^{\mathcal{A}_m}$ by $v_{\mathsf{prg},m}$;
- the empirical estimation of the advantage by $w_{\mathsf{prg},m}$.

As argued in the proof of Lemma 5.3, if m is "good", $\mathcal{A}_m$ will succeed in distinguishing f from random functions with advantage > 0.1. Since the reduction is also good, it follows that the distinguishing advantage $v_{\mathsf{prg},m}$ is at least

$$\delta_{\mathsf{prg}} = \frac{1}{p_{\mathsf{prg}}(n_{\mathsf{prg}}, 1/\varepsilon)}$$

Therefore, to obtain an accurate empirical estimation (with high probability), the sampling experiment will be repeated for $\mathsf{poly}(\delta_{\mathsf{prg}}^{-1} n)$ times. After computing the empirical estimations $w_{\mathsf{prg},m}$ for all m, let m^*_{prg} be such that the empricial estimated distinguishing advantage $w_{\mathsf{prg},m^*_{\mathsf{prg}}}$ is maximized. If $w_{\mathsf{prg},m^*_{\mathsf{prg}}} \geq \frac{1}{2}\delta_{\mathsf{prg}}$, our uniform attacker attacker $\mathcal{A}'$ will output what $\mathcal{A}_{m^*_{\mathsf{prg}}}$ outputs. Otherwise, the attacker $\mathcal{A}'$ will redo the above steps to attack the PRF f_{prf} using the black-box reduction. If the attacker doesn't succeed either, it will simply abort. This concludes the construction of our attacker $\mathcal{A}'$.

To reach a contradiction, we will show that either $R_{\mathsf{prg}}^{\mathcal{A}'}$ will break f_{prg}, or $R_{\mathsf{prf}}^{\mathcal{A}'}$ will break f_{prf}, on infinitely many input lengths. Since M succeeds (in deciding $\mathsf{MK}^t\mathsf{P}[T_2^{-1}]$ conditioned on Q^t_β) on infinitely many input lengths m, fix some such m and let n be the input length for f such that $m \in [T_2(n+\gamma-1), T_2(n+\gamma)-1]$. It follows that $\mathcal{A}_m$ will be a good attacker for f on input length n. We turn to showing that $\mathcal{A}'$ will pick some input length that is "as good as" m. For each input length m' that $\mathcal{A}'(1^n)$ enumerates, since the sampling experiment (to estimate the distinguishing advantage $v_{\mathsf{prg},m'}$) is repeated for $\mathsf{poly}(\delta_{\mathsf{prg}}^{-1} n)$ times, by a standard Chernoff-type argument, with probability at least $1 - 2^{-n}$, it holds that

$$|v_{\mathsf{prg},m'} - w_{\mathsf{prg},m'}| \leq \frac{\delta_{\mathsf{prg}}}{4}$$

(And the same also holds for experiments w.r.t the PRF.) We refer to an estimation as being "good" if the above holds. By a union bound, with probability at least $1 - 2 \cdot T_2(n+\gamma) \cdot 2^{-n}$, all empirical estimations done by $\mathcal{A}'$ are good, and we denote the event for which this happens by E. Conditioned on E, either $R_{\mathsf{prg}}^{\mathcal{A}_m}$ will break f_{prg} on input length n_{prg}, or $R_{\mathsf{prf}}^{\mathcal{A}_m}$ will break f_{prf} on input length n_{prf}. It follows that either m^*_{prg} is a good input length and $R_{\mathsf{prg}}^{\mathcal{A}_{m^*_{\mathsf{prg}}}}$ has distinguishing advantage at least

$$v_{\mathsf{prg},m^*_{\mathsf{prg}}} \geq w_{\mathsf{prg},m^*_{\mathsf{prg}}} - \frac{\delta_{\mathsf{prg}}}{4} \geq w_{\mathsf{prg},m} - \frac{\delta_{\mathsf{prg}}}{4} \geq v_{\mathsf{prg},m} - \frac{\delta_{\mathsf{prg}}}{2} \geq \frac{\delta_{\mathsf{prg}}}{2}$$

in the PRG game, or m^*_{prf} is a good input length and $R_{\mathsf{prf}}^{\mathcal{A}_{m^*_{\mathsf{prf}}}}$ has distinguishing advantage at least

$$v_{\mathsf{prf},m^*_{\mathsf{prf}}} \geq w_{\mathsf{prf},m^*_{\mathsf{prf}}} - \frac{\delta_{\mathsf{prf}}}{4} \geq w_{\mathsf{prf},m} - \frac{\delta_{\mathsf{prf}}}{4} \geq v_{\mathsf{prf},m} - \frac{\delta_{\mathsf{prf}}}{2} \geq \frac{\delta_{\mathsf{prf}}}{2}$$

in the PRF game. Recall that $\mathcal{A}'$ in the end will accept $\mathcal{A}_{m^*_{\mathsf{prg}}}$ (resp $\mathcal{A}_{m^*_{\mathsf{prf}}}$) as the attacker if $v_{\mathsf{prg},m^*_{\mathsf{prg}}} \geq \frac{\delta_{\mathsf{prg}}}{2}$ (resp $v_{\mathsf{prf},m^*_{\mathsf{prf}}} \geq \frac{\delta_{\mathsf{prf}}}{2}$), by a union bound (taking into account that E may not happen), either $R_{\mathsf{prg}}^{\mathcal{A}'}$ will break f_{prg} with advantage $\frac{\delta_{\mathsf{prg}}}{2} - 2 \cdot T_2(n+\gamma) \cdot 2^{-n} \geq \frac{\delta_{\mathsf{prg}}}{4} \geq \varepsilon_{\mathsf{prg}}(n_{\mathsf{prg}})$ on input length n_{prg}, or $R_{\mathsf{prf}}^{\mathcal{A}'}$ will break f_{prf} with advantage $\frac{\delta_{\mathsf{prf}}}{2} - 2 \cdot T_2(n+\gamma) \cdot 2^{-n} \geq \frac{\delta_{\mathsf{prf}}}{4} \geq \varepsilon_{\mathsf{prf}}(n_{\mathsf{prf}})$ on input length n_{prf}, which is a contradiction.

Finally, we analyze the running time of $\mathcal{A}'$. Recall that $\mathcal{A}'$ tries all possible $m \in [T_2(n+\gamma-1), T_2(n+\gamma)-1]$, and for each m, it runs the PRG and the PRF (of running time $\mathsf{poly}(n)$) and the reduction $R_{\mathsf{prg}}^{\mathcal{A}_m}$ and $R_{\mathsf{prf}}^{\mathcal{A}_m}$ (of running time $\mathsf{poly}(n) \cdot m^{\delta'}$) for $\mathsf{poly}(\delta_{\mathsf{prg}}^{-1} n + \delta_{\mathsf{prf}}^{-1} n) = \mathsf{poly}(n)$ times. So $\mathcal{A}'(1^n)$ runs in time $\mathsf{poly}(n) m^{1+\delta'} \leq \mathsf{poly}(n) T_1(n)^{1+\delta'} < T_1(n)^{\delta}$ (since T_1 is super polynomial and $\delta' + 1 < \delta$). ■

We notice that Lemma 5.6 also holds w.r.t. polynomial hardness (since the reduction runs in polynomial time).

Lemma 5.7. *Let $s(\cdot)$ be a threshold function, $n^{\varepsilon} \leq s(n) \leq n-2$, $\varepsilon > 0$. Let $d(\cdot)$ be a polynomial such that $s^{-1}(n) \leq d(n/2)$, and $\beta > 0$ be a constant. Assume that there exists a rate-1 efficient $(\mathsf{poly}, 0.1)$-cond $(\beta/10)$-EP-PRF $f : \{0,1\}^n \times [d(n)] \to \{0,1\}$ with a black-box security reduction to a $(\mathsf{poly}, \varepsilon_{\mathsf{prg}})$-PRG f_{prg} and a $(\mathsf{poly}, \varepsilon_{\mathsf{prf}})$-PRF f_{prf}. Then, for every constant $\varepsilon' > 0$, every polynomial $t(n) \geq (1+\varepsilon')n$, $\mathsf{MK}^t\mathsf{P}[s]|_{Q_\beta^t} \notin \mathsf{ioBPP}$.*

6 Rudich's Conjecture and Non-containemnt in coAM

In this section, we show that the promise problem that characterized OWFs is unlikely to be in coAM. As such, it yields the first example of problem outside of AM ∩ coAM whose worst-case hardness even just implies the existence of OWFs.

We will rely on Rudich's conjecture as well as standard derandomization assumptions. Rudich [Rud97] conjectured the existence of a pseudorandom generator secure against (co-)non deterministic attackers. Let us recall the definition of such PRGs.

Definition 6.1 ([Rud97])**.** *Let $g : \{0,1\}^n \to \{0,1\}^{n+1}$ be an efficiently computable function. We say that g is a* pseudorandom generator with non-deterministic hardness *if for all poly-time non-uniform non-deterministic machine $\mathcal{A}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$,*

$$\Pr[\mathcal{A}(1^n, \mathcal{U}_{n+1}) = 1] - \Pr[\mathcal{A}(1^n, g(\mathcal{U}_n)) = 1] \leq \mu(n)$$

In other words, no non-uniform attacker can prove random strings are "random" with higher probability than pseudorandom strings. Notice that the order of the probabilities is important – there exists a trivial attacker (by just guesing the seed) if the order is switched.

We are now ready to state the Rudich's conjecture that we rely on.

Conjecture 6.2 (Implied by [Rud97, Conjecture 4])**.** *There exists a PRG $g : \{0,1\}^n \to \{0,1\}^{n+1}$ with non-deterministic hardness.*

We rely on the following Coding Theorem for our local compression variant of time-bounded Kolmogorov complexity.

Lemma 6.3 (Coding Theorem for the local compression version of K^t, implicit in [LOZ22, GKLO22])**.** *Assume that* $\mathsf{E} \not\subseteq \mathsf{ioNSIZE}[2^{\Omega(n)}]$*. There exists a polynomial* $p_c(\cdot)$ *such that the following holds. For any polynomial* $q(\cdot)$*, any distribution* $\mathcal{D}$ *over* $\{0,1\}^n$*. If* $\mathcal{D}$ *can be sampled by an algorithm* $M_{\mathcal{D}}$ *within time* $q(n)$*, then for every* $x \in \mathsf{supp}(\mathcal{D})$*,* $\mathcal{D}(x) \leq 2^{-n/2}$*, it holds that*

$$K^t(x|M_{\mathcal{D}}) \leq \log \frac{1}{\mathcal{D}(x)} + \log(t(n))$$

where t *is a polynomial such that* $t(n) = p_c(q(2n))$*.*

Proof: [LOZ22, GKLO22] proved that there exists a polynomial p_c, such that for any q, $\mathcal{D}$, $M_{\mathcal{D}}$, any $x \in \mathsf{supp}(\mathcal{D})$, $t'(n) = p_c(q(n))$, there exists a program Π of length $\leq \log \frac{1}{\mathcal{D}(x)} + \log(t'(n))$ such that $\Pi(M_{\mathcal{D}})$ prints the whole string x with in time $t'(n)$. Π can be easily made into a program that outputs each bit of x in time roughly $t'(n)$. However, in our local compression variant of K^t-complexity, the running time is measured with respect to $|\Pi|$ (rather than n). If Π is too short, the running time of Π will no longer be polynomial in its length (but still polynomial in n). In this lemma, we only consider x such that $\mathcal{D}(x) \leq 2^{-n/2}$, and $t(n) = p_c(q(2n))$. It follows that we can pad any Π until it is of length $\geq n/2$, and Π runs in time $t'(n) = p_c(q(n)) \leq t(|\Pi|)$. Thus, Π will be a valid witness of (the local compression version of) K^t-complexity for x, which completes the proof. ■

We proceed to proving that under Rudich's conjecture (and assuming an appropriate derandomization assumption), the promise problem we consider is not contained in coAM. (In more detail, we will show that it is not in $\mathsf{io\text{-}coNP/poly}$, which contains coAM.)

Theorem 6.4. *Assume that Conjecture 6.2 holds and* $\mathsf{E} \not\subseteq \mathsf{ioNSIZE}[2^{\Omega(n)}]$*. Then, there exists a constant* $\beta > 0$*, a polynomial* $t_1(n)$*, such that for all polynomials* $t(n) \geq t_1(n)$*,* $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q^t_\beta} \notin \mathsf{io\text{-}coNP/poly}$*.*

Proof: Let g be the PRG assumed to exist in Conjecture 6.2. Let γ be a sufficiently large constant that we will fix later. Using a standard hybrid argument, we can construct a PRG $g' : \{0,1\}^n \to \{0,1\}^{n+\gamma}$ with γ-bit stretch and non-deterministic hardness from g. By a padding argument, there exists a rate-1 efficient PRG $g'' : \{0,1\}^n \to \{0,1\}^{n+\gamma}$ (with non-deterministic hardness). For any $n \in \mathbb{N}$, consider the function $G_n : \{0,1\}^{n-\gamma} \to \{0,1\}^n$ defined to be

$$G_n(x) = x_0 || g''(x_1)$$

where $x = x_0||x_1$, $|x_0| = n/2$, $|x_1| = n/2 - \gamma$. Note that G_n is also rate-1 efficient PRG (since g'' is). Let $\mathcal{D}_n = G_n(\mathcal{U}_{n-\gamma})$ be the pseudorandom distribution G_n defines; it follows that $\mathcal{D}_n$ can be sampled using G_n in time $O(n)$ (since G_n is rate-1 efficient). Observe that for any $y \in \mathsf{supp}(\mathcal{D}_n)$, $\mathcal{D}_n(y) \leq 2^{-n/2}$ since G_n is copying the first half of the seed to the output. Thus, by the Coding Theorem for our local notion of K^t, it follows that there exists a polynomial $t_1(n)$ such that for all n, for all $y \in \mathsf{supp}(\mathcal{D}_n)$,

$$K^{t_1}(y|G_n) \leq \log \frac{1}{\mathcal{D}_n(y)} + \log t_1(n)$$

Since G_n can be described using $\leq 2\log n$ bits, it follows that $K^{t_1}(y) \leq \log \frac{1}{\mathcal{D}_n(y)} + \log t_1(n) + 2\log n$.

On the other hand, by [HIL+23, Lemma 9], with probability at least $1 - \frac{1}{n}$ over $y \leftarrow \mathcal{D}_n$, it holds that $K(y) \geq \log \frac{1}{\mathcal{D}_n(y)} - \log n \geq n/2 - \log n$. Pick β be a constant such that $\beta \log(n/2 - \log n) \geq \log t_1(n) + 3\log n$. Then, with probability at least $1 - \frac{1}{n}$ over $y \leftarrow \mathcal{D}_n$, the computational depth of y is at most

$$K^{t_1}(y) - K(y) \leq \log t_1(n) + 3\log n \leq \beta \log(n/2 - \log n) \leq \beta \log K(y) \tag{2}$$

and thus $y \in Q_\beta^{t_1}$.

We move on to showing the hardness of $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q_\beta^t}$ for any polynomial $t(n) \geq t_1(n)$. Suppose for contradiction that there exist a polynomial T and a T-time non-uniform co-non deterministic algorithm M that decides $\mathsf{MK}^t\mathsf{P}[n-2]|_{Q_\beta^t}$ for infinitely many $n \in \mathbb{N}$. Fix some sufficiently large $n \in \mathbb{N}$. We will show that M as an attacker will break the PRG G_n. For any $y \leftarrow \mathcal{D}_n$, we argue that the K^t-complexity of y is at most $n-2$. (Recall we have shown that $K^t(y) \leq K^{t_1}(y) \leq \log \frac{1}{\mathcal{D}_n(y)} + O(\log n)$, but this bound falls short here.) Let x be the seed of y. Consider the program with x (of length $n-\gamma$), the constant γ, and the code of g'' hardwired. It first reads the constant γ and the string x, and computes the length of x. It lets $n = |x| + \gamma$, and after this it can compute $G_n(x)$ using the code of g''. This program can be described using $\leq n-2$ bits if γ is sufficiently large. Since G_n is rate-1 efficient and $t(n) \geq t_1(n)$, this program witnesses that $K^t(y) \leq n-2$. So y is also a YES instance of $\mathsf{MK}^t\mathsf{P}[n-2]$. Recall that by Equation 2, with probability $1 - \frac{1}{n}$, $y \in Q_\beta^{t_1} \subseteq Q_\beta^t$ (since $t(n) \geq t_1(n)$) and thus $M(y)$ will output 0 (since M is a co-non deterministic algorithm). We turn to considering $y \leftarrow \mathcal{U}_n$. By a standard counting argument, with probability $\geq 1/2$, $K^t(y) \geq n-1$, and with probability $\geq 1 - O(\frac{1}{n}) \geq 0.9$, $K(y) \geq n - \log n$. When both events hold, $M(y)$ will output 1. Combining the above two arguments, we conclude that M breaks the PRG G_n, which is a contradiction since M runs in time T and T is a polynomial. ■

7 Impossibility of Fully Black-box Constructions

In this section, we will show that there is no fully black-box construction of OWFs from hardness of $\mathsf{MK}^t\mathsf{P}|_Q$. In a fully black-box construction, the construction "treats" Kolmogorov complexity in a black box fashion: When considering Kolmogorov complexity, we usually fix a universal Turing machine. In this section, we also consider "black-box" universal Turing machines and Kolmogorov complexity defined w.r.t. these machines. For any function $U : \{0,1\}^* \times 1^{\mathbb{N}} \to \{0,1\}^* \cup \perp$, we say that U is a *black-box universal Turing machine (black-box UTM)* if

- (Universality) Informally, this requires that U simulates "valid programs" correctly: There exists a standard universal Turing machine U_0 such that for any $(M, 1^t)$, if M is a valid description of a Turing machine (w.r.t U_0), $U(M, 1^t)$ outputs what M outputs after t steps.
- (Any "program" has a unique output) For any $M \in \{0,1\}^*, t_1, t_2 \in \mathbb{N}, t_1 \leq t_2$, if $U(M, 1^{t_1}) \neq \perp$, $U(M, 1^{t_2}) = U(M, 1^{t_1})$.

We remark that the above definition is black-box in the following two ways: (1) U is defined to be a function; (2) U is allowed to assign the output of invalid "programs" with an arbitrary string.

For any black-box UTM U, we can define Kolmogorov complexity with respect to U. Formally, for any black-box universal Turing machine U, any string $x \in \{0,1\}^*$, let

$$K_U(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| | \exists t \in \mathbb{N}, U(\Pi, 1^t) = x\}$$

In addition, we can define K_U^t, $\mathsf{MK}_U^t\mathsf{P}$, and event $Q_{U,\beta}^t$ in an analogous way. (Also notice that Fact 2.5 holds no matter which U we pick.)

We turn to introducing the notion of fully black-box construction. Roughly speaking, the construction is a efficient function that takes U as an oracle, and it should be one-way if $\mathsf{MK}_U^t\mathsf{P}[s]|_{Q_\beta^t}$ is hard. The later is captured by having a black-box reduction R such that if any oracle $\mathcal{O}$ breaks the construction, $R^{\mathcal{O},U}$ decides $\mathsf{MK}_U^t\mathsf{P}[s]|_{Q_\beta^t}$.

Definition 7.1 (Fully Black-box Constructions)**.** *For any polynomial t, any constant $\beta > 0$, and any threshold function $s(\cdot)$, a* fully black-box construction of one-way function from the assumption that $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$ *consists of a deterministic poly-time oracle machine f and a non-uniform poly-time oracle machine R such that the following holds:*

- *For any black-box UTM U, and any oracle $\mathcal{O}$ that inverts f^U almost-everywhere, namely, for all $n \in \mathbb{N}$*
$$\Pr[x \leftarrow \{0,1\}^n : \mathcal{O}(1^n, f^U(x)) \in (f^U)^{-1}(f^U(x))] \geq \frac{1}{2},$$
it holds that $R^{\mathcal{O},U}$ decides $\mathsf{MK}^t_U\mathsf{P}[s]|_{Q^t_{U,\beta}}$ on infinitely many input lengths.

Notice that our definition considers non-uniform reductions that work with almost-everywhere inverters and decide the problem only on infinitely many input lengths. Ruling out such reductions will also rule out probabilistic reductions, and reductions that either work with infinitely-often inverters or succeed almost-everywhere.

We move on to stating the main theorem we aim to prove in this section.

Theorem 7.2 (No Fully Black-box Constructions)**.** *For any $\delta > 0$, any threshold $n^\delta \leq s(n) < n-1$, there exists a constant $\beta > 0$ such that for any polynomials $t(n) \geq 2n$, no fully black-box construction of OWFs from $\mathsf{MK}^t\mathsf{P}[s]|_{Q^t_\beta} \notin \mathsf{ioP/poly}$ exists.*

Proof Overview. Before jumping into the formal proof, let us first sketch the high level ideas behind our proof. Assume for contradiction that there exists a fully black-box construction; let f be the construction, and R the reduction. We will consider the black-box universal Turing machine U that has a random oracle hardwired in it: U simulates any "normal" machines in a standard way, and any oracle machines with the random oracle. We will design an oracle $\mathcal{O}$ such that $\mathcal{O}$ inverts f^U but the problem $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$ will be hard for $R^{\mathcal{O},U}$.

To achieve this, we let $\mathcal{O}$ select a random "slice" of the random oracle, parameterized by an prefix w_n, and block any query to this slice: $\mathcal{O}$ will invert $f^U(x)$ as long as none of the queries $f^U(x)$ makes to the random oracle have w_n as a prefix. We choose the length of w_n to be at least $\mathsf{poly}\log(n)$ so that the slice parameterized by w_n is relatively "thin", and thus $\mathcal{O}$ will still invert $f^U(x)$ with high probability (since the prefix w_n is hit with negligible probability).

We turn to arguing that $R^{\mathcal{O},U}$ cannot decide $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$. Although the random oracle is no longer random in the eyes of $R^{\mathcal{O},U}$ (since $\mathcal{O}$ depends on it), observe that the slice (in which $\mathcal{O}$ does not help do anything) still remains random. The key idea is to use the random oracle (given by the slice) to construct a OWF (since random functions are one-way [Imp96]), and next use the OWF to get a cond EP-PRF (as in Section 5.1) and finally conclude (as in Section 5.2) the hardness of $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$.

There is seemingly a contradiction here since as we mentioned above, $\mathcal{O}$ is supposed to invert all OWFs. The point, however, is that to specify the OWF, we need to know the prefix w_n—that is, we only get a so-called *auxiliary-input* OWFs. Furthermore, through the proof of [Imp96], the auxiliary-input OWF that we get is also non-uniformly secure; as a consequence, we can use still use the proof in Section 5.1 to get a auxiliary-input cond EP-PRF.

To show that the existence of such a auxiliary-input cond EP-PRF implies hardness of $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$ requires a bit more care. We assume for contradiction that $R^{\mathcal{O},U}$ decides $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$, and our goal is to break the cond EP-PRF using $R^{\mathcal{O},U}$. Recall that the truth table of the cond EP-PRF can be described using the seed x together with the index w_n, so the K^t_U-complexity of it is at most $|x| + |w_n| + O(1)$. In addition, it is also entropy-preserving and has entropy $|x| - O(\log n)$, and thus the K_U-complexity of it is at least $|x| - O(\log n)$ with high probability. However, this is not enough for us since the

computational depth of the truth table could be roughly $|w_n| + O(\log n) = \mathsf{poly}\log(n) > O(\log n)$, and the truth table will not necessarily fall into the promise $Q^t_{U,\beta}$. Our solution is to consider w_n concatenated with the truth table of cond EP-PRF. It is not hard to see that the concatenation should still have K^t_U-complexity $\leq |w_n| + |x| + O(1)$. On the other hand, it has entropy at least $|w_n| + |x| - O(\log n)$ since w_n is random and the cond EP-PRF is entropy-preserving even conditioned on w_n, and we can then rely on a similar proof to that in Section 5.2 to conclude hardness of $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$. We observe that the above proof approach does not generically show that *any* auxiliary input cond EP-PRF implies hardness of $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$; rather, we here rely on the fact that the auxiliary input is Kolmogorov random. (Rafael's Note: *can we show it generically for aux input cond EPRF when the aux input is K-random?)*

Defining U and $\mathcal{O}$. We proceed to introducing the black-box UTM we construct in our impossibility result. We will consider an oracle universal Turing machine U_{oracle} such that for any oracle $\mathcal{O}$, $U^{\mathcal{O}}_{\mathsf{oracle}}$ simulates any oracle machine Π with $\mathcal{O}$. In addition, if Π does not make oracle query, U_{oracle} will also simulate the execution of Π. We will also need a random function F. F is selected as follows. $F = (F_1, F_2, \ldots)$ consists of a random function $F_n : \{0,1\}^n \to \{0,1\}^n$ for each $n \in \mathbb{N}$. Or alternatively,

$$F_n \leftarrow \{f : \{0,1\}^n \to \{0,1\}^n\}$$

Our black-box UTM U is defined as follows. For any $\Pi \in \{0,1\}^*, t \in \mathbb{N}$, let

$$U(\Pi, 1^t) = U^F_{\mathsf{oracle}}(\Pi, 1^t)$$

Observe that U is a valid black-box UTM (since it behaves just like a UTM if Π does not make any oracle query, and each oracle machine has a unique output).

Suppose there exists a fully black-box construction, and let f be the polynomial-time black-box construction (of a OWF) and R be the black-box reduction. We will construct an inverting oracle $\mathcal{O}$ such that $\mathcal{O}$ inverts f^U but the Kolmogorov complexity problem we consider is still intractable under $\mathcal{O}$. Let $k(n) = \log^6(n)$ be a fixed function. $k(n)$ will be a length parameter in our construction. We will also need a (secret) index sequence $W = (w_1, w_2, \ldots)$ where $|w_m| = k(m)$ for each $m \in \mathbb{N}$ in $\mathcal{O}$. Roughly speaking, the inverting oracle $\mathcal{O}$ on input $(1^n, y)$ will truthfully reveal a pre-image of y, x, only if $f^U(x)$ does not make oracle query on any x' to F (via U) with x' having $w_{|x'|}$ as a prefix. The index sequence W will be also picked randomly. For any $m \in \mathbb{N}$, we let

$$w_m \leftarrow \{0,1\}^{k(m)}$$

We are now ready to construct our inverting oracle $\mathcal{O}$. The oracle $\mathcal{O}$, on input $(1^n, y)$, will enumerate all $x \in \{0,1\}^n$ and simulate $f^U(x)$. For each oracle query $(\Pi, 1^t)$ that $f^U(x)$ makes to U, $\mathcal{O}$ will simulate Π^F, and censor oracle queries Π^F made to F. For each oracle query z by Π^F, let $m = |z|$:

- If $m^{\log m} < n$ (that is, z is a string that is too short), $\mathcal{O}$ will provide $F(z)$ as the answer.
- (censoring) If $m^{\log m} \geq n$, roughly speaking, $\mathcal{O}$ will block the query if z has a prefix of length $k(m)$ that is identical to w_m: $\mathcal{O}$ will ouput $\perp$ if $[z]_{k(m)} = w_m$. Otherwise, it will return $F(z)$.

The simulation of $f^U(x)$ will output $\perp$ if it queries some $(\Pi, 1^t)$ to U and the simulation of Π^F returns $\perp$. Note that $f^U(x)$ runs in polynomial time, so when simulating $f^U(x)$, $\mathcal{O}$ will only look up polynomially many points in F. Finally, if there exists $x \in \{0,1\}^n$ such that the simulation of $f^U(x)$ succeeds and outputs y, $\mathcal{O}(1^n, y)$ will successfully invert f on y (and outputs x). Otherwise it outputs $\perp$.

$\mathcal{O}$ is a good inverter. We first show that for a random choice of F and W, $\mathcal{O}$ will be a good inverting oracle (with high probability). The intuition behind this is that $\mathcal{O}$ will only refuse answering queries that start with certain prefix, and such queries are rare. We will define what it means by w_m being "good" and will show that (1) for all $m \in \mathbb{N}$, with high probability, w_m will be good (formally stated and proved in Lemma 7.3) and (2) if for all $m \in \mathbb{N}$, w_m is good, then $\mathcal{O}$ will be a good inverting oracle (formally stated and proved in Lemma 7.4). We refer to w_m as being "good" if for all $n \leq m^{\log m}$,

$$\Pr[x \leftarrow \{0,1\}^n : f^U(x) \text{ makes a query to } F \text{ on } z, |z| = m, [z]_{k(m)} = w_m] \leq \frac{2^{\log^5 m}}{2^{k(m)}} \tag{3}$$

where we say that f^U makes a query to F on z, we mean f^U does so through making a query to U: f^U makes a query to U on $(\Pi, 1^t)$ and the machine Π makes an query to F on z. If $[z]_{k(m)} = w_m$, $\mathcal{O}$ will block its answer to this query, and the simulate of $f^U(x)$ will fail. Thus, w_m is good if for all $n \leq m^{\log m}$, the probability over random x that $\mathcal{O}$ receives a query that starts with the prefix w_m is small.

Lemma 7.3. *For any choice of F, any $m \in \mathbb{N}$, with probability at least $1 - \frac{1}{m}$ over the choice of w_m, w_m is good.*

Proof: Consider any fixed function F and any fixed $n \leq m^{\log m}$. Since f runs in polynomial time, let $r(n)$ be a polynomial such that f runs in time $r(n)$. We first claim that with probability at most $1/2^{\log^5 m/2}$, a random choice of w_m will be "bad" with respect to input length n. For each $x \in \{0,1\}^n$, $f^U(x)$ makes at most $r(n)$ oracle queries. Therefore, over a random choice of w_m, $f^U(x)$ makes a query to F on some z, $|z| = m$, $[z]_{k(m)} = w_m$ with probability at most

$$\frac{r(n)}{2^{k(m)}} \leq \frac{2^{\log^4 m}}{2^{k(m)}}$$

Thus, there are in expectation at most a $\frac{2^{\log^4 m}}{2^{k(m)}}$ fraction of x on which f^U makes such a query. (We refer to such string x as being "bad".) It follows from the Markov bound that, with probability at most

$$\frac{2^{\log^4 m}}{2^{k(m)}} / \frac{2^{\log^5 m}}{2^{k(m)}} \leq \frac{1}{2^{\log^5 m/2}},$$

over a random w_m, the fraction of bad x of length n is more than $\frac{2^{\log^5 m}}{2^{k(m)}}$. Finally, by taking a union bound (over all input lengths $n \leq m^{\log m}$), we conclude that the probability that w_m is bad is at most

$$\frac{1}{m^{\log^5 m/2}} \cdot m^{\log m} \leq \frac{1}{m}$$

■

Lemma 7.4. *For any choice of F, if for $m \in \mathbb{N}$, w_m is good, then $\mathcal{O}$ inverts f^U. Namely, $\mathcal{O}$ satisfies the condition in Definition 7.1.*

Proof: Consider any $n \in \mathbb{N}$. We will show that $\mathcal{O}(1^n, f^U(x))$ will invert $f^U(x)$ with probability at least $\frac{1}{2}$ over random $x \leftarrow \{0,1\}^n$. For any $x \in \{0,1\}^n$, we refer to x as being "censored" if when $f^U(x)$ being simulated by $\mathcal{O}$, $f^U(x)$ will output $\perp$. Note that this happens if $f^U(x)$ makes a query to F on some z, $m = |z|$ satisfying that $n \leq m^{\log m}$, and $[z]_{k(m)} = w_m$. Let r be the polynomial such that f runs in time $r(n)$. Notice that $f^U(x)$ can make queries to U (and thus to F) of length

at most $r(n)$, w_m is good for all $m \in \mathbb{N}$. By a union bound over all $m \leq r(n)$ such that $m^{\log m} > n$, the probability that $x \leftarrow \{0,1\}^n$ is censored is at most

$$\sum_{m=2^{\sqrt{\log n}}}^{r(n)} \frac{2^{\log^5 m}}{2^{k(m)}} \leq \sum_{m=2^{\sqrt{\log n}}}^{r(n)} \frac{1}{2^{\log^6 m/2}} \leq \sum_{m=2^{\sqrt{\log n}}}^{r(n)} \frac{1}{2^{\log^3 n/2}} \leq \frac{r(n)}{2^{\log^3 n/2}} < \frac{1}{2}.$$

Note that if x is not censored, $\mathcal{O}(1^n, f^U(x))$ will succeed in inverting $f^U(x)$, and therefore $\mathcal{O}$ will succeed with probability at least $\frac{1}{2}$, which concludes that $\mathcal{O}$ is a good inverting oracle. ■

$\mathsf{MK}^t_U\mathsf{P}[s]|_Q$ is still hard. We turn to proving that $\mathsf{MK}^t_U\mathsf{P}[s]|_Q$ is still (non-uniformly) hard even in the presence of $\mathcal{O}$. We rely on the fact that (exponentially hard) OWFs exist relative to a random oracle [Imp96]. For any oracle function f and any oracle $\mathcal{O}'$, we say that $f^{\mathcal{O}'}$ is a *OWF relative to* $\mathcal{O}'$ if $f^{\mathcal{O}'}$ is a OWF when the attacker also has oracle access to $\mathcal{O}'$. (We can define this analogously for any other crypto primitives, like PRGs, PRFs.) In the rest of this section, we are interested in functions that are *only defined on a particular input length*. We refer to such functions as being a OWF (or PRG, PRF, etc) if they are one-way (or pseudorandom) on the input length they are defined, and in the security game we will consider non-uniform attackers.

Theorem 7.5 ([Imp96])**.** *Let F be the random oracle defined as above. For any oracle $\mathcal{O}'$, there exist a constant $b > 0$ and a polynomial time oracle algorithm g such that for all $n \in \mathbb{N}$, all $\Omega(\log n) \leq \ell \leq n$, with probability $1 - 2^{-b\ell}$ over F_n, $g^{F_n}(1^\ell) : \{0,1\}^\ell \to \{0,1\}^\ell$ is a non-uniformly $2^{b\ell}$-hard OWF (only defined on input length ℓ) relative to $F_n, \mathcal{O}'$.*

We cannot use this result directly since F_n is no longer a random function given $\mathcal{O}$. Notice that when being restricted to the inputs starting with w_n, F_n behaves just like a random function even in the presence of $\mathcal{O}$ (since they are independent). Combining this observation together with Theorem 7.5 and the cond EP-PRF construction from OWF (see Section 5), we will obtain the following theorem.

Lemma 7.6. *Let $F, U, \mathcal{O}$ be as above. For any $0 < \delta < 1$, there exists a rate-1 efficient oracle algorithm h such that for all sufficiently large $n \in \mathbb{N}$, all $n^\delta \leq \ell \leq n - k(n)$, with probability $1 - \frac{1}{n}$ over F_n, $h^{F_n}(\ell, n, w_n) : \{0,1\}^\ell \times [n - k(n)] \to \{0,1\}$ is a non-uniformly secure $(n^{\log n}, 0.01)$-cond 0.01-EP-PRF (defined only over input length ℓ) relative to U and $\mathcal{O}$.*

Proof: Consider any $0 < \delta < 1$. For any sufficiently large $n \in \mathbb{N}$, let $F_n|w_n$ denote the random oracle F_n after being restricted to inputs starting with w_n. Let $N = n^{\log n}$, and consider the oracle $\mathcal{O}_N$ that refuses to answer any oracle query of length $\geq N$. It follows from our construction of $\mathcal{O}$ and $n^{\log n} \leq N$ that $\mathcal{O}_N$ will not reveal information in $F_n|w_n$, so $F_n|w_n$ is still a random oracle given $\mathcal{O}_N$.

We are now ready to construct our cond EP-PRF. Our construction proceeds as follows.

- (Getting a OWF.) Let g be the oracle algorithm, b be the constant as in Theorem 7.5. By Theorem 7.5, for any $n^\delta \leq \ell \leq n - k(n)$, with probability at least $1 - 2^{-b\ell}$, $g^{F_n|w_n}(1^\ell)$ is a non-uniformly $2^{b\ell}$-hard OWF (defined on input length ℓ) relative to U, $\mathcal{O}_N$, and $F_n|w_n$. (Note that $g^{F_n|w_n}(1^\ell)$ has hardness exponential in its input length.)

- (Obtaining a cond EP-PRF.) Given this exponential hard OWF, by Theorem 5.2, there exists a rate-1 efficient oracle algorithm h_0, such that for any $n^\delta \leq \ell \leq n - k(n)$, with probability $1 - 2^{-b\ell}$ over F_n, $h_0^{F_n|w_n}(1^\ell) : \{0,1\}^\ell \times [\ell^{1/\delta}]$ is a non-uniformly secure $(n^{\log n}, 0.01)$-cond 0.01-EP-PRF relative to U and $\mathcal{O}_N$. (Notice that this PRF has only quasi-poly hardness and its truth table is of length polynomial in its seed.)

- (Post processing.) Note that $\ell^{1/\delta} \geq (n^{\delta})^{1/\delta} \geq n$, by truncating the function $h_0^{F_n|w_n}(1^{\ell})$ properly and simulating oracle access to $F_n|w_n$ by using F_n and w_n, we obtain that there exists h such that $h^{F_n}(\ell, n, w_n) : \{0,1\}^{\ell} \times [n - k(n)] \to \{0,1\}$ is a non-uniformly secure $(n^{\log n}, 0.01)$-cond 0.01-EP-PRF (defined only on input length ℓ) relative to U and $\mathcal{O}_N$.

Finally, notice that no $n^{\log n}$-time attacker can distinguish $\mathcal{O}_N$ from $\mathcal{O}$ (since $n^{\log n}$-time attacker can only make queries of length $\leq n^{\log n} \leq N$), so we can switch $\mathcal{O}_N$ to $\mathcal{O}$ which concludes the proof. ■

We now use the existence of a cond EP-PRF relative to U and $\mathcal{O}$ to deduce the hardness of $\mathsf{MK}^t_U\mathsf{P}$ w.r.t attackers having access to U and $\mathcal{O}$. Technically, the cond EP-PRF we use is only an *auxiliary-input* cond EP-PRF (which is only secure when the construction gets additionally w_n as input), but we will show that such a cond EP-PRF suffices to deduce the hardness of $\mathsf{MK}^t_U\mathsf{P}$.

Lemma 7.7. *For any constant $\delta > 0$, any threshold function $n^{\delta} \leq s(n) < n - 1$, there exists a constant $\beta > 0$ such that for any polynomial $t(n) \geq 2n$, for all sufficiently large $n \in \mathbb{N}$, with probability 0.05 over F_n and w_n, $\mathsf{MK}^t_U\mathsf{P}[s]|_{Q^t_{U,\beta}} \cap \{0,1\}^n \notin \mathsf{SIZE}[n^{0.1 \log n}]^{U,\mathcal{O}}$ where $U, \mathcal{O}, F$ are as above.*

Proof: Consider any δ, s as in the lemma statement. Let $\beta = 8/\delta$ be a constant. Consider any polynomial $t(n) \geq 2n$. Let h be the oracle $(n^{\log n}, 0.01)$-cond 0.01-EP-PRF construction guaranteed to exist by Lemma 7.6 (by taking δ in Lemma 7.6 to be $\delta/2$). Consider any sufficiently large $n \in \mathbb{N}$.

Assume for contradiction that there exists a non-uniform deterministic $n^{0.1 \log n}$-time algorithm M that, given oracle access to U and $\mathcal{O}$, decides $\mathsf{MK}^t_U\mathsf{P}[s]|_{Q^t_{U,\beta}}$ on input length n. Let $\ell = s(n) - k(n) - 2 \log n$, and let $s = s(n), k = k(n)$. We consider the following function $f : \{0,1\}^{\ell} \times [n - k] \to \{0,1\}$ defined as

$$f \stackrel{\text{def}}{=} h^{F_n}(\ell, n, w_n)$$

(Notice that f has a fixed seed length, ℓ.) Since $\ell = s - k - 2 \log n \geq n^{\delta} - k - 2 \log n \geq n^{\delta/2}$, by Lemma 7.6, f is a non-uniformly secure $(n^{\log n}, 0.01)$-cond 0.01-EP-PRF (with probability at least $1 - \frac{1}{n}$ over F_n). Fix some such good choice of F_n (which happens with probability $1 - \frac{1}{n}$). Let E_{ℓ} be the event associated with the cond EP-PRF f. We will next use the $\mathsf{MK}^t_U\mathsf{P}$ algorithm M to construct a distinguisher D and break the cond EP-PRF f.

Our distinguisher D, given access to a function $f' : [n - k] \to \{0,1\}$ where f' is either a random function or a pseudorandom function, proceeds as follows. D picks w_n (of length k) as non-uniform advice, and then D queries all entries of f' to obtain the truth table $\mathsf{tt}(f')$. Finally, D feeds $w_n || \mathsf{tt}(f')$ (of length $k + n - k = n$) to M and simply outputs $M(w_n || \mathsf{tt}(f'))$. Note that D runs in time $O(n^{0.1 \log n}) + |k| \leq n^{\log n}$.

We turn to showing that D will distinguish the cond EP-PRF from random functions. We rely on the following claims.

Claim 5. *With probability 0.2 over the choice of w_n, $K_U(w_n || \mathsf{tt}(f')) \geq n - 1$ holds with probability ≥ 0.2 over f' when f' is a random function $f' : [n - k] \to \{0,1\}$.*

Proof: By a standard counting argument, it follows that over a random choice of w_n and f', $K_U(w_n || \mathsf{tt}(f')) \geq n - 1$ holds with probability 0.5. By an averaging argument, for a 0.2 fraction of w_n, the probability that $K_U(w_n || \mathsf{tt}(f')) \geq n - 1$ is at least 0.2 over random choice of f'. ■

Claim 6. *$K^t_U(w_n || \mathsf{tt}(f')) \leq s$ holds with probability 1 over f' when $f' \leftarrow f(\mathcal{U}_{\ell} | E_{\ell})$.*

Proof: Consider any $f' \leftarrow f(\mathcal{U}_{\ell} | E_{\ell})$ and let x be its seed. (Recall that $f(\cdot) = h^{F_n}(\ell, n, w_n)(\cdot)$, so $f' = f(x)$.) We will construct a program Π^F that produces the string $w_n || \mathsf{tt}(f')$. Π will hardwire

the string w_n (of length k), the seed x (of length $\ell = s - k - 2\log n$), together with the integer n, the descriptions of $k(\cdot)$, $s(\cdot)$, and the code of h (using no more than $2\log n$ bits). Π^F will first get n from its description, compute $k = k(n)$ and $\ell = s(n) - k(n) - 2\log n$. It will further read w_n of length k and x of length ℓ from its tape. It will then simulate $f(x) = h^{F_n}(\ell, n, w_n)(x)$ and answer oracle query made by $f(x)$ using F. Since each bit in f' (given the seed x and w_n as inputs) can be printed in time $t(\ell)$ (since f is rate-1 efficient) and Π can trivially output each bit in w_n, it follows that $K^t_U(w_n||\mathsf{tt}(f')) \leq k + s - k - 2\log n + 2\log n \leq s$. ∎

Claim 7. *With probability 0.9 over the choice of w_n, $w_n||\mathsf{tt}(f') \in Q^t_{U,\beta}$ holds with probability 0.9 over f' when $f' \leftarrow f(\mathcal{U}_\ell|E_\ell)$.*

Proof: Let $X = W_n||\mathsf{tt}(f(\mathcal{U}_\ell|E_\ell))$ where W_n is the random variable of w_n and $\mathsf{tt}(f(\mathcal{U}_\ell|E_\ell))$ is the random variable of the truth table of f'. We first show that X will have relatively high (time-unbounded) K_U-complexity. Note that f is 0.01-entropy-preserving, we have that

$$H(\mathsf{tt}(f(\mathcal{U}_\ell|E_\ell)) \mid W_n) \geq \ell - 0.01\log\ell.$$

Therefore, the random variable X has entropy at least

$$H(W_n) + H(\mathsf{tt}(f(\mathcal{U}_\ell|E_\ell)) \mid W_n) \geq \ell + k - 0.01\log\ell.$$

Let

$$S = \{z \in \{0,1\}^n : z \in X\}$$

be the support of X. Since $H(X) \geq \ell + k - 0.01\log\ell$, it follows that $2^{\ell+k-0.01\log\ell} \leq |S|$. On the other hand, it holds that $|S| \leq |\mathsf{supp}(W_n)| \times |\mathsf{supp}(\mathcal{U}_\ell)| \leq 2^{\ell+k}$. Let

$$Z = \{z \in S : K_U(z) \leq \ell + k - 2\log\ell\}$$

be the set of strings in the support but having small K_U-complexity. By a standard counting argument w.r.t. K_U-complexity, we have that $|Z| \leq 2^{\ell+k-2\log\ell+1}$. By Lemma 2.6, it follows the probability that $X \in Z$ is at most

$$\Pr[X \in Z] \leq \frac{\log|S| + 1 - H(X)}{\log|S| - \log|Z|} \leq 0.01$$

In other words, with probability at least 0.99 over choice of f' and w_n, it holds that

$$K_U(w_n||\mathsf{tt}(f')) \geq \ell + k - 2\log\ell \geq s - 2\log n - 2\log\ell \geq s - 4/\delta\log s. \tag{4}$$

If the above inequality holds, it follows that

$$K^t_U(w_n||\mathsf{tt}(f')) - K_U(w_n||\mathsf{tt}(f')) \leq s - K_U(w_n||\mathsf{tt}(f')) \leq 4/\delta\log s \leq \beta\log K_U(\mathsf{tt}(f')||w_n).$$

where the first inequality follows from Claim 6, and the last inequality follows from our choice of β. Thus, we have that if Equation 4 holds, then $w_n||\mathsf{tt}(f') \in Q^t_{U,\beta}$. Finally, by an averaging argument, it holds that with probability at least 0.9 over choice of w_n, Equation 4 holds with probability at least 0.9 over randomness of $\mathsf{tt}(f')$, which completes the proof for this claim. ∎

Conditioned on some w_n that satisfies all the above three claims (which by a union bound happens with probability at least $0.2-0.1 \geq 0.1$), by claim 5, the distinguisher D will output 1 with probability at most $1-0.2$ when f' is a random function. By claim 6 and 7, D will output 1 with probability at least 0.9 when f' is a pseudorandom function. So D distinguishes the cond EP-PRF f from random functions with advantage $0.1 > 0.01$, which is a contradiction.

Finally, we notice that for this proof to work, the choice of F_n we need will be sampled with probability $1 - \frac{1}{n}$, and the choice of w_n we need will be sampled with probability 0.1. By a union bound, the statement will hold with probability at least 0.05 over the choice of F_n and w_n. ∎

Return to the proof of Theorem 7.2. In the end of the section, we return to the proof of Theorem 7.2.

Proof: For any t, s as in the theorem statement, let β be the constant as in Lemma 7.7. Suppose that there exists a fully black-box construction (f, R), let $U, F, \mathcal{O}$ be as above.

For each $n \in \mathbb{N}$, by a union bound, a random choice of F_n and w_n will satisfies the conditions in Lemma 7.3 and Lemma 7.7 with probability ≥ 0.01. Fix some such F_n and w_n for each $n \in \mathbb{N}$. It follows from Lemma 7.4 that $\mathcal{O}$ is a good inverting oracle, and therefore the reduction R with oracle access to U and $\mathcal{O}$ should decide $\mathsf{MK}^t_U\mathsf{P}[s]|_{Q^t_{U,\beta}}$, which contradicts to Lemma 7.7. ■

8 Acknowledgements

We thank the anonymous FOCS reviewers for their helpful comments.

References

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

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 284–293. ACM, 1997.

[AF09] Luis Antunes and Lance Fortnow. Worst-case running times for average-case algorithms. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 298–303. IEEE, 2009.

[AFvMV06] Luis Antunes, Lance Fortnow, Dieter van Melkebeek, and N Variyam Vinodchandran. Computational depth: concept and applications. *Theoretical Computer Science*, 354(3):391–404, 2006.

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

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

[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.

[BDV17] Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In *Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part I*, pages 696–723. Springer, 2017.

[BI87] Manuel Blum and Russell Impagliazzo. Generic oracles and oracle classes. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 118–126. IEEE, 1987.

[BLMP23] Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. Kolmogorov comes to cryptomania: On interactive kolmogorov complexity and key-agreement. 2023.

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

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

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

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

[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.

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

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

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

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

[GKLO22] Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C Oliveira. Probabilistic kolmogorov complexity with applications to average-case complexity. In *37th Computational Complexity Conference (CCC 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

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

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

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

[HIL+23] Shuichi Hirahara, Rahul Ilango, Zhenjian Lu, Mikito Nanashima, and Igor C Oliveira. A duality between one-way functions and average-case symmetry of information. *Cryptology ePrint Archive*, 2023.

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

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

[Hir22] Shuichi Hirahara. Np-hardness of learning programs and partial mcsp. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 968–979. IEEE, 2022.

[HN23] Shuichi Hirahara and Mikito Nanashima. Learning in pessiland via inductive inference. 2023.

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

[Ila20] Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and AC^0[p]. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, pages 34:1–34:26, 2020.

[Ila21] Rahul Ilango. The minimum formula size problem is (eth) hard. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–432. IEEE, 2021.

[Ila22] Rahul Ilango. Constant depth formula and partial function versions of mcsp are hard. *SIAM Journal on Computing*, (0):FOCS20–317, 2022.

[ILO20] Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization for multi-output functions. In *35th Computational Complexity Conference, CCC 2020*, pages 22:1–22:36, 2020.

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

[Imp96] Russell Impagliazzo. Very strong oneway functions and pseudorandom generators exist relative to a random oracle. Manuscript, 1996.

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1575–1583, 2022.

[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.

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

[LOZ22] Zhenjian Lu, Igor C Oliveira, and Marius Zimand. Optimal coding theorems in time-bounded kolmogorov complexity. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

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

[LP21a] Yanyi Liu and Rafael Pass. Cryptography from sublinear time hardness of time-bounded kolmogorov complexity. In *STOC*, 2021.

[LP21b] Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on EXP $\neq$ BPP. In *CRYPTO*, 2021.

[LP22a] Yanyi Liu and Rafael Pass. Leakage-resilient hardness vs randomness. *Electronic Colloquium on Computational Complexity*, 2022. https://eccc.weizmann.ac.il/report/2022/113/.

[LP22b] Yanyi Liu and Rafael Pass. On one-way functions from np-complete problems. In *Proceedings of the 37th Computational Complexity Conference*, pages 1–24, 2022.

[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.

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

[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.

[Reg04] Oded Regev. New lattice based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004.

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

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

[Rud88] Steven Rudich. *Limits on the provable consequences of one-way functions*. University of California at Berkeley, 1988.

[Rud97] Steven Rudich. Super-bits, demi-bits, and np/qpoly-natural proofs. In *Randomization and Approximation Techniques in Computer Science: International Workshop RANDOM'97 Bologna, Italy, July 11–12, 1997 Proceedings 1*, pages 85–93. Springer, 1997.

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

[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 Cond EP-PRF from OWFs

We will show how to construct a cond EP-PRF from a OWF, and the cond EP-PRF we obtain will have a polynomial-time black-box security reduction from a PRG and a PRF. We recall the notion of a cond EP-PRG [LP20].

Definition A.1. *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 α-entropy-preserving pseudorandom generator ((T, ε)-cond α-EP-PRG) *if there exists a sequence of events* $= \{E_n\}_{n\in\mathbb{N}}$ *such that the following conditions hold:*

- **(pseudorandomness):** *For every probabilistic $T(n)$-time attacker $\mathcal{A}$ and sufficiently large $n \in \mathbb{N}$, $\{g(\mathcal{U}_n|E_n)\}_{n\in\mathbb{N}}$ and $\{\mathcal{U}_{n+n^\xi}\}_{n\in\mathbb{N}}$ are $(T(n), \varepsilon(n))$-indistinguishable;*
- **(entropy-preserving):** *For all sufficiently large $n \in \mathbb{N}$, $H([g(\mathcal{U}_n|E_n)]_n) \geq n - \alpha \log n$.*

We refer to the constant α as the entropy-loss constant. *We say that g is non-uniformly secure if the pseudorandomness condition holds w.r.t. all non-uniform T-time attackers. We refer to g as a ε-cond α-EP-PRG if g is secure w.r.t. all* PPT *attackers.*

Note that we can define a polynomial-time black-box security reduction for a cond EP-PRG to any PRG in the same way as we define this for a cond EP-PRF in section 5.3.

In [LP21a], they showed that a cond EP-PRG can be constructed from a PRG whose security is based on a poly-time black-box reduction.

Lemma A.2 ([LP21a])**.** *Let $\varepsilon = 0.1$, $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exists $T_1 \in \mathcal{F}$ such that T_1-one-way functions exist. Then there exist $T_2 \in \mathcal{F}$, $T_2 \leq T_1$, constants $\alpha > 0, \xi < 1$, and a $(T_2(m), \varepsilon/4)$-cond α-EP-PRG $G : \{0,1\}^m \to \{0,1\}^{m+m^\xi}$ with a poly-time black-box security reduction to a PRG g.*

In addition, this lemma also holds in the non-uniform setting.

Notice that Lemma A.2 also holds in the polynomial hardness setting.

Lemma A.3 ([LP21a])**.** *Let $\varepsilon = 0.1$. Assume that OWFs exist. Then, there exist constants $0 < \xi < 1, \alpha > 0$ and a $(\mathsf{poly}, \varepsilon/4)$-cond α-EP-PRG $G : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$ with a poly-time black-box security reduction to a PRG g.*

In addition, this lemma also holds in the non-uniform setting.

We next construct a desired cond EP-PRF from a cond EP-PRG (together with some padding arguments that are needed). Our security reduction is black-box.

Lemma A.4. *Let $\varepsilon = 0.1$. Assume there exist a constant $\alpha > 0$, a $(T_1(n), \varepsilon/4)$-cond α-EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+n^\xi}$, $0 < \xi < 1$, with a poly-time black-box security reduction to a $(T_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-PRG f_{prg} and a $(T_1(n), \varepsilon/4)$-PRF $h : \{0,1\}^n \times [T_2(n)] \to \{0,1\}$. Then, for any constant $0 < \beta < \alpha$, there exist a constant $0 < \theta < 1$ and a rate-1 efficient $(T_1(n^\theta), \varepsilon)$-cond β-EP-PRF $f : \{0,1\}^n \times [T_2(n^\theta)] \to \{0,1\}$ with a poly-time black-box security reduction to the PRG f_{prg} and the PRF h.*

In addition, this lemma also holds in the non-uniform setting.

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 $x \in \{0,1\}^n$. 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 both rate-1 efficient and of small entropy loss. We will need a constant c_1 (defined as follows) to parameterize the padding argument. Note that both g and h are efficient, so let c_1 be a constant such that g, h run in time n^{c_1}. Notice that we want to construct a β-EP-PRF from the α-EP-PRG g ($\alpha > \beta$), the padding trick will also be parameterized by α and β.

Now we proceed to a formal construction. Let $\theta = \xi/(2c_1) \cdot \beta/\alpha$ be the padding parameter. Let $m = n^{2c_1\alpha/\beta} = n^{\xi/\theta}$. Consider the function $f : \{0,1\}^m \times [T_2(n^\xi)] \to \{0,1\}$ defined as the following:

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

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 $n = m^{1/(2c_1\alpha/\beta)}$ 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 $T_2(n^\xi) = T_2(m^{\xi/(2c_1\alpha/\beta)}) = T_2(m^\theta)$. Thus, f indeed maps $\{0,1\}^m \times [T_2(m^\theta)]$ to $\{0,1\}$.

We first show that f is entropy-preserving with entropy loss β. 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,

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

where α is the entropy-loss constant of g.

We then show that f is pseudorandom by a standard hybrid argument. For any $T_1(m^\theta)$-time adversary $\mathcal{A}$ and all sufficiently large m, let Real denote the quantity $\Pr[x \leftarrow \{0,1\}^m; \mathcal{A}^{f(x,\cdot)}(1^m) = 1|E_m]$, 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

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

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 $|\mathsf{Real} - \mathsf{Ideal}| < \varepsilon$ and thus f satisfies the pseudorandomness property.

Claim 8. $|\mathsf{Real} - \mathsf{Hybrid}| < \varepsilon/4$.

Proof: This claim follows from the fact that g is pseudorandom w.r.t. all $T_1(n)$-time attackers. Note that by our choice of parameters, $\mathcal{A}$ runs in time $T_1(m^\theta) \leq T_1(n)$. ■

Claim 9. $|\mathsf{Hybrid} - \mathsf{Ideal}| < \varepsilon/4$.

Proof: This claim follows immediately from the fact that h is a $(T_1(\ell), \varepsilon/4)$-pseudorandom function (where ℓ is the input length of h) and note that by our choice of parameters, $\ell = n^\xi$. Note that $\mathcal{A}$ runs in time $T_1(m^\theta) = T_1(\ell)$. ■

We turn to showing 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 $O(n^{c_1})$, and the running time of f depends on i: If $i \leq n$, f runs in time $O(n^{c_1}) \leq O(\sqrt{m})$. If $n < i \leq m$, 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 $O(n^{c_1}) + O(n^{\xi c_1}) \leq O(\sqrt{m})$. Thus, f runs in time $m + O(\log m)$.

We finally notice that this reduction is a polynomial-time black-box security reduction to g and h. The specs (the input length function, security loss function, and reduction) needed in the black-box reduction can be found in the above proof. Since g has a poly-time black-box security reduction to the PRG f_{prg}, we conclude that f has a poly-time black-box reduction to f_{prg} and h. We also observe that the above proof also works in the non-uniform setting. ■

Now we are ready to show the cond EP-PRF construction with black-box security reductions to OWFs.

Theorem A.5. *The following statements hold.*

- *Let $\mathcal{F}$ be a nice class of super-polynomial functions. Assume that there exist $T \in \mathcal{F}$, and a T-hard OWF. Then, for any constant $\delta > 1$, $\beta > 0$, there exist a function $T_1 \in \mathcal{F}$ and a rate-1 efficient $(T_1^{\delta}, 0.1)$-cond β-EP-PRF $f : \{0,1\}^n \times [T_1(n)] \to \{0,1\}$ with a poly-time black-box security reduction to a PRG and a PRF.*
- *Assume that OWFs exist. Then, for any constant $\beta > 0$, any polynomial $d(\cdot)$, there exists a rate-1 efficient 0.1-cond β-EP-PRF $f : \{0,1\}^n \times [d(n)] \to \{0,1\}$ with a poly-time black-box security reduction to a PRG and a PRF.*

In addition, the above statements also hold in the non-uniform setting.

Proof: (a): This statement follows from Lemma A.4, Lemma A.2, and the fact that we can get a $\mathcal{F}$-hard PRF from a $\mathcal{F}$-hard OWF [GGM84]. Also note that all the these results still hold in the non-uniform setting.

(b): By [GGM84], for any polynomial $d'(\cdot)$, there exists a PRF $h : \{0,1\}^n \times [d'(n)] \to \{0,1\}$. This statement then follows from Lemma A.4 by considering T_1 being any sufficiently large polynomial and T_2 being any arbitrary polynomial. Also note that all the results needed in this proof still hold in the non-uniform setting. ■