Electronic Colloquium on Computational Complexity, Report No. 51 (2026)

Cryptographic Implications of Worst-Case Hardness of Time-Bounded Kolmogorov Complexity

Yanyi Liu* Noam Mazor† Rafael Pass‡

April 6, 2026

Abstract

We consider the worst-case hardness of the gap version of the classic time-bounded Kolmogorov complexity problem—$\mathrm{Gap_pMK^tP}[s_1, s_2]$—where the goal is to determine whether for a given string x, $\mathrm{K}^t(x) \leq s_1(n)$ or $\mathrm{K}^{p(t)}(x) > s_2(n)$, where $\mathrm{K}^t(x)$ denotes the t-bounded Kolmogorov complexity of x. As shown by Hirahara (STOC'18), if $\mathrm{Gap_pMK^tP}[s_1, s_2] \notin \mathrm{prBPP}$ for every polynomial p, then (under appropriate derandomization assumption) $\mathrm{Gap_pMK^tP}$ is *errorless* average-case hard with respect to BPP heuristics. The notion of errorless average-case hardness, however, is seemingly insufficient for cryptographic applications where one needs to consider average-case hardness against attacks that simply may err with some probability (i.e., two-sided error hardness).

In this work, we present several new consequences of the assumption that $\mathrm{Gap_pMK^tP}[s_1, s_2] \notin \mathrm{P/poly}$ for all polynomials p, for appropriate choices of s_1, s_2, and under appropriate (worst-case) derandomization assumptions. In particular, we show that this assumption implies:

- The existence of an (inefficient-prover) *zero-knowledge proof system* for NP with a non-uniform simulator w.r.t. adversaries with a-priori bounded-length auxiliary-input.
- The existence of a hard disjoint NP pair, defined as a promise problem $(\mathcal{Y}, \mathcal{N})$ where both $\mathcal{Y}, \mathcal{N} \in \mathrm{NP}$; this provides a barrier towards showing that $\mathrm{Gap_pMK^tP}$ is NP-complete.

The above results are proven via first showing that the above assumption implies the existence of a so-called *conditional PRG*—roughly speaking, a cryptographic PRG where indistinguishability only needs to hold for some (potentially not efficiently sampleable) distribution over the seed to the PRG. (In fact, this notion of a PRG also almost directly implies average-case hardness of $\mathrm{Gap_pMK^tP}$, and as such, this provides a modular explanation to Hirahara's results.)

Finally, we show that for the results on conditional PRGs and Zero-knowledge Proofs, unconditional results can be obtained (that is, without making any derandomization assumptions), if considering an appropriate version of $\mathrm{Gap_pMK^tP}$ concerning *randomized* K^t.

*Cornell Tech. E-mail: yl2866@cornell.edu. Research partly supported by NSF CNS-2149305.

†New York University. E-mail: noammaz@gmail.com. Research was partly done while visiting the Simons institute.

‡Technion, Cornell Tech and Tel Aviv University. E-mail: rafael@cs.cornell.edu. Supported in part by AFOSR Award FA9550-23-1-0387, AFOSR Award FA9550-23-1-0312, AFOSR Award FA9550-24-1-0267, ISF Award 2338/23 and ERC Advanced Grant KolmoCrypt - 101142322. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, or the AFOSR.

ISSN 1433-8092

Contents

1 Introduction

The *Time-Bounded Kolmogorov Complexity Problem* [Kol68; Sol64; Cha69; Ko86; Har83; Sip83]—that is, determining the length, denoted $\mathrm{K}^t(x)$ of the shortest program (evaluated on some particular Universal Turing machine U) that generates a given string x, within time t, where $t = poly(|x|)$ is a polynomial—is one of the oldest and most classic computational problems; see [Tra84] for historical accounts of the study of it (and related problems). In essence, the time-bounded Kolmogorov complexity problem captures the fundamental question of understanding how well a string x can be compressed if requiring that the compressed version can be *efficiently decompressed* (in time $t(|x|)$). As such, it also provides a clean mathematical formalization of Occam's razor (i.e., the question of finding the "best explanation" for some given data).

Given the fundamental nature of this problem, one would expect it to be NP-complete, but perhaps surprisingly this question remains open: Indeed, a major problem since the 1960s is whether a decisional version of time-bounded Kolmogorov complexity, or gap-versions of it, are hard, and in particular whether they are NP-complete (the problem is trivially in NP). In the recent decades, however, there has been a lot of amazing progress on this question—providing evidence pointing towards both a positive and a negative answer; we review these results in Section 1.3, but here simply note that all positive results either consider NP-complete *variants* of these problems that in some fundamental ways have a *different structure* (despite perhaps a syntactic similarity), and that the negative results (i.e., indications against NP hardness) only apply to limited types of reductions. The question, essentially, remains as wide open as when it was first formulated.[1]

Beyond being interesting on its own, the time-bounded Kolmogorov complexity problem is endowed with some intriguing properties:

- **Worst-case to *Errorless* Average-case Hardness** As demonstrated by Hirahara [Hir18], worst-case hardness of a gap version of the problem, $\mathrm{Gap_pMK^tP}[s_1, s_2]$ w.r.t. any s_1, s_2 that are sufficiently far apart—where the goal is to determine whether the $\mathrm{K}^t(x) \leq s(n)$ or $\mathrm{K}^{p(t)}(x) > s_2(n)$— w.r.t *any* polynomial p, implies *errorless* average-case hardness.[2] In other words, there exists a worst-case to *errorless* average-case reduction for the problem. Of course, such reductions are known for other problems (even yielding two-sided error average-case hardness), but all those problems are significantly more structured (e.g. [AD97; Reg09; DH76; ElG84]. In particular, they all sit within $\mathrm{AM} \cap \mathrm{coAM}$, whereas the above problem is not under some reasonable assumption [Hir18]). Arguably, it is also the simplest/cleanest problem whose worst-case hardness implies average-case hardness on the uniform distribution.

- **Equivalence to OWFs, in the Regime of Two-sided Error Average-case** Additionally, as shown by [LP20], mild (two-sided error) average-case hardness on the uniform distribution of even just the plain K^t-problem (i.e., even just computing $\mathrm{K}^t(x)$) is *equivalent* to the existence of cryptographic *one-way functions*, the central object in complexity-based cryptography: A one-way

[1] One may argue that the results of [MP24] shows that at least according to Levin's original definition of NP-hardness (w.r.t. so-called "Levin reductions"), the question has been answered at least under some strong cryptographic hardness assumptions, and in some parameter regimes. But more modern definitions of NP-hardness allow for a more relaxed notion of a reduction (that of a Karp reduction), and for such reduction, limitations do not apply.

[2] Errorless average-case hardness requires hardness against all algorithms that either provide the correct answer or answer $\perp$, and furthermore only answer $\perp$ with small probability over the instances. This is a (seemingly) weaker notion of average-case hardness than the standard notion that requires hardness against algorithms that may arbitrarily err with small probability (i.e., two-sided error hardness) on the *uniform* distribution w.r.t. to deterministic polynomial-time algorithms.

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: OWFs are both necessary [IL89] and sufficient for many of the most central cryptographic primitives and protocols (e.g., pseudorandom generators [BM82; HILL99], pseudorandom functions [GGM86], private-key encryption [GM84], digital signatures [NY89; Rom90], commitment schemes [Nao91; HR07], identification protocols [FS86], coin-flipping protocols [Blu83], and more).

The above two results point to the tantalizing possibility that OWFs may be based on just the worst-case hardness of the (Gap) $\mathrm{MK^tP}$ problem—the "only" thing that needs to be done is to bridge the difference between errorless and two-sided error average-case hardness. However, as argued in several more recent works [HS22; LP21a], this will require substantially new techniques; indeed, more recent works side-step this issue by characterizing OWFs based on the worst-case hardness of somewhat less natural variants of the $\mathrm{GapMK^tP}$ problem [LP24; HN23; LP25]; see Section 1.3 for more details.

Additionally, if this gap could be bridged, and we can furthermore show that the worst-case problem we start with is NP complete, we will have based OWFs on just the assumption that $\mathrm{NP} \not\subseteq \mathrm{BPP}$, solving what has become known as the "holy-grail of cryptography" [DH76].

1.1 Our Results

Our goal is to provide a better understanding of worst-case hardness of *standard* version of the time-bounded Kolmogorov complexity problem, both in terms of its NP-hardness status, and in terms of understanding whether just worst-case hardness of it suffices for cryptographic applications.

Roughly speaking, our key results show the following *under appropriate derandomization assumptions*:

Worst-case Hardness of $\mathrm{GapMK^tP}$ implies hardness of Disjoint NP Pair . We consider the hardness of disjoint NP pairs (DisjNP) [ESY84; Raz94], a *gap* version of $\mathrm{NP} \cap \mathrm{coNP}$ that was extensively studied in the proof complexity [Raz94; Pud03; GSZ06; Pud17; Bey06]. A promise problem $(\mathcal{Y}, \mathcal{N})$ is in DisjNP if both $\mathcal{Y}$ and $\mathcal{N}$ are in NP. We show that worst-case hardness of $\mathrm{GapMK^tP}$ implies hardness of DisjNP.

Theorem 1.1 (Informal, Corollary 5.2)**.** *Assume that there exists a constant $\epsilon > 0$ such that $\mathrm{E} \not\subseteq \mathrm{ioNTIME}[2^{kn}]/2\epsilon n$ for every $k \in \mathbb{N}$. Then there exists a constant c such that the following holds. Assume that for some constant $\alpha > 0$, $\mathrm{Gap_pMK^tP}[n^{\alpha/4c}, n^{\alpha}] \notin \mathrm{P/poly}$ for any polynomial p. Then $\mathsf{DisjNP} \not\subseteq \mathrm{BPP}$.*

It is widely believed that NP-complete languages cannot be inside $\mathrm{NP} \cap \mathrm{coNP}$—indeed, if they were, the Polynomial Hierarchy would collapse. As such, it is generally believed that the assumption that $\mathrm{NP} \not\subseteq \mathrm{BPP}$, or even that $\mathrm{NP} \not\subseteq \mathrm{P/poly}$, does not suffice to show that $\mathrm{NP} \cap \mathrm{coNP} \not\subseteq \mathrm{BPP}$ (or at least, this would be a major development). The notion of DisjNP provides the natural analog/generalization of $\mathrm{NP} \cap \mathrm{coNP}$ in this respect. Even, Selman, and Yacobi [ESY84] conjectured that a problem in DisjNP cannot be NP-hard.[3]

[3]We emphasize that containment in DisjNP is a significantly stronger condition than containment in promise-

Thus, and the same difficulties should apply for basing $\mathsf{DisjNP} \subseteq \mathrm{BPP}$ on the hardness of NP; thus, Theorem 1.1 yields an indication that proving NP-hardness of $\mathrm{GapMK^tP}$ will yield an unexpected development (i.e., that $\mathrm{NP} \not\subseteq \mathrm{P/poly}$ implies that $\mathsf{DisjNP} \not\subseteq \mathrm{BPP}$).

While Theorem 1.1 does not provide a poly-time Turing reduction, it does provide (under the derandomization assumption) a BPP-*restricted reduction* [Hir23]. A polynomial-time Turing reduction from a problem A to a problem B is called BPP-restricted if its correctness is guaranteed for every oracle in BPP that solves B, rather than for every oracle that solves B.

Let us also provide some context for the derandomization assumption we rely on. The most classic derandomization assumptions in the literature assert the existence of a function $f\colon \{0,1\}^n \to \{0,1\}$ computable in 2^{cn} time for some constant c, that cannot be computed in by deterministic [IW97] or non-deterministic [KM99; MV99; SU05] circuits of size $2^{\epsilon n}$ for some constant ϵ. Following [CT21], [LP25] considered an even stronger version, and this is the version that we rely on: it requires the existence of c, ϵ such that for every k, there exists a function running in time $2^{c \cdot kn}$ that is secure against non-deterministic attackers with running time $2^{\epsilon kn}$ but still only $2^{\epsilon n}$ advice. In other words, we allow the attacker to run in time $> 2^n$, but the hard function can be computed in time $\gg 2^n$. (Note that, just as for [IW97; KM99], for each k, there is a $\mathrm{poly}(n)2^{kn}$-time complete hard function.) As for intuition, this assumption relies on exactly the same intuition as [IW97; KM99]: non-determinism and short advice should not enable polynomial speed-up for computations (and this should hold equally for 2^n [IW97; KM99] as 2^{kn} time ([CT21], [LP25] and this paper).

Worst-case Hardness of $\mathrm{GapMK^tP}$ implies ϵ-ZK with a non-uniform simulator. While we are not able to show that just plain worst-case hardness of $\mathrm{GapMK^tP}$ implies OWFs, we are able to show that it implies an appropriate ϵ-secure (for any inverse polynomial, as opposed to negligible, ϵ) version of *zero-knowledge proofs*. Zero-knowledge (ZK) proofs introduced by Goldwasser, Micali and Rackoff [GMR89] are paradoxical constructs that enable a Prover to convince a Verifier that some instance x belongs to a language L without revealing any additional information. The ZK property is formalized by requiring that the view of any potentially-malicious efficient Verifier in an interaction with the prover can be indistinguishably "simulated" by an efficient simulator. It is well known that OWFs imply ZK proof for all of NP, and furthemore, ZK proofs for all of NP, together with the assumption that $\mathrm{NP} \not\subseteq \mathrm{BPP}$[4] also imply OWFs [OW93; HN23; LMP24].

The notion of ϵ-ZK [DNS04] (for some inverse polynomial ϵ) relaxes the notion of zero-knowledge to only require ϵ-indistinguishability of the simulated interaction. We here consider a further relaxation of the notion of ϵ-ZK where we restrict the length of the so-called "auxiliary input" provided to both the Verifier and the distinguisher to some a-priori bounded polynomial (a.k.a. bounded-auxiliary input ZK, see e.g., [Bar01; BC20]) and furthemore allow the simulator to be a *non-uniform PPT* algorithm—in particular, the non-uniform advice of the simulator may be longer than the bound on auxiliary input provided to the verifier and

NP∩coNP; the notion of promise-NP∩coNP simply requires that $\mathcal{Y}$ instances have witnesses proving *non-containment* in $\mathcal{N}$ (i.e., completeness holds for $\mathcal{Y}$ instances, but soundness only holds w.r.t. $\mathcal{N}$ instances), and analogously for $\mathcal{N}$-instances. Indeed, it is well known that NP $\not\subseteq$ BPP implies that promise-NP $\cap$ coNP $\not\subseteq$ BPP. In contrast, the notion of DisjNP requires soundness everywhere (just as NP $\cap$ coNP for languages), and completeness on either $\mathcal{Y}$ or $\mathcal{N}$).

[4] We emphasize that for this converse to hold, it is important to consider ZK secure w.r.t. non-uniform distinguishers.

the distinguisher; we refer to this as (a, ϵ)-ZK *with a non-uniform simulator*. We note that this notion is a strengthening of the notion of "Weak Uniform ZK" considered in [DNRS03]; that notion is even weaker in that the non-uniform simulator is allowed to be chosen after the distinguisher and its running time may be longer than the running time of the distinguisher. In contrast, our notion is closer to the standard notion of ZK in that the same simulator works for all PPT distinguishers, it is just that the non-uniform advice of the simulator, and its running time, is allowed to depend on the bound on auxiliary input provided to the verifier and the distinguisher.

Theorem 1.2 (Informal, Theorem 8.1)**.** *Assume that* $\mathrm{E} \not\subseteq \mathrm{io}\,\mathrm{SIZE}[2^{\Omega(n)}]$*, and that for some* $t \in \mathrm{poly}$ *and some constant* $\epsilon > 0$*,* $\mathrm{Gap_pMK^tP}[n^\epsilon, n - \log n] \notin \mathsf{ioP}/\mathrm{poly}$*, for any* $p \in \mathrm{poly}$*. Then for any* $q \in \mathrm{poly}$*, there exists an (inefficient prover)* $(q(\lambda), 1/q(\lambda))$*-zero-knowledge proof for* NP *with non-uniform simulator, perfect completeness and soundness error* $2^{-\lambda}$*.*

We note that if we consider a notion of randomized K^t [LOS21; BLM05; Oli19] instead of just K^t, we can also get the same result without assuming derandomization assumptions (see Theorem 8.1 and Theorem 9.1).

Theorem 1.3 (Informal, Theorem 9.1)**.** *Assume that for some* $t \in \mathrm{poly}$ *and some constant* $1/3 > \epsilon > 0$*,* $\mathrm{Gap_pMrK^tP}[n^\epsilon, n - \log n] \notin \mathsf{ioP}/\mathrm{poly}$*, for any* $p \in \mathrm{poly}$*. Then for any* $q \in \mathrm{poly}$*, there exists a* $(q(\lambda), 1/q(\lambda))$*-zero-knowledge proof with non-uniform simulator, with* $1 - \mathrm{neg}$*-completeness and soundness error* $2^{-\lambda}$*.*

We also emphasize that the above notion of ZK suffices for many standard applications of ZK (just with inverse polynomial security); for instance, it implies e.g. the notion of Witness Hiding [FS90] w.r.t. uniform algorithms (or those with bounded-auxiliary input) for all relations that are hard for polynomial-size circuits—see Lemma 7.4.

Additionally, our protocol is a proof system (i.e., enjoys unconditional soundness). On the other hand, the protocol requires an *inefficient* prover—we leave it as an intriguing open problem to get a protocol with an efficient prover. (We note that alternatively, we can get a protocol with an efficient *non-uniform* PPT prover; see Theorem 7.2 for more details.)

Characterizing Worst-Case Hardness Through *Conditional PRGs*. Both of the above results pass through a new characterization of the worst-case hardness of $\mathrm{GapMK^tP}$ through the notion of a conditional pseudorandom generator (PRG). Roughly speaking, a (cryptographic) PRG [BM82; Yao82] is an efficiently computable function G that is (1) expanding, and (2) its output given a random input (i.e., $G(\mathbf{U}_n)$) is indistinguishable from random w.r.t. all PPT distinguishers (or in the case of non-uniform security, w.r.t. all non-uniform PPT distinguishers.)

In [LP20], a relaxation referred to as a *conditional PRG (cond-PRG)*; this notion is defined identically the same way, except that condition (2) is replaced by the requirement that there exists some (potentially inefficiently sampleable) distribution X (over seeds) such that $G(X)$ (as opposed to $G(\mathbf{U}_n)$) is indistinguishable from uniform. There it was shown that cond-PRG satisfying an additional *entropy-preserving* property (roughly that the Shannon-entropy of the output of the PRG remains as high as the length of the seed) are equivalent to mild average-case hardness of K^t, and also to OWFs (and thus also to regular, i.e., non-conditional, PRGs by [HILL99]).

One of our principal conceptual contributions is showing that once we drop the entropy-preserving requirement, worst-case hardness of $\text{GapMK}^{\text{t}}\text{P}$ (under derandomization assumptions) implies a conditional PRG. This holds both in the regime of uniform indistinguishability, and in the regime of bounded non-uniform indistinguishability; we here focus on the regime of bounded non-uniform indistinguishability as this is what will be useful for us.

Theorem 1.4 (Informal, Corollary 4.2)**.** *Assume that* $\text{E} \not\subseteq \text{io}\,\text{SIZE}[2^{\Omega(n)}]$*, and that for some efficiently computable functions* $\ell, s \colon \mathbb{N} \to \mathbb{N}$ *and* $t \in \text{poly}$

$$\text{Gap}_{\text{p}}\text{MK}^{\text{t}}\text{P}[s(n), \ell(n)] \notin \mathsf{ioP}/\text{poly},$$

for any polynomial p*. Then for every efficiently computable function* $m(n) \in \text{poly}$ *there exists a function* $G \colon \{0,1\}^{d(n)=s(n)+\log^2 m(n)} \to \{0,1\}^{3d(n)}$ *which is an* ϵ*-conditional PRG secure against* $\text{BPP}/a(n)$ *for* $a(n) \approx (\ell(n)/d(n))^{1/3}$ *and* $\epsilon = 1/a(n)$*.*

In fact, this notion of a conditional PRG also almost directly implies errorless *average-case* hardness of $\text{Gap}_{\text{p}}\text{MK}^{\text{t}}\text{P}$ (see Theorem 3.2), and as such, this notion provides a modular explanation to Hirahara's result [Hir18]).

Note that in Theorem 4.1 and Lemma 3.3 we are simply relying on the most basic derandomization assumption of [IW97] used to derandomize BPP into P. Alternatively, if considering a variant of $\text{GapMK}^{\text{t}}\text{P}$ based on rK^t [LOS21; BLM05; Oli19], then we can again get an unconditional characterization (see Corollary 9.4).

As we shall see, Conditional PRGs are sometimes sufficient for cryptographic application, and indeed, the above two results more generally hold assuming the existence of conditional PRGs (instead of worst-case hardness of $\text{Gap}_{\text{p}}\text{MK}^{\text{t}}\text{P}$), and finally are concluded by appealing to Theorem 1.4.

We finally remark that combined with the results of [LP20], we get that bridging the gap between worst-case hardness and two-sided average-case hardness of $\text{GapMK}^{\text{t}}\text{P}$ is equivalent to bridging the gap between cond-PRGs and cond entropy-preserving PRGs; this opens up a new cryptographic approach towards presenting a worst-case to (two-sided error) average-case reduction for $\text{GapMK}^{\text{t}}\text{P}$.

1.2 Proof Overview

We start with an informal definition of our notion of conditional PRG: An efficiently computable function $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ is a *ϵ-weak conditional PRG against class* C, if (1) $m(n) > d(n)$ and (2) for every distinguisher $\mathsf{Dist} \in \text{C}$ there exists some distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ over $\{0,1\}^{d(n)}$, such that $G(\mathcal{D}_n)$ is ϵ-indistinguishable from uniform in the eyes of Dist. We call G an *ϵ-strong* conditional PRG, or simply a ϵ-conditional PRG, if there exists a single distribution $\mathcal{D}$ that fools all distinguishers $\mathsf{Dist} \in \text{C}$ (and it is a "standard PRG" if $\mathcal{D}$ is simply the uniform distribution).

In the following, we start with constructing a weak-conditional PRG against BPP algorithms with bounded advice. Later, we will show how to convert weak conditional PRG into a strong one, and finally, we will use the strong conditional PRG to construct ZK proofs.

Weak Conditional PRG against BPP distinguishers with bounded advice. We start with describing the construction of weak conditional PRG from the hardness of $\mathrm{Gap_pMK^tP}[s(n), \ell(n)]$. Our PRG relies on intuitions similar to those in the worst-case to average-case reduction of Hirahara [Hir18]. We construct our PRG by constructing an extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ with efficient black-box reconstruction procedure Rec. The guarantee of the reconstruction procedure is that for every $x \in \{0,1\}^n$, if A can ϵ-distinguish between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$, then there exists an advice z of length $a(n)$ such that $\Pr\left[\mathsf{Rec}^{\mathsf{A}}(z, 1^{1/\epsilon}) = x\right] \geq 2/3$. When A is by itself a polytime algorithm with non-uniform advice at most $a(n)$, this implies that $\mathsf{rK}^{\mathrm{poly}}(x) \leq 2a(n) + O(1)$. We choose $a(n)$ such that $2a(n) + O(1) \leq \ell(n)$.

The converse of the above implies that for any $x \in \{0,1\}^n$ with $\mathrm{K}^{\mathrm{poly}(t')}(x) > \ell(n)$, no t' algorithm with advice length $a(n)$ can ϵ-distinguish between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and uniform (note that here we must have $a(n) \leq n/2$ to get a meaningful bound on $\mathrm{K}^{\mathrm{poly}(t')}(x)$. However, the bound is still meaningful for any running time t').

We now define our (weak) conditional PRG $G\colon \{0,1\}^{s(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ as follows: for a program $\Pi \in \{0,1\}^{s(n)}$, we let $G(\Pi, w) = \mathsf{Ext}(\mathsf{U}(\Pi, 1^t), w)$. By the observation above, it follows that if a polytime distinguisher Dist with advice of length $a(n)$ can ϵ-distinguish $G(\Pi, \mathbf{U}_{d(n)})$ from uniform for *any* $\Pi \in \{0,1\}^{s(n)}$, then $\mathrm{Gap_pMK^tP}[s(n), \ell(n)] \in \mathrm{P/poly}$. Indeed, we can construct an algorithm A that solves $\mathrm{Gap_pMK^tP}[s(n), \ell(n)]$ as follows. Given a string x, A simply estimates the distinguishing advantage of Dist between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$. If this distinguishing advantage is at least ϵ, A answers Yes. Otherwise, A answers No.

By the reconstruction property of Ext, we know that for every x with large $\mathsf{rK}^{\mathrm{poly}(t)}$ A must answer No. On the other case, if for some x with $\mathrm{K}^t(x) \leq s(n)$ A answer no with high probability, it follows that Dist cannot distinguish between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and uniform. But $\mathsf{Ext}(x, \mathbf{U}_{d(n)}) = G(\Pi, \mathbf{U}_{d(n)})$ for Π being the K^t-witness for x, which implies that G is secure with respect to Dist (this is realized by the distribution $\mathcal{D}_n = (\Pi, \mathbf{U}_n)$).

Note that in the above we get security against bounded advice, but any polynomial running time. We get this because in the definition of $\mathrm{MK^tP}$ there is a decoupling between the gap in running time and the gap in description length.[5] Looking ahead, this property will be important for our ZK protocol.

We next explain how to construct Ext. Following [Hir18], we use the STV PRG [STV99; NW94] using an encoding of x with a list-decodable error correcting code E. Our error correcting code is concatenation of Reed-Solomon and Hadamard, as in [Hir18]. Our construction is however different in two points, as we want to deal with a larger class of adversaries:

- First, we want a single PRG G which is secure against any distinguishing advantage $\epsilon \in 1/\mathrm{poly}$.[6] For this, we need an extractor Ext with reconstruction procedure for any such ϵ. For this, we take the size of the field $\mathbb{F}$ of the Reed-Solomon code to be super-polynomial in n. This implies that the encoding $\mathrm{E}(x)$ of $x \in \{0,1\}^n$ is super-polynomial in n, and thus we cannot compute $\mathrm{E}(x)$ efficiently. Luckily, for the NW PRG we only need to be able to compute $\mathrm{E}(x)$ locally (that is, given an index i we need to be able to compute $\mathrm{E}(x)_i$ in polynomial time), a property which both Reed-Solomon and Hadamard have.

[5] This wouldn't work if we had started with hardness of e.g., MCSP[KC00; Tra84].

[6] For the remainder of our results, getting ϵ to be negligible will not have an effect, and for those, this step can be omitted, but we believe it is of independent interest showing that worst-case hardness of $\mathrm{Gap_pMK^tP}$ implies weak cond PRG with a negligible indistinguishability gap.

In the reconstruction stage, given the distinguishing advantage ϵ, we simply choose a large enough subset of the field $\mathbb{F}$, of sufficiently large size $\text{poly}(1/\epsilon)$, and execute the list-decoding of the concatenation of Reed-Solomon and Hadamard on this sub-code [GS98; Vad+12]. Here we also rely on the local-list-decoding procedure of the Hadamard code ([GL89]). We note that the reconstruction here is randomized, which leads us to the next change.

- Second, we want to consider *randomized* adversaries A (and a randomized reconstructing procedure). This means that the advice z we need for the reconstruction cannot be dependent in the randomness the adversary A uses.

  The above mentioned concatenation of RS and Hadamard naturally supports list decoding with list of size $\text{poly}(1/\epsilon)$. Together with the reconstruction procedure of NW, we get that for every (randomized) adversary A there exists a short advice z, such that $\mathsf{Rec}^{\mathsf{A}}(z)$ outputs a list $\mathcal{L}$ of size $\text{poly}(1/\epsilon)$ such that $x \in \mathcal{L}$ with high probability. When both A and Rec are deterministic, we can now simply add to z the index i of x in the list $\mathcal{L}$ (using additional $O(\log 1/\epsilon) = O(\log n)$ bits).[7]

  When A and Rec are randomized, we do not know the list $\mathcal{L}$ in advance, as this list can be dependent on the randomness. A natural approach is to add a short hash $h(x) \in \{0,1\}^{\text{poly}(\log n)}$, such that with high probability over the randomness of Rec, A, there is no other $x' \in \mathcal{L}$ with $h(x) = h(x')$. By adding $h(x)$ to the advice, the reconstruction procedure will be able to identify the right x inside the list.

  One can try to do it by taking h to be a 2-universal hash function. Indeed, by adding a short hash $h(x)$ of length $O(\log n)$, we can with high probability identify x uniquely. The problem is that in order to describe a 2-universal hash $h\colon \{0,1\}^n \to \{0,1\}$, we need at least $O(n)$ bits, which is too much (note that we need to choose h at random to the above argument to work).

  Instead, we use almost-universal hash-functions, that can be constructed, for examples, from error-correcting codes. We simply encode x using an error-correcting code with distance $1/2$ E' (we use RS again here), and then simply choose random indexes $i_1, \ldots, i_{O(\log 1/\epsilon)}$, and add to the advice the $(i_1, \ldots, i_{O(\log 1/\epsilon)})$ together with $\mathrm{E}'(x)_{i_1}, \ldots, \mathrm{E}'(x)_{i_{O(\log 1/\epsilon)}}$. This adds $O(\log n \log 1/\epsilon) = O(\log^2 n)$ bits to the advice, and since for any x' we have that $\Pr_i[\mathrm{E}'(x)_i = \mathrm{E}'(x')] \leq 1/2$, we get that

$$\Pr_{i_1,\ldots,i_{O(\log 1/\epsilon)}}\left[(\mathrm{E}'(x)_{i_1}, \ldots, \mathrm{E}'(x)_{i_{O(\log 1/\epsilon)}}) = (\mathrm{E}'(x')_{i_1}, \ldots, \mathrm{E}'(x')_{i_{O(\log 1/\epsilon)}})\right] \leq 1/\text{poly}(1/\epsilon),$$

  which implies that there are no collisions for x with high probability.

We highlight that the running time of the resulting conditional PRG is larger than the bound on the advice of the distinguisher, but does not depend on the distinguisher's running time. In particular, it is secure w.r.t. all poly time distinguishers.

Weak conditional PRG to Strong conditional PRG. So far we constructed a PRG

$$G\colon \{0,1\}^{d'(n)=s(n)+d(n)} \to \{0,1\}^{m(n)}$$

[7] Similarly, for pK^t we can do the same as we can choose the advice after the randomness is chosen.

such that for every A there is some Π such that $G(\Pi, \mathbf{U}_{d(n)})$ is indistinguishable from uniform. We want to have a single distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ such that $G(\mathcal{D})$ is hard against any adversary in $\mathrm{BPP}/a(n)$.

We get such a distribution using the minimax theorem. However, for this to work we need to fix the distinguishing advantage ϵ and the distinguisher running time $T \in \mathrm{poly}$. Using the minimax theorem, we get that for some distribution $\mathcal{D}^T = \{\mathcal{D}^T_n\}_{n\in\mathbb{N}}$, $G(\mathcal{D}^T)$ is secure against all distinguishers in $\mathsf{BPTIME}[T]/(\epsilon^2(n) \cdot a(n)/m(n))$. Therefore, to get meaningful advice length, we first need to fix $\epsilon \gg 1/\sqrt{a(n)}$.

Next, we notice that the above only gives a hard distribution $\mathcal{D}^T = \{\mathcal{D}^T_n\}_{n\in\mathbb{N}}$ against $\mathsf{BPTIME}[T]/a'(n)$ for any $T \in \mathrm{poly}$, while we want a single distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ which is secure against any polynomial running time. Luckily, since we do not care about the distribution being sampleable, we are able to show that these are in fact equivalent. Indeed, we take $\mathcal{D}_n = \mathcal{D}^{n^{\alpha(n)}}_n$ for some carefully chosen $\alpha(n) = \omega(1)$.

(Weak) conditional PRG to average-case hardness of $\mathrm{MK^tP}$. We note that conditional PRGs almost immediately imply the errorless hardness of $\mathrm{GapMK^tP}$. Indeed, any output y of a conditional PRG $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ has K^t at most $d(n)+O(1)$, as it can be described by the input x for G together with the short description of G itself. This implies that for any distribution $\mathcal{D}$, $G(\mathcal{D})$ is a distribution over strings with low K^t. On the other hand, uniform y has K^t close to $m(n)$ with high probability. Thus, any errorless algorithm for $\mathrm{GapMK^tP}[d(n)+O(1), m(n)-O(1)]$ must output "No" with high probability on uniformly chosen y, while, on the other hand, it must always output "Yes" (or $\perp$) on samples from $G(\mathcal{D})$. In other words, any errorless $\mathrm{GapMK^tP}$ algorithm can distinguish $G(\mathcal{D})$ from uniform, for any distribution $\mathcal{D}$. If the assumed $\mathrm{GapMK^tP}$ algorithm is poly-time, this violets the security of the (weak) conditional PRG G.

Distribution-Aided Commitments from (strong) Conditional PRG. We next explain how to construct commitment schemes from a conditional PRG. For this, we will use Naor commitment scheme [Nao91]. Given a $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{3d(n)}$, in Naor's commitments scheme, the committer Com interacts with the receiver Rec to commit on a bit b. For this, Rec sends to Com a random string $r \in \{0,1\}^{3d(n)}$, and Com samples $w \leftarrow \mathcal{D}$, and answers with $y = G(w)$ to commit on $b = 0$, or $y = G(w) \oplus r$ to commit on 1.

In the revealing stage, to open the commitment, Com simply sends w and b to Rec. The binding property (that Com cannot open the commitment both to 0 and 1 simultaneously) holds for any unbounded algorithm Com. This follows by the fact that for a random r, there are no w, w' such that $G(w) \oplus G(w') = r$. Thus, Com cannot find y such that $y = G(w)$ and $y = G(w') \oplus r$.[8]

On the other hand hiding holds to the same class against the PRG is secure. In our case, as for any Dist with bounded advice of length $a'(n)$, $G(\mathsf{Dist})$ is pseudorandom. It follows that Dist cannot distinguish between commitment to $b = 0$ and commitment to $b = 1$, when w is sampled from $\mathcal{D}$.

We note however that Com need to be able to sample from the distribution $\mathcal{D}$, and thus it is not efficient. For now, we will assume Com has an oracle-access to the distribution but otherwise is efficient (i.e., Com is an efficient distribution-aided algorithm). Looking ahead, we will use such

[8]The argument here is similar to the "Razborov trick" [Raz94], who showed that if $\mathrm{K}^t(r) \geq s$ is large, then for any string x either $\mathrm{K}^t(r \oplus x) \gtrsim s/2$ or $\mathrm{K}^t(x) \gtrsim s/2$. Similarly, for any r with large enough $\mathrm{K}^t(r)$, and for any x, either x or $x \oplus r$ are not in the image of G.

an efficient distribution-aided Com to construct our ZK proof, and then replace the oracle with non-uniform advice.

Finally, we want to be able to use the commitment multiple times, and for this we need the commitment to be secure under repetitions. Using a standard hybrid argument, it follows that (Com, Rec) is secure for $r(n)$ repetitions, as long as $r(n) \leq a'(n)/d(n)$. In the ZK proof, we will need $r(n) \geq n^{\epsilon}$ for some constant $\epsilon > 0$.

We remark that here the decoupling between the running time of the distinguisher and its advice in the security definition of conditional PRG is important. In the hybrid argument, we need to hardcode in the advice $r(n)$ pairs of commitments and their openings. An opening to the commitment is an input to the PRG, which is luckily short, and includes a short program Π that produces some GapMK$^{\mathrm{t}}$P instance x. However, to verify the opening, Rec (and therefore the distinguisher Dist) needs to run the PRG to make sure it gets the correct output. Since the PRG needs to execute Π to get x, its running time is at least as large as the length of x. While this is polynomial, is it much larger than the length of the non-uniform advice we can support (recall that in the security proof we use Dist to compress x, so to get non trivial compression, its description length must be shorter than x).

Distribution-Aided Commitments from Hardness of DisjNP. Alternatively, instead of starting from conditional PRG, we can construct distribution-aided commitments from the worst-case hardness of DisjNP (which is implied by conditional PRG, see below). Using this construction, we get that the worst-case hardness of DisjNP implies the existence of weak-ZK with non-uniform simulator for NP.[9]

The commitments construction is as follows: First, we use the minimax theorem together with the assumed worst-case hardness of $\Pi = (\mathcal{Y}, \mathcal{N}) \in$ DisjNP to get an (inefficiently samplable) distribution $\mathcal{D}$ on which Π is hard on average. We extend the distribution $\mathcal{D}$ so it will be a distribution over pairs (x, w), where $x \in \mathcal{Y}$ or $x \in \mathcal{N}$, and w is its appropriate NP-witness for x. Next, given an oracle to $\mathcal{D}$ and a bit $b \in \{0, 1\}$ to commit on, Com samples a pair (x, w). If $x \in \mathcal{Y}$, the Com sends b to the Rec. If $x \in \mathcal{N}$, the Com sends $1 - b$ to the Rec. To open the commitment, the Com simply sends the witness w. The hiding of the scheme follows by the hardness of Π over $\mathcal{D}$, while the binding follows by the fact that both $\mathcal{Y}, \mathcal{N}$ are in NP.

In our actual construction we do not take this path, and instead use the above described construction of commitments from condition PRGs, due to simplicity and better parameters.

ZK from Commitments. We next use the commitment scheme to construct a ZK proof for NP. As a first step, we construct such a proof where both the prover and the simulator have oracle access to the distribution $\mathcal{D}$. This will allow them to take the role of Com in the commitment scheme. Later, we will show how to get rid of this assumption using non-uniform advice. In the following fix a security parameter λ, and $q \in$ poly. We will construct a $1/q(\lambda)$-ZK with $q(\lambda)$-bounded auxiliary input.

To get 4-message proof with a constant soundness error, we simply use the protocol of [Blu86] for HamiltonianCycle (or we can use any commitment-based ZK proof). For smaller soundness error, we simply use sequential repetitions.

This proof for HamiltonianCycle uses k^2 commitments for an instance which is a graph with k vertices (or λk^2 to get soundness error $\approx 2^{-\lambda}$). Therefore we need the commitment to be secure

[9] Note that we get a similar result assuming the worst-case hardness of TFUP as it implies the hardness of DisjNP.

for distinguishers that see $\lambda \cdot k^2$ commitments, and have additional advice of length $q(\lambda)$.

For this, given a HamiltonianCycle instance of size k, security parameter λ, and desired advice length $q(\lambda)$, we simply choose n such that the commitment will be secure against distinguisher with advice of length $n^\epsilon \gg \lambda k^2$ repetition, and with additional advice length at least $a'(n) \geq 2q(\lambda)$. We get this by taking n to be $q(\lambda) + (\lambda k^2)^{1/\epsilon}$, and using the commitments on samples from $\mathcal{D}_n$.

We note that to show that the protocol is ZK when taking sequential repetition (needed to get soundness error $2^{-\lambda}$), we cannot use the standard closure of ZK under sequential repetition [GO94]. The reason is that in this proof, to prove the ZK of the i-th round, the simulator needs to get the randomness of the verifier in previous rounds. However, as we did not limit the running time of the verifier, the randomness can be longer than the auxiliary-input we are able to deal with.

Instead, we directly construct a simulator that simulates all of the λ repetition of the protocol together (relying on specific properties of this protocol). In the proof, we use a hybrid argument, and use the fact that the distinguisher can simulate both the honest-prover and the simulator behavior given short advice—the verifier description and the witness for the HC instance, and most importantly, a short advice string that enables simulating the commitments.

Removing the oracle to $\mathcal{D}$. Lastly, we need to make the simulator (and prover) efficient. For this, we simply replace the oracle access to $\mathcal{D}$ with a non-uniform advice containing $\text{poly}(1/\epsilon, q(\lambda))$ samples from $\mathcal{D}$. A simple argument shows that for a random choice of the samples, no distinguisher with bounded advice can distinguish between the non-uniform simulator/prover and the distribution-aided one. This implies that the non-uniform proof is still ZK.

Obtaining a Hard Promise Problem in DisjNP from Cond PRG. We finally mention a few words on how to obtain a hard promise problem in DisjNP. Our starting point is the construction of [BOV03] of a non-interactive commitment scheme from any 2-round public-coin commitment scheme under derandomization assumption; their construction simply uses an appropriate PRG against non-deterministic bounded-time attackers to instantiate the receiver message. In more detail, the committer enumerates all seeds to the PRG and provides a commitment to its value under the output of the PRG on each such seed (interpreted as the first message of the 2-round public-coin commitment). Note that any non-interactive commitment directly yields a hard problem in DisjNP.

As a first approach, one could hope to apply this approach to the commitments constructed above. The problem with this approach, however, is that the seed-length needed for the approach of [BOV03] grows logarithmically with the running time of the committer in the 2-round scheme. In our scheme, however, to enable running $Q = 2^s$ commitments in parallel (where s is the seed length), we need a commitment that is secure w.r.t. attackers with non-uniform advice that grows with Q, and the running time of our conditional PRG grows at least linearly with Q. In essence, the dependency is circular: increasing the seed length requires using more instances of the commitment, which in turn requires us to increase the running time of the PRG, which in turn requires further increasing the seed length! This was not an issue in [BOV03] since they started with a commitment that is secure under any number of repetitions (with the same running time).

We overcome this issue by relying on a seed efficient complexity-theoretic PRG as recently constructed in [LP25], following [CT21]; most notably, there it was shown that under the complexity assumption from Theorem 1.1, we can construct a PRG whose seed length is always $\log n$ but it can be made to fool attackers with longer running time simply by increasing the running time of

the PRG; this breaks the above circularity.

1.3 Additional Related Works

We review some related work, in terms of (1) the NP-completeness status of the time-bounded Kolmogorov complexity problem, (2) *variants* of the GapMK$^{\mathrm{t}}$P problem whose worst-case hardness is known to imply/characterize OWFs, and (3) other results on achieving computationally-secure cryptographic primitives based on hardness assumption that are not known to imply OWFs.

Earlier Results on NP-completeness

Towards NP-completeness: While it is still unknown whether the original problems are NP-complete, several generalizations of them have been proven to be NP-complete. Most notably, Ilango first demonstrated this for an oracle version of MCSP [Ila20]; this was subsequently extended to a multi-bit version of MCSP referred to as Multi-MCSP [ILCO20], to a conditional version of the MKTP problem, McKTP [LP22], and to other variants [Hir22]. [HIR23] recently improved the parameters of the reduction to McKTP [LP22], assuming that witness encryption scheme exists. Additionally, Ilango [Ila23] demonstrates that NP-hardness of a variant of MCSP and MKTP where the programs are allowed to access a random oracle, yielding a *heuristic* NP-completeness Karp (i.e., many-one) reduction for these problems (if instantiating the random oracle with a concrete hash function). Impagliazzo, Kabanets, and Volkovich [IKV23] provides various different results that can be interpreted as giving evidence that MCSP is NP-complete with respect to randomized reductions. Very recently, Hirahara and Ilango [HI25] showed the MCSP is NP-hard under quasi-polynomial non-adaptive reductions assuming subexponentially secure NIWI and worst-case assumptions.

Towards Non NP-completeness: There is also evidence pointing towards non NP completeness: Allender and Hirahara [AH19] showed that assuming one-way functions, the gap version of MCSP is not NP complete for super-polynomial gap. Ko [Ko91] showed that a version of MKTP is not NP complete with respect to an oracle, and Ren and Santhanam [RS22] gave an oracle with respect to which MCSP is not NP complete. Other works prove limitations on the structure of reduction to meta-complexity problems. Murray and Williams [MW17] prove that MCSP is not NP complete under so-called *local reductions*. Kabanets and Cai [KC00] and Saks and Santhanam [SS20] show that the NP-completeness of MCSP under Turing reductions with certain properties implies circuit lower bounds. For example if MCSP is complete under so-called *parametric honest* Turing reductions, then $\mathrm{E} \not\subseteq \mathrm{SIZE}[\mathrm{poly}]$. More recently, Saks and Santhanam [SS22] gave evidence that the running time of any randomized non-adaptive reduction from SAT to K^t approximation must grow with the time parameter t. [MP24] showed that assuming indistinguishability Obfuscation (iO), GapMK$^{\mathrm{t}}$P is not NP complete with respect to Levin reductions.

Earlier work on basing Cryptography on Worst-case hardness of Kolmogorov complexity style problem. Recently, worst-case characterizations of OWFs were obtained [LP24; HN23; LP25], considering some variants of the time-bounded Kolmogorov complexity problem. In particular:

- [LP24] characterize OWFs through the problem of determining whether $\mathrm{K}^t(x)$ is large or small, but restricting attention to instances x with very large *unbounded* Kolmogorov complexity (i.e., $\mathrm{K}(x) > n - O(\log n)$), or alternatively, to strings x whose so-called "computational depth", $cd^t(x) = \mathrm{K}^t(x) - \mathrm{K}(x) < O(\log n)$, is small (whereby "restricting attention" means that we consider worst-case hardness of a promise problem that only considers those instances; that is, any efficient algorithm must fail on one of those instances in the promise).

- [HN23] provide a worst-case characterization of a variant of OWFs, referred to as infinitely-often OWFs, through the problem of "estimating the probability that a random time-bounded program outputs a certain string" (which can be shown to be related to the notion of probabilistic K^t), while restricting attention to instances satisfying an analog of small computational depth (with respect to this complexity notion).[10]

The unappealing aspect of the above characterizations is that once we add the "conditioning", it becomes less clear what the intuitive interpretation of the problems is. On a technical level, the property that we condition on (i.e., computational depth being small, or Kolmogorov complexity being large) is not *decidable*.

Very recently, [LP25] presented an approach towards a more natural problems whose worst-case hardness may characterize OWFs. Ideally, we would like to show that the classic $\mathrm{MINK}^{\mathrm{t,poly(t)}}$ problem—that is, the problem of simply determining whether a string is "structured" (i.e., $\mathrm{K}^t(x) < n - 1$) or "random" (i.e., $\mathrm{K}^{\mathrm{poly}(t)} \geq n - 1$)—suffices to imply the existence of one-way functions (OWF). [LP25] recently showed that under some derandomization assumptions, worst-case hardness of a *boundary* version of $\mathrm{MINK}^{\mathrm{t,poly(t)}}$—where, roughly speaking, the goal is to decide whether given an instance x, deciding whether (a) x is $\mathrm{K}^{\mathrm{poly}}$-random (i.e., $\mathrm{K}^{\mathrm{poly}(t)}(x) \geq n - 1$), or just close to $\mathrm{K}^{\mathrm{poly}}$-random (i.e., $\mathrm{K}^t(x) < n - 1$ *but* $\mathrm{K}^{poly(t)} > n - \log n$)—characterizes OWF.

Weak forms of ZK from Assumptions not known to imply OWFs. We are aware of very few examples of cryptographic primitives that are not known to exist unconditionally, yet can be instantiated based on the hardness of some problem not known to imply OWFs.

Recently, a few such examples have been given; in contrast to our work, however, they all rely on some form of *average-case* hardness; in contrast, all our results simply require worst-case hardness of DisjNP (as well as worst-case derandomization assumption):

- The work of [BDS25] constructs a so-called "ZAPs with inefficient prover" from "Hard to invert functions". ZAPs are 2-round witness indistinguishable (WI) protocols, where WI is a significant relaxation of ZK; the notion of a hard-to-invert function, while being a relaxation of the notion of a OWF is still an average-case hardness notion. They also obtain a notion of non-uniform simulator *honest-verifier* ZK (similar to our notion of non-uniform simulator ZK) from the same assumption.

  [BDS25] also show how to instantiate Hard-to-invert functions based on the average-case hardness of TFUP; in contrast, we can use just worst-case hardness of TFUP (see Theorem 8.2 and Lemma 5.4).

[10] An alternative type of worst-case characterization of OWFs was also obtained in [HN23], where it is shown that OWFs exists iff NP $\not\subseteq$ BPP and a certain "distributional"-K^t problem is NP-complete w.r.t. a certain type of (restricted) reductions. Note that simply worst-case hardness of the distributional K^t problem alone is not sufficient to get OWFs. Rather, the characterization is in terms of both a hardness assumption (NP $\not\subseteq$ BPP) combined with an feasibility assumption (the existence of a certain type of NP-completeness reduction).

- The work of [BKPRV24] show that so called "batch arguments" (BARG) give a form of a weak WI with non-uniform prover. BARGs are not known to imply OWFs, but are a sophisticated cryptographic protocols satisfying an average-case hardness notion.

2 Preliminaries

2.1 Notations

All logarithms are taken in base 2. We use calligraphic letters to denote sets and distributions, uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Let PPT stand for probabilistic poly-time, and n.u.-poly-time stand for non-uniform poly-time. An n.u.-poly-time algorithm A is equipped with a (fixed) poly-size advice string set $\{z_n\}_{n\in\mathbb{N}}$ (that we typically omit from the notation), and we let A_n stand for A equipped with the advice z_n (used for inputs of length n). For a randomized algorithm A, we denote by $\mathsf{A}(\cdot;r)$ the algorithm A with fixed randomness $r \in \{0,1\}^*$. Let neg stand for a negligible function. Given a vector $v \in \Sigma^n$, let v_i denote its i^{th} entry, let $v_{<i} = (v_1, \ldots, v_{i-1})$ and $v_{\leq i} = (v_1, \ldots, v_i)$. Similarly, for a set $\mathcal{I} \subseteq [n]$, let $v_{\mathcal{I}}$ be the ordered sequence $(v_i)_{i\in\mathcal{I}}$. For $x, y \in \{0,1\}^*$, we use $x||y$ to denote the concatenation of x and y.

For a language $\mathcal{L}$, and a string x, we let $\mathcal{L}(x)$ be the indicator function that outputs 1 if $x \in \mathcal{L}$ and 0 otherwise.

2.2 Distributions and Random Variables

When unambiguous, we will naturally view a random variable as its marginal distribution. The support of a finite distribution $\mathcal{P}$ is defined by $\mathrm{Supp}(\mathcal{P}) := \{x\colon \Pr_{\mathcal{P}}[x] > 0\}$. For a (discrete) distribution $\mathcal{P}$, let $x \leftarrow \mathcal{P}$ denote that x was sampled according to $\mathcal{P}$. Similarly, for a set $\mathcal{S}$, let $x \leftarrow \mathcal{S}$ denote that x is drawn uniformly from $\mathcal{S}$. For $m \in \mathbb{N}$, we use $\mathbf{U}_m$ to denote a uniform random variable over $\{0,1\}^m$ (that is independent from other random variables in consideration). The statistical distance (also known as, variation distance) of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P},\mathcal{Q}) := \max_{\mathcal{S}\subseteq\mathcal{X}} |\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{S}} |\mathcal{P}(x) - \mathcal{Q}(x)|$.

For an ensemble of distributions $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, and an oracle-aided algorithm A, we use $\mathsf{A}^{\mathcal{D}}$ to denote the algorithm A with an oracle to the randomized function $\mathcal{O}$ that on input 1^n outputs a random sample from $\mathcal{D}_n$.

Indistinguishability For a complexity class $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ and a function $\epsilon\colon \mathbb{N} \to [0,1]$, we say that two ensembles $\mathcal{X} = \{\mathcal{X}_n\}_{n\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ are ϵ-indistinguishable against $\mathcal{C}$, if for every large enough $n \in \mathbb{N}$ and for every $\mathsf{Dist} \in \mathcal{C}_n$ it holds that

$$|\Pr[\mathsf{Dist}(\mathcal{X}) = 1] - \Pr[\mathsf{Dist}(\mathcal{Y}) = 1]| \leq \epsilon(n).$$

2.3 Complexity Classes

In this paper, for a fixed universal TM U, and for functions $T, a\colon \mathbb{N} \to \mathbb{N}$, we let $\mathsf{TIME}_{\mathsf{U}}[T(n)]/a(n)$ be the class of algorithms that can be described as the universal TM with advice of length $a(n)$ and time bound $T(n)$. Formally,

Definition 2.1. *For functions $T, a\colon \mathbb{N} \to \mathbb{N}$, a non-uniform algorithm A is in $\mathsf{TIME}_{\mathsf{U}}[T(n)]/a(n)$, if there exists an ensemble $\{s_n\}_{n\in\mathbb{N}}$ with $|s_n| \leq a(n)$, such that for every $n \in \mathbb{N}$ and $x \in \{0,1\}^n$, $\mathsf{A}(x) = \mathsf{U}(s_n(x), 1^{T(n)})$.*

We note that there is a fixed $p \in$ poly (the overhead of U) such that for every T-time non-uniform algorithm A with advice of length $a(n)$ there exists a constant c such that $\mathsf{A} \in \mathsf{TIME}_{\mathsf{U}}[p(T(n), n)]/(a(n) + c)$.

We proceed to define the notion of errorless average-case hardness.

Definition 2.2 (AvgBPP)**.** *A pair $(\mathcal{L}, \mathcal{D})$ of a language $\mathcal{L}$ and a samplable distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ is in* AvgBPP *if there exists a PPT A such that the following holds for infinitely many $n \in \mathbb{N}$:*

- *For every $x \in \mathsf{Supp}(\mathcal{D}_n)$, $\Pr[A(x) \in \{\perp, \mathcal{L}(x)\}] \geq 0.9$*
- $\Pr_{x\leftarrow\mathcal{D}_n}[A(x) = \perp] \leq 1/4$.

We will need the following result, from Liu and Pass [LP25].

Theorem 2.3 ([LP25])**.** *Assume that there exists a constant $\epsilon > 0$ such that $\mathrm{E} \nsubseteq \mathrm{ioNTIME}[2^{kn}]/2\epsilon n$ for every $k \in \mathbb{N}$. Then there exists a constant c such that the following holds. For every polynomial t, there exists a function $G_{NT}\colon \{0,1\}^{\lceil c \log n\rceil} \to \{0,1\}^n$ computable in time $\mathrm{poly}(n)$, such that*

$$\left|\Pr[\mathsf{Dist}(U_n)] - \Pr\left[\mathsf{Dist}(G_{NT}(U_{\lceil c\log n\rceil}))\right]\right| \leq 1/6$$

for every $\mathsf{Dist} \in \mathrm{NTIME}[t(n)]/n$.

2.3.1 Disjoint NP Pairs

We next define the class of disjoint NP pairs, DisjNP [ESY84; Raz94].

Definition 2.4. *A pair of languages $\Pi = (\mathcal{Y}, \mathcal{N})$ is in* DisjNP *if $\mathcal{Y} \cap \mathcal{N} = \emptyset$ and $\mathcal{Y} \in \mathrm{NP}, \mathcal{N} \in \mathrm{NP}$.*

2.4 Kolmogorov Complexity

Roughly speaking, the *t-time-bounded Kolmogorov complexity*, $\mathrm{K}^t(x)$, of a string $x \in \{0,1\}^*$ is the length of the shortest program $\Pi = (M, y)$ such that, when simulated by a universal Turing machine, Π outputs x in $t(|x|)$ steps. Here, a program Π is simply a pair of a Turing Machine M and an input y, where the output of P is defined as the output of $M(y)$. When there is no running time bound (i.e., the program can run in an arbitrary number of steps), we obtain the notion of Kolmogorov complexity.

In the following, fix universal TM U with polynomial simulation overhead, and let $\mathsf{U}(\Pi, 1^t)$ denote the output of Π when emulated on U for t steps. We now define the notion of Kolmogorov complexity with respect to the universal TM U.

Definition 2.5. *Let t be a polynomial. For all $x \in \{0,1\}^*$, define the t-*bounded Kolmogorov complexity *of x*

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

where $|\Pi|$ *is referred to as the* description length *of* Π. *For probability parameter* $\lambda \in [0,1]$, *the* randomized t-bounded Kolmogorov complexity *of* x *is defined as*

$$\mathsf{rK}^t_\lambda(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \Pr_{r \leftarrow \{0,1\}^{t(n)}} \left[\mathsf{U}((\Pi, r), 1^{t(|x|)}) = x\right] \geq \lambda\}.$$

When $\lambda = 2/3$, *we omit it and simply use* $\mathsf{rK}^t(x)$.

We will use the following bound on the Kolmogorov complexity of strings sampled from the uniform distribution.

Lemma 2.6. *For any universal TM* U *and every* $n \in \mathbb{N}$, *it holds that*

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathrm{K}^t_{\mathsf{U}}(x) \geq n - i\right] \geq 1 - 2^{-i}.$$

Lemma 2.7 (Derandomizing rK^t, [GKLO22; LP25], Theorem 2.3)**.**

- *Assume* $\mathrm{E} \not\subseteq \mathrm{io}\,\mathrm{SIZE}[2^{\Omega(n)}]$. *Then there exists* $p \in \mathrm{poly}$ *such that* $\mathrm{K}^{p(t)}(x) \leq \mathsf{rK}^t(x) + \log p(t(|x|))$ *for every* $x \in \{0,1\}^*$ *and* $t \colon \mathbb{N} \to \mathbb{N}$ *with* $t(n) \geq n$.
- *Assume that there exists a constant* $\epsilon > 0$ *such that* $\mathrm{E} \not\subseteq \mathrm{ioNTIME}[2^{kn}]/2\epsilon n$ *for every* $k \in \mathbb{N}$. *Then there exists a constant* d *such that for every* $t \in \mathrm{poly}$ *the following holds. There exists* $p \in \mathrm{poly}$ *such that* $\mathrm{K}^{p(t)}(x) \leq \mathsf{rK}^t(x) + d \log|x|$ *for every* $x \in \{0,1\}^*$.

We next define two promise problems related to K^t. We start with the definition of $\mathrm{Gap_pMK^tP}$. For $p, t \in \mathrm{poly}$, let $\mathrm{Gap_pMK^tP}[s, \ell]$ be the following promise problem:

- $\mathcal{Y} = \left\{x \in \{0,1\}^n \colon \mathrm{K}^{t(n)}(x) \leq s(n)\right\}$
- $\mathcal{N} = \left\{x \in \{0,1\}^n \colon \mathrm{K}^{p(t(n))}(x) > \ell(n)\right\}$

Next, we define $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})$.

Definition 2.8 ($(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})$)**.** $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n)] = (\mathcal{Y}, \mathcal{N})$ *is the following promise problem:*

- $\mathcal{Y} = x \in \{0,1\}^n \colon \mathrm{K}^t(x) \leq s(n)$
- $\mathcal{N} = x \in \{0,1\}^n \colon \mathsf{rK}^{p(t)}(x) \geq \ell(n)$

We observe the following corollary of Lemma 2.7.

Lemma 2.9. *Assume that* $\mathrm{E} \not\subseteq \mathrm{io}\,\mathrm{SIZE}[2^{\Omega(n)}]$, *and that for some functions* $\ell, s, : \mathbb{N} \to \mathbb{N}$ *and for* $t \in \mathrm{poly}$ *with* $t(n) \geq n$, $\mathrm{Gap_pMK^tP}[s, \ell] \notin \mathsf{ioP}/\mathrm{poly}$, *for any polynomial* p. *Then* $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n) - \omega(\log t(n))] \notin \mathsf{ioP}/\mathrm{poly}$, *for any polynomial* p.

Proof. Let q be the polynomial promised in the first item of Lemma 2.7. Assume that for some p, $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n) - \log q(p(t(n)))] \in \mathsf{ioP}/\mathrm{poly}$. We observe that the set of No instances of $\mathrm{Gap_{q(p)}MK^tP}[s, \ell]$ is a subset of the set of No instances of $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{q(p(t))})[s(n), \ell(n) - \log q(p(t(n)))]$, and thus $\mathrm{Gap_{q(p)}MK^tP}[s, \ell][s, \ell] \in \mathsf{ioP}/\mathrm{poly}$, in contradiction. □

2.5 Useful Facts

We will also use the well-known Chernoff bound in our proof.

Fact 2.10 (Hoeffding's inequality)**.** *Let* $\mathbf{A}_1, ..., \mathbf{A}_n$ *be independent random variables s.t.* $\mathbf{A}_i \in \{0,1\}$*. Let* $\widehat{\mathbf{A}} = 1/n \cdot \Sigma_{i=1}^{n} \mathbf{A}_i$ *and* $\mu = \mathrm{E}\left[\widehat{\mathbf{A}}\right]$*. For every* $\epsilon \in [0,1]$ *it holds that:*

$$\Pr\left[\left|\widehat{\mathbf{A}} - \mu\right| \geq \epsilon\right] \leq 2 \cdot e^{-\epsilon^2 \cdot n}.$$

Lemma 2.11 (Distinguishability to prediction)**.** *There exists an oracle-aided* PPT *algorithm* P *such that the following holds. Let* Q *be a distribution over* $\{0,1\}^* \times \{0,1\}$*, let* Dist *be an algorithm and* $\alpha \in [0,1]$ *such that,*

$$\Pr_{(x,y)\leftarrow Q, z\leftarrow\{0,1\}}[\mathsf{Dist}(x,z) = 1] - \Pr_{(x,y)\leftarrow Q}[\mathsf{Dist}(x,y) = 1] \geq \alpha.$$

Then

$$\Pr_{(x,y)\leftarrow Q}\left[\mathsf{P}^{\mathsf{Dist}}(x) = y\right] \geq 1/2 + \alpha.$$

3 Conditional PRGs

In this section we define conditional PRGs and prove some basic results. We start with the definition.

Definition 3.1 (Conditional PRG)**.** *Let* $\epsilon\colon \mathbb{N} \to \mathbb{N}$ *be a function. An efficiently computable function* $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ *is an* ϵ-weak conditional PRG against distinguishers class $\mathcal{C}$ *if for every* $n \in \mathbb{N}$ $m(n) > d(n)$*, and for every distinguisher* $\mathsf{Dist} \in \mathcal{C}$ *there exists a distribution ensemble* $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ *over* $\{0,1\}^{d(n)}$*, such that*

$$\left|\Pr[\mathsf{Dist}(G(\mathcal{P}_n)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right]\right| \leq \epsilon(n)$$

for every large enough $n \in \mathbb{N}$*.*

If there exists a fixed distribution ensemble $\{\mathcal{P}_n\}_{n\in\mathbb{N}}$ *such that the above holds for every distinguisher* $\mathsf{Dist} \in \mathcal{C}$*, we say that* G *is an* ϵ-strong conditional PRG *against* $\mathcal{C}$*.*

We start with showing that the existence of weak conditional PRG implies the hardness of $\mathrm{Gap_pMK^tP}$ on the uniform distribution, for zero-error algorithms. By extending the result from [San20; Hir18; LP21b; LMP24], we get the following theorem.

Theorem 3.2. *Assume the existence of a* 1/10*-weak conditional PRG* $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ *against* BPP *computable in time* $t(m(n))$*, and let* $d'\colon \mathbb{N} \to \mathbb{N}$ *be such that* $d'(m(n)) = d(n)$*. Then* $(\mathrm{Gap_pMK^tP}[d'(n) + \log n, n-1], \mathbf{U}_n) \notin \mathsf{AvgBPP}$*.*

Proof sketch. Let $n' = m(n)$. By Lemma 2.6, a random n' bit string has $\mathrm{K}^{p(t)}$ complexity at least $n' - 1$ with probability at least 1/2. In contrast, any output of G can be succinctly described by its $d'(n')$-bit seed plus the fixed program (of constant length) that computes G. It follows that $\mathrm{K}^t(G(s)) \leq d'(n') + O(1) \leq d'(n') + \log n'$ for any input s. Thus, if an efficient algorithm could

decide $\mathrm{Gap_pMK^tP}[d'(n') + \log n', n' - 1]$, then it would distinguish a truly uniform string (with complexity near n') from an output of G (with complexity near $d'(n')$).

Assume there exists a zero-error poly-time algorithm such that for every $n \in \mathbb{N}$ $\Pr_{x \leftarrow \{0,1\}^{n'}}[A(x) = \perp] \leq 1/3$. We observe that it must hold for every $s \in \{0,1\}^{d'(n')}$ that $A(G(s))$ indicates that $\mathrm{K}^t(G(s))$ is smaller than $d'(n') + \log n$ (that is, $A(G(s))$ outputs Yes), or it outputs $\perp$. On the other hand, for a truly uniform $z \leftarrow \{0,1\}^{n'}$,

$$\Pr_{z \leftarrow \{0,1\}^{n'}}[A(z) = \perp \vee A(z) = \text{Yes}] \leq 1/3 + \Pr_{z \leftarrow \{0,1\}^{n'}}\left[\mathrm{K}^{p(t)}(z) < n' - 1\right] \leq 1/3 + 1/2.$$

This implies that we can distinguish the output of the PRG from a truly uniform string with advantage $1 - (1/3 + 1/2) = 1/6$, contradicting the security of G. □

We next prove the following result, showing that the existence of weak conditional PRG implies the existence of a strong conditional PRG.

Lemma 3.3 (Weak to strong conditional PRG)**.** *Assume the existence of an ϵ-weak conditional PRG $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ against $\mathsf{BPTIME_U}[T(n)]/a(n)$. Then G is a 3ϵ-strong conditional PRG against $\mathsf{BPTIME_U}[T'(n)]/a'(n)$, where $T'(n) = T(n) - O(a(n))$ and $a'(n) = \Omega(\epsilon^2(n)a(n)/m(n))$.*

We prove Lemma 3.3 next, but first let us prove a corollary.

Corollary 3.4. *Assume that G is an $\epsilon(n)$-weak conditional PRG against $\mathsf{BPP_U}/a(n)$. Then G is a $3\epsilon(n)$-strong conditional PRG against $\mathsf{BPP_U}/a'(n)$, for $a'(n) = \Omega(\epsilon^2(n)a(n)/m(n))$.*

Proof of Corollary 3.4. Lemma 3.3 immediately implies the existence of 3ϵ-strong conditional PRG against $\mathsf{BPTIME_U}[T(n)]/a'(n)$ for any $T(n) \in \text{poly}$. We next claim that this implies that there is a single distribution ensemble $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ w.r.t G is secure against any $T \in \text{poly}$ simultaneously, which implies the corollary. To see this, for every $c \in \mathbb{N}$, let $\mathcal{P}^c = \{\mathcal{P}^c\}_{n\in\mathbb{N}}$ be the ensemble of distributions with respect to which G is secure against $\mathsf{BPTIME_U}[n^c]/a'(n)$. Let $\mathsf{Dist}^c \in \mathsf{BPTIME_U}[n^c]/a'(n)$ be the algorithm with advice of length $a'(n)$ that for every n maximize the distinguishing advantage between $G(\mathcal{P}^c_n)$ and uniform. For every $c \in \mathbb{N}$, let $n_c \in \mathbb{N}$ be the security parameter after which Dist^c cannot distinguish between $G(\mathcal{P}^c_n)$ to uniform with advantage 3ϵ for every $n > n_c$ (promised by the security definition of conditional PRG). Note that by the choice of Dist^c, it follows that there is no algorithm in $\mathsf{BPTIME_U}[n^c]/a'(n)$ that distinguish between $G(\mathcal{P}^c_n)$ and uniform with advantage 3ϵ for any $n > n_c$.

Finally, for every n, let $c_n = \max\{c\colon n_c < n\}$, and let $\mathcal{P}_n = \mathcal{P}^{c_n}_n$. We claim that G is 3ϵ-strong conditional PRG against $\mathrm{BPP}/a'(n)$ with respect to $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$. Indeed, let $\mathsf{Dist} \in \mathrm{BPP}/a'(n)$ a distinguisher that break the security of G. Then $\mathsf{Dist} \in \mathsf{BPP_U}[n^c]/a'(n)$ for some $c \in \mathbb{N}$. But for any $n > n_c$, it holds that $c_n > c$, and therefore no distinguisher with running time $n^{c_n} \geq n^c$ can break the security of $G(\mathcal{P}_n)$. □

We are now ready to prove Lemma 3.3.

Proof of Lemma 3.3. We use the min-max theorem. Fix T and a, let T', a' as defined in Lemma 3.3, and let $a''(n) = a'(n) + c$ for some constant c we will choose later. Let G be a weak conditional

PRG, $n \in \mathbb{N}$, and consider a game between A and B, where A chooses a distribution $\mathcal{P}$ over inputs for G, and B chooses a distinguisher $\mathsf{Dist} \in \mathsf{BPTIME}_{\mathsf{U}}[T'(n)]/a''(n)$. The value of the game is

$$\Pr_{x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right],$$

and B wants to maximize this value. In the following we show that, for a large enough $n \in \mathbb{N}$, when B plays first with any mixed-strategy, A has a strategy that makes the expected value of B at most 3ϵ. More formally, we show that for every distribution $\mathcal{D}$ over $\mathsf{BPTIME}_{\mathsf{U}}[T'(n)]/a''(n)$, there exists a distribution $\mathcal{P}$ over inputs, such that,

$$\mathrm{E}_{\mathsf{Dist} \leftarrow \mathcal{D}}\left[\Pr_{x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right]\right] \leq 3\epsilon(n). \tag{1}$$

By the min-max theorem, it then follows that there exists a distribution $\mathcal{Q}$ over distributions $\mathcal{P}$, such that for any $\mathsf{Dist} \in \mathsf{BPTIME}_{\mathsf{U}}[T'(n)]/a''(n)$,

$$\begin{aligned} &\Pr_{\mathcal{P} \leftarrow \mathcal{Q}, x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right] \\ &= \mathrm{E}_{\mathcal{P} \leftarrow \mathcal{Q}}\left[\Pr_{x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right]\right] \leq 3\epsilon(n). \end{aligned} \tag{2}$$

We conclude that G is a 3ϵ-strong conditional PRG with respect to the marginal distribution of x in the proceed $\{\mathcal{P} \leftarrow \mathcal{Q}; x \leftarrow \mathcal{P}\}$, by noticing that if there exists $\mathsf{Dist} \in \mathsf{BPTIME}_{\mathsf{U}}[T'(n)]/a'(n)$ such that

$$\left|\Pr_{\mathcal{P} \leftarrow \mathcal{Q}, x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right]\right| > 3\epsilon(n),$$

then there exists $\mathsf{Dist}' \in \mathsf{BPTIME}_{\mathsf{U}}[T'(n)]/a''(n)$ such that

$$\Pr_{\mathcal{P} \leftarrow \mathcal{Q}, x \leftarrow \mathcal{P}}[\mathsf{Dist}(G(x)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right] > 3\epsilon(n).$$

We are left to show that Equation (1) holds, when B plays first. Let $\mathcal{D}$ be the distribution chosen by B. Toward the above goal, we construct a distinguisher $\widehat{\mathsf{Dist}} \in \mathsf{BPTIME}_{\mathsf{U}}[T(n)]/a(n)$, such that for every $y \in \{0,1\}^{m(n)}$ it holds that

$$\left|\Pr_{\mathsf{Dist} \leftarrow \mathcal{D}}[\mathsf{Dist}(y) = 1] - \Pr\left[\widehat{\mathsf{Dist}}(y) = 1\right]\right| \leq \epsilon(n). \tag{3}$$

The existence of such a distinguisher would finish the proof, as by assumption, for every $\widehat{\mathsf{Dist}} \in \mathsf{BPTIME}_{\mathsf{U}}[T(n)]/a(n)$ there is a distribution over inputs $\mathcal{P}$ with respect to G is an ϵ-weak conditional PRG against $\widehat{\mathsf{Dist}}$. Together with the triangle inequality, we get that with respect to the same distribution, G is an 3ϵ-indistinguishable from random against $\mathcal{D}$.

To prove the existence of such $\widehat{\mathsf{Dist}}$, we simply take $k(n) = 4m(n)/\epsilon^2(n)$ independent random samples from $\mathcal{D}$. We will show that Equation (3) holds with positive probability. Indeed, for every fixed y, by Chernoff inequality it holds that

$$\Pr_{\mathsf{Dist}_1, \ldots, \mathsf{Dist}_{k(n)} \leftarrow \mathcal{D}}\left[\left|\mathrm{E}_{i \leftarrow [k(n)]}[\mathcal{D}_i(y)] - \mathrm{E}_{\mathsf{Dist} \leftarrow \mathcal{D}}[\mathsf{Dist}(y)]\right| \geq \epsilon\right] < 2^{-m(n)}.$$

By taking a union bound over all possible inputs $y \in \{0,1\}^{m(n)}$, we get that

$$\Pr_{\mathsf{Dist}_1,\ldots,\mathsf{Dist}_{k(n)} \leftarrow \mathcal{D}}\left[\exists y \in \{0,1\}^{m(n)} \ s.t. \ \left|\mathrm{E}_{i \leftarrow [k(n)]}[\mathcal{D}_i(y)] - \mathrm{E}_{\mathsf{Dist} \leftarrow \mathcal{D}}[\mathsf{Dist}(y)]\right| \geq \epsilon\right] < 1.$$

We get that there is a choice of $\mathsf{Dist}_1, \ldots, \mathsf{Dist}_{k(n)}$ such that the distinguisher $\widehat{\mathsf{Dist}}$ that chooses $i \leftarrow [k(n)]$ at random and answers according to Dist_i fulfills Equation (3). We note that by adding the description of $\mathsf{Dist}_1, \ldots, \mathsf{Dist}_{k(n)}$ to the advice of $\widehat{\mathsf{Dist}}$, we get that $\mathsf{Dist} \in \mathsf{BPTIME}_{\mathsf{U}}[T(n)]/a(n)$. □

4 Conditional PRGs from Worst-Case Assumptions

In this section we construct weak conditional PRG from the hardness of approximating rK^t. We next state the main result of this part.

Theorem 4.1. *Assume that for some efficiently computable functions $\ell, s, : \mathbb{N} \to \mathbb{N}$ and for $t \in$ poly with $t(n) \geq n$, $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n)] \notin \mathsf{ioP}/\mathrm{poly}$, for any polynomial p. Then for every efficiently computable function $m(n) \in$ poly there exists a function $G\colon \{0,1\}^{s(n)+\log^4 n} \to \{0,1\}^{m(n)}$ which is an ϵ-weak conditional PRG secure against $\mathsf{BPP}_{\mathsf{U}}/a(n)$ for $a(n) = \ell(n) - O(m^2(n) + \log^4 n)$ and for any $\epsilon \in 1/\mathrm{poly}$.*

We get the following corollary.

Corollary 4.2. *Assume that for some efficiently computable functions $\ell, s\colon \mathbb{N} \to \mathbb{N}$ and $t \in$ poly with $t(n) > n$,*

$$(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n)] \notin \mathsf{ioP}/\mathrm{poly}$$

for any polynomial p. Then there exists an ϵ-conditional PRG

$$G\colon \{0,1\}^{d(n)=s(n)+\log^4 n} \to \{0,1\}^{3d(n)}$$

which is secure against $\mathrm{BPP}/a(n)$ for $a(n) = \Omega((\ell(n)/d(n) - d(n))^{1/3})$ and $\epsilon = 1/a(n)$.

Theorem 1.4 follows from Corollary 4.2 using Lemma 2.9.

Proof of Corollary 4.2. Assume that $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n)] \notin \mathsf{ioP}/\mathrm{poly}$ for any $p \in$ poly. By Theorem 4.1, there exists a weak ϵ'-conditional PRG $G\colon \{0,1\}^{d(n)=s(n)+\log^4 n} \to \{0,1\}^{3d(n)}$ against $\mathsf{BPP}_{\mathsf{U}}/a'(n)$ for

$$a'(n) = \ell(n) - O(d^2(n) + \log^4 n) = \ell(n) - O(d^2(n))$$

for any $\epsilon' \in 1/\mathrm{poly}$.

By Corollary 3.4, G is ϵ-strong conditional PRG against $\mathsf{BPP}_{\mathsf{U}}/a(n)$, for $a(n) = \Omega(\epsilon^2(n)a'(n)/d(n))$, or equivalently,

$$a(n)/\epsilon^2(n) = \Omega(a'(n)/d(n)).$$

The claim follows by taking $a(n) = 1/\epsilon(n) = (\Omega(a'(n)/d(n)))^{1/3}$. □

To prove the theorem, we will make use of extractors with *reconstruction procedure* ([BSW03; Sha04]). A (k, ε)-Extractor is an algorithm $\mathsf{Ext}\colon \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n$, such that for every distribution $\mathcal{X}$ with min-entropy at least k, it holds that $\mathsf{Ext}(\mathcal{X}, \mathbf{U}_d)$ is ε-close in statistical distance to $\mathbf{U}_n$.

Definition 4.3 (Reconstruction Procedure)**.** *An* (ℓ, ε)-reconstruction procedure *for* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a pair of algorithms* (C, R) *such that* $\mathsf{C}\colon \{0,1\}^n \to \{0,1\}^\ell$ *is an* advice function*, and* R *is a (possibly randomized) oracle-aided* reconstruction algorithm *with the following promise: For every string* $x \in \{0,1\}^n$ *and a distinguisher* Dist*, if*

$$|\Pr[\mathsf{Dist}(\mathbf{U}_m) = 1] - \Pr[\mathsf{Dist}(\mathsf{Ext}(x, \mathbf{U}_d)) = 1]| \geq \varepsilon,$$

then $\Pr\left[\mathsf{R}^{\mathsf{Dist}}(\mathsf{C}(x)) = x\right] \geq 2/3$*.*

When R *runs in time* $\mathrm{poly}(n, 1/\epsilon)$ *we will say that* R *is an* efficient reconstruction procedure*.*

Note that we only need R to be efficient. It is well known [Tre99; BSW03] that Ext with (ℓ, ε)-reconstruction procedure is $(\ell + \log 1/\varepsilon, 3\varepsilon)$-Extractor, and therefore the reconstruction procedure is sometimes called a black-box proof ([Sha04]).[11]

We will make use in the following lemma.

Lemma 4.4. *For every efficiently computable function* $m \in \mathrm{poly}$ *there exists an efficiently computable function*

$$\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$$

where $d = O(\log^4 n)$*, such that for every function* $\epsilon\colon \mathbb{N} \to (0,1]$ *with* $\epsilon(n) > n^{-\log n/4}$ Ext *admits an efficient randomized* (τ, ϵ)*-reconstruction procedure for* $\tau = O(m^2 + \log^4 n + \log n \log 1/\epsilon)$ *.*

Moreover, computing $\mathsf{Ext}(x, s)_i$ *can be done in time* $\mathrm{poly}(n, \log m)$ *for every index* i*.*

We prove Lemma 4.4 in Section 4.1, but first let us use it to prove Theorem 4.1.

Proving Theorem 4.1 We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. The proof follows by Lemma 4.4. Let $G\colon \{0,1\}^{s(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ be the function defined by

$$G(\Pi, w) = \mathsf{Ext}(\mathsf{U}(\Pi, 1^{t_1}), w)$$

(where if $\left|\mathsf{U}(\Pi, 1^{t_1})\right| \neq n$, $G(\Pi, w)$ outputs $\perp$). We claim that for every distinguisher $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/a(n)$, there exists a program $\Pi \in \{0,1\}^{s(n)}$, such that Dist does not distinguish between $G(\Pi, \mathbf{U}_d)$ and $\mathbf{U}_m$.

Assume toward a contradiction that this is not the case. That is, there exists a distinguisher $\mathsf{Dist} \in \mathsf{BPTIME}_{\mathsf{U}}[n^c]/a(n)$ for some constant $c \in \mathbb{N}$, such that for infinite many n''s and every $\Pi \in \{0,1\}^{s(n)}$, Dist distinguishes between $G(\Pi, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$ with advantage ϵ for some $\epsilon \in 1/\mathrm{poly}$. By definition of G, this implies that for every string $x \in \{0,1\}^n$ with $\mathrm{K}^{t_1(n)}(x) \leq s(n)$, Dist distinguishes between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$ with advantage ϵ.

Using the reconstruction procedure of Ext, we get that Dist can be used to solve $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[s(n), \ell(n)]$ for some $p \in \mathrm{poly}$ as follows: Consider the algorithm that given $x \in \{0,1\}^n$, estimated the distinguishing advantage of Dist between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$ within an error of $\epsilon/4$ (using $\mathrm{poly}(n, 1/\epsilon)$ samples from $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and from $\mathbf{U}_{m(n)}$, and with a small error probability of 0.1). If this estimation of the distinguishing probability is larger that $\epsilon/2$, Dist answers Yes, while if it is smaller, Dist answers No.

By our assumption of Dist, the above algorithm outputs Yes for any $x \in \{0,1\}^n$ with $\mathrm{K}^{t(n)}(x) \leq s(n)$ with high probability. On the other hand, for every $x \in \{0,1\}^n$ on which the algorithm outputs

[11] And an efficient reconstruction procedure is also equivalent to the notion of strongly black-box PRG [LP23].

Yes with high probability it must holds that Dist distinguishes between $\mathsf{Ext}(x, \mathbf{U}_{d(n)})$ and $\mathbf{U}_{m(n)}$ with advantage at least $\epsilon/4$. Let (C, R) be the reconstruction procedure of Ext, and let $t_2 \in \text{poly}$ be an upper bound on the running time of $\mathcal{R}^{\mathsf{Dist}}$. Consider the (randomized) algorithm $\mathsf{R}^{\mathsf{Dist}}(\mathsf{C}(x))$ (where $\mathsf{C}(x)$ is given to the algorithm as part of its input). Then $\mathsf{R}^{\mathsf{Dist}}(\mathsf{C}(x))$ runs in polynomial time and outputs x with probability at least $1/2$. Moreover, $\mathsf{R}^{\mathsf{Dist}}(\mathsf{C}(x))$ can be described using $a(n) + |\mathsf{C}(x)| + O(1) \leq a(n) + O(m^2(n) + \log^4 n) \leq \ell(n)$. It follows that $\mathsf{rK}^{t_2(n)}(x) < \ell(n)$ for some $t_2(n) = p(t(n)) \in \text{poly}$. □

4.1 Proving Lemma 4.4

To constract Ext, we use the PRG from Nisan and Wigderson [NW94] after encoding $x \in \{0,1\}^n$ using a list-decodable error correcting code, which is a concatenation of Reed-Solomon and Hadamard codes. We start with the definition of combinatorial design.

Definition 4.5 (Design)**.** *A family of sets $\{A_1, \ldots, A_m \subset [n]\}$ is a (n, k, d)-*combinatorial design *if for every $i \in [m]$ $|A_i| = k$, and, for every $i \neq j \in [m]$, $|A_i \cap A_j| \leq d$.*

Lemma 4.6 ([NW94])**.** *For every $d \leq k$ there exist $n \in [k^2, 2k^2]$ and a (n, k, d)-combinatorial design of size $m = O(k^d)$.*

Moreover, the set A_i can be computed in time $\text{poly}(d, k)$ given i, d, k and n.

Proof. Let $A_1, \ldots, A_m$ be the combinatorial design defined by polynomials of degree d: Let $\mathbb{F}$ be a field of size $2^{\lceil \log k \rceil}$. Let $m = (2^{\lceil \log k \rceil})^{(d+1)}$ and let p_i be the i-th degree d polynomial over $\mathbb{F}$ (where we parse the binary representation of $i \in [(2^{\lceil \log k \rceil})^(d+1)]$, $\langle i \rangle \in \{0,1\}^{(d+1)\cdot\lceil \log k \rceil}$ as a vector in $\mathbb{F}^{d+1}$, representing the encoding of the $d+1$ coefficient of p_i). Fix k field elements $\alpha_1, \ldots, \alpha_k$, and let $A_i = \{(j, p_i(\alpha_j)) \colon j \in [k]\} \subseteq [k] \times \mathbb{F}$. Finally, we identify $[k] \times \mathbb{F}$ with the set $[k \cdot |\mathbb{F}|] = [O(k^2)]$. □

Lemma 4.7. *Let $k(n) = 2\lceil \log^2 n \rceil$. There exists an list-decodable error-correcting code $\mathsf{C} \colon \{0,1\}^n \to \{0,1\}^{2^{k(n)}}$ such that given $x \in \{0,1\}^n$ and $i \in [2^{k(n)}]$, $\mathsf{C}(x)_i$ can be computed in time $\text{poly}(n)$.*

Moreover, there exists an $\text{poly}(n, 1/\epsilon)$-time randomized algorithm Enc such that for any $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $\epsilon \geq n^{-\log n/8}$ and a noisy code word $\widehat{c} \in \{0,1\}^{2^{k(n)}}$ the following holds. Assume that for some block $B = [i \cdot 2^{\lceil \log^2 n \rceil}, (i + 10n/\epsilon^6) \cdot 2^{\lceil \log^2 n \rceil}] \subseteq [2^{k(n)}]$ of length $10n^{\lceil \log n \rceil + 1}/\epsilon^6$ it holds that $\mathsf{Ham}(\mathsf{C}(x)_B, \widehat{c}_B) \leq \epsilon \cdot |B|$. Then $\mathsf{Enc}^{\widehat{c}}(1^n, 1^{\lceil 1/\epsilon \rceil}, i)$ outputs a list $\mathcal{L}$ of size at most $O(1/\epsilon^5)$ such that $x \in \mathcal{L}$ with probability at least $2/3$ (over the randomness of Enc).

Proof. We use a concatenation of Reed-Solomon and Hadamard. Let $\mathbb{F}$ be a field of size $2^{\lceil \log^2 n \rceil}$. Phrase $x \in \{0,1\}^n$ as a polynomial p_x of degree $d = \lceil n/\log|\mathbb{F}| \rceil - 1 = O(n/\log^2 n)$. In the following we identify $\mathbb{F}$ with the set $\{0,1\}^{\log|F|}$, and the set $[2^{k(n)}]$ with $\mathbb{F} \times \{0,1\}^{\log|F|}$. Using this notation, we define for $(\alpha, r) \in \mathbb{F} \times \{0,1\}^{\log|F|}$

$$\mathsf{C}(x)_i = \langle p_x(\alpha), r \rangle.$$

We now move to describe the encoding procedure. Fix n, ϵ, x and a noisy code word $\widehat{c} \in \{0,1\}^{2^{k(n)}}$ with $\mathsf{Ham}(\mathsf{C}(x), \widehat{c}) \leq (1/2 - \epsilon)|\mathsf{C}(x)|$. Let $B = [i, i + 10n/\epsilon^6]$ and let $B' = B \times \{0,1\}^{\log|\mathbb{F}|}$. By assumption,

$$\mathsf{Ham}(\mathsf{C}(x)_{B'}, \widehat{c}_{B'}) \leq (1/2 - \epsilon)\big|B'\big|.$$

Then, Enc simply executed the list-decoding procedure for concatenation of Reed-Solomon and Hadamard on $\widehat{c}_{B'}$ to get a list of length $O(1/\epsilon^5)$.

In more detail, for at least $\epsilon/2$ fraction of the $\alpha \in B$, we have that

$$\mathsf{Ham}(\mathsf{C}(x)_{\{\alpha\}\times\{0,1\}^{\log|\mathbb{F}|}}, \widehat{c}_{\{\alpha\}\times\{0,1\}^{\log|\mathbb{F}|}}) \leq (1/2 - \epsilon/2) \cdot |\mathbb{F}|.$$

Thus, for every $\alpha \in B$ we run the Goldreich and Levin [GL89] local list-decoder for the Hadamard code on the corrupted block $\widehat{c}_{\{\alpha\}\times\{0,1\}^{\log|\mathbb{F}|}}$. This returns a list $L_\alpha \subseteq \mathbb{F}$ of size $O(1/\epsilon^2)$ such that, for at least $\epsilon/2$-fraction of $\alpha \in B$, we have $p_x(\alpha) \in L_\alpha$.

Next, we run Reed-Solomon list-recovery (see [GS98; Vad+12]) on the pairs $\{(\alpha, L_\alpha)\}_{\alpha\in B}$: we find all degree-$d$ polynomials $q \in \mathbb{F}[T]$ such that $q(\alpha) \in L_\alpha$ for at least $(\epsilon/2) \cdot |B|$ values of $\alpha \in B$. This yields a list $\mathcal{L} = \{q_1, \ldots, q_t\}$ of size $t = O(1/\epsilon^5)$ that includes p_x.

Finally, from each q_ℓ we reconstruct a candidate message x_ℓ (by reading off its coefficients), and output the list $\{x_\ell\}_{\ell=1}^t$. With probability at least $3/4$ over the decoder's randomness, some x_ℓ equals x.

□

We are now ready to prove Lemma 4.4

Proof of Lemma 4.4. We start by describing the construction of Ext. Fix $n, m(n)$. Let $k(n) = 2\lceil \log^2 n \rceil$ and $d(n) = \log_{k(n)} m(n)$. Let $\{A_1, \ldots, A_{m(n)}\}$ be an (ℓ, k, d)-combinatorial design for $\ell(n) \in [k^2, 2k^2]$. For a string $w \in [\ell(n)]$, let $w(A_i) \in \{0,1\}^{k(n)}$ be the projection of w on the indices in the set A_i. Let $\mathsf{C}\colon \{0,1\}^n \to \{0,1\}^{2^{k(n)}}$ be the code promised by Lemma 4.7. Let $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ be the function defined by $\mathsf{Ext}(x, w)_i = \mathsf{C}(x)_{w(A_i)}$

Reconstruction. Next, we describe the reconstruction procedure for Ext. Fix n and omit it from the notation. Fix $x \in \{0,1\}^n$ and ϵ, and let Dist be a distinguisher that distinguishes the output of $\mathsf{Ext}(x, \mathbf{U}_d)$ and $\mathbf{U}_m$ with distinguishing advantage ϵ. By a standard hybrid argument, there exists an index i and a fixing of $a_{>i} \in \{0,1\}^{m-i}$ such that Dist distinguishes $(\mathsf{Ext}(x, \mathbf{U}_d)_{\leq i}, a_{>i})$ from $(\mathsf{Ext}(x, \mathbf{U}_d)_{<i}, \mathbf{U}_1, a_{>i})$ with advantage ϵ/m. By a distinguishing to prediction (Lemma 2.11), Dist with advice $a \in \{0,1\}^{m+\log m}$ can predict the value of $\mathsf{Ext}(x, \mathbf{U}_d)_i$ given $\mathsf{Ext}(x, \mathbf{U}_d)_{<i}$ with probability at least $1/2 + \epsilon/m$.

By construction of Ext, $\mathsf{Ext}(x, \mathbf{U}_d)_i = \mathsf{C}(x)_{w(A_i)}$. We can thus fix w at all locations outside of A_i in a way that maximize the prediction advantage. Then, for every assignment for $w(A_i)$, we need to be able to compute the prefix $\mathsf{Ext}(x, \mathbf{U}_d)_{<i}$ to apply the predictor. By the properties of the design, each set A_j for $j < i$ intersects with A_i in at most d indices, thus we can add to the advice 2^d possible values it can get.

It follows that there is an algorithm P that given non-uniform advice of length $(m + \log m) + \ell(n) + m \cdot 2^{d(n)} \cdot m(n) \in O(m^2)$, can compute $\mathsf{C}(x)_i$ with probability at least $1/2 + \epsilon/m$ (where the probability is taken over $i \leftarrow [m(n)]$ and the randomness of P.

Next, Partition $[m(n)]$ into $m(n)/(10n \cdot 2^{\lceil \log n \rceil}/\epsilon^4)$ blocks of size $(10n \cdot 2^{\lceil \log n \rceil}/\epsilon^4)$ each, where $B_j = [(i-1)(10n \cdot 2^{\lceil \log n \rceil}/\epsilon^6) + 1, \ldots, j(10n \cdot 2^{\lceil \log n \rceil}/\epsilon^6)]$. By an averaging argument, there exists a block j such that

$$\Pr_{i \leftarrow [m(n)], rand}[\mathsf{P}(i; rand) \neq \mathsf{C}(x)_i] \leq 1/2 - \epsilon/m.$$

By Markov, with probability at least $1 - 1/(1+\epsilon/m) \geq \epsilon/2m$ over the choice of $rand$,

$$\Pr_{i \leftarrow [m(n)]}[\mathsf{P}(i; rand) \neq \mathsf{C}(x)_i] \leq (1/2 - \epsilon/m)(1+\epsilon/m) \leq 1/2 - \epsilon/2m.$$

Let $\epsilon' = \epsilon/2m$

For each such $rand$, by the list-decodability of C, there is an oracle-aided algorithm Dec such that $\mathsf{Dec}^{\mathsf{P}(\cdot;rand)}(j, \epsilon')$ runs in time $\mathrm{poly}(1/\epsilon', n)$ and outputs a list of size $1/\epsilon'^5$ containing x with probability at least $3/4$ (over the internal randomness of Dec). By executing $\mathsf{Dec}^{\mathsf{P}(\cdot,\cdot;rand)}(j, \epsilon')$ nm/ϵ' times with independently chosen randomness $rand$ in each time, we get an randomized algorithm that runs in time $\mathrm{poly}(1/\epsilon', n)$ and outputs a list of size $1/\epsilon'^5 \cdot nm/\epsilon' = nm/\epsilon'^6$, which contains x with overwhelming probability. Moreover, the advice needed to execute $\Pi'() = \mathsf{Dec}^{\mathsf{P}(\cdot;rand)}(j, \epsilon')$ is of length $O(m^2)$.

Finally, we let Π be the program that additionally to Π', has a hash value of x, $h(x)$, such that with high probability on the randomness of Π', there is no $x' \neq x$ in the list found by Π' with $h(x) = h(x')$. Then Π we take Π to be simply the program that executes Π' to get a list of candidates, and then output one of the elements x' in the list with $h(x') = h(x)$. With high probability, we get that $x' = x$. We left to show that this can be done in a way that the description of h and $h(x)$ is short. We can do it using almost-universal hashing.

For example, let $E \colon \{0,1\}^n \to \Sigma^{2n}$ be an error correcting code with distance $1/2$. This can be done using Reed-Solomon with $|\Sigma| = O(n)$. Let $t = \log(nm/\epsilon'^6) + \log n$. Let $\mathcal{H}$ be the hash family where a function $h_{i_1,\ldots,i_k} \in \mathcal{H}$ is described using k indices in $[2n]$, where $h_{i_1,\ldots,i_k}(x) = E(x)_{i_1}, \ldots, E(x)_{i_k}$. Then $h_{i_1,\ldots,i_k}, h_{i_1,\ldots,i_k}(x)$ can be described using $O(k \log n) = O(\log^2 n + \log n \log 1/\epsilon')$ bits.

Moreover, for over $x \neq x'$, the probability that for a random $h_{i_1,\ldots,i_k} \leftarrow \mathcal{H}$, $h_{i_1,\ldots,i_k}(x) = h_{i_1,\ldots,i_k}(x')$ is at most $1/2^k = 1/n \cdot 1/(nm/\epsilon'^6)$. By the union bound, for any fixed list L of size at most (nm/ϵ'^6), the probability that L contains $x \neq x'$ with $h_{i_1,\ldots,i_k}(x) = h_{i_1,\ldots,i_k}(x')$ is at most $1/n$. Thus, by an averaging argument, there is a fixing of $h_{i_1,\ldots,i_k}$, such that with probability at least $1 - 1/n$ over the randomness of Π', Π will output x. □

5 Hardness of DisjNP

In this part we formally prove the results related to the hardness of DisjNP. recall that a promise problem $\Pi = (\mathcal{Y}, \mathcal{N})$ is in DisjNP if both $\mathcal{Y} \in \mathrm{NP}$ and $\mathcal{N} \in \mathrm{NP}$.

We next prove the following theorem, basing the hardness of DisjNP on the existence of conditional PRGs.

Theorem 5.1. *Assume that there exists a constant $\epsilon > 0$ such that $\mathrm{E} \not\subseteq \mathrm{ioNTIME}[2^{kn}]/2\epsilon n$ for every $k \in \mathbb{N}$. There exists a constant c such that the following holds. Let $\delta > 0$ be a constant, $d \colon \mathbb{N} \to \mathbb{N}$ be a function such that $n^\delta \leq d(n)$ and $\lceil c \log n \rceil + 2d(n) + 1 \leq n$, and let $m(n) = (\lceil c \log n \rceil + 2d(n) + 1)^{c+1}$. Assume there exists a $1/4$-weak conditional PRG $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ against $\mathsf{BPTIME}_{\mathsf{U}}[T(m(n))]/a(m(n))$. Then $\mathsf{DisjNP} \not\subseteq \mathsf{BPTIME}_{\mathsf{U}}[T(n)]/(a(n) - O(1))$.*

Proof. We use Naor's commitment [Nao91] scheme together with derandomization. Let c be the constant from Theorem 2.3. Let $b(n) = \lceil c \log n \rceil + 2d(n) + 1$, and let $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{(b(n))^c \cdot b(n)}$ be a weak conditional PRG, computable in time $t(b(n))$. and let $G_{NW} \colon \{0,1\}^{\lceil c \log b(n) \rceil} \to \{0,1\}^{b(n)}$ be a PRG against $\mathrm{NTIME}[t(b(n))]/b(n)$ promised by Theorem 2.3. Consider the promise problem $\Pi = (\mathcal{Y}, \mathcal{N})$ defined by:

- $\mathcal{Y} = \left\{ G(x) \colon x \in \{0,1\}^{d(n)} \right\}$
- $\mathcal{N} = \left\{ G(x) \oplus (G_{NT}(0^{\lceil c \log b(n) \rceil}) \circ \cdots \circ G_{NT}(1^{\lceil c \log b(n) \rceil})) \colon x \in \{0,1\}^{d(n)} \right\}$

We start by showing that $\mathcal{Y}$ and $\mathcal{N}$ are disjoint. For $i \in [2^{\lceil c \log b(n) \rceil}]$, let $G_i(x)$ be the i-th $b(n)$-bit long block $G(x)$. Consider the set $\mathcal{S} = \left\{ y \in \{0,1\}^{b(n)} \colon \exists i, x, x' \text{ s.t. } G_i(x) \oplus G_i(x') = y \right\}$. Then

$$|\mathcal{S}| \leq \left| \left\{ i \in [b(n)^c], x, x' \in \{0,1\}^{d(n)} \right\} \right| = 2^{2d(n) + c \log b(n)} \leq 2^{b(n)}/2.$$

It follows that with probability at least $1/2$ over the choice of a random string y, there is no i, x and x' such that $G_i(x) = G_i(x') \oplus y$. Let Dist be the non-deterministic distinguisher that given y checks if there are such i, x, x'. Then Dist runs in time $O(t(b(n)))$ and answers 'No' with probability at least $1/2$ over the choice of a random y. Thus by the security of G_{NW}, Dist answer 'No' with probability at least $1/3$ also over a random output of G_{NW}. We conclude that there exist some i, such that for any x_i, x_i', it never holds that $G(x_i) = G(x_i') \oplus G_{NW}(i)$. Thus, $\mathcal{Y}$ and $\mathcal{N}$ must be disjoint.

Next, we observe that both $\mathcal{Y}$ and $\mathcal{N}$ are in NP, as G, G_{NW} are efficiently computable. Thus we get that $\Pi \in \mathsf{DisjNP}$.

Lastly, we prove the hardness of Π. Assume there is an algorithm $\mathsf{A} \in \mathsf{BPTIME}_{\mathsf{U}}[T(n')]/a(n')$ that decides Π, for input lengths $n' = m(n)$. We claim that we can use A to distinguish between $G(\mathcal{D}_n)$ and uniform for any distribution $\mathcal{D}$, contradicting the security of G. Indeed,

$$\Pr[\mathsf{A}(G(\mathcal{D}_n)) = Yes] = 1, \quad \textit{and} \quad \Pr\left[\mathsf{A}(G(\mathcal{D}_n) \oplus (G_{NT}(0^{\lceil c \log b(n) \rceil}) \circ \cdots \circ G_{NT}(1^{\lceil c \log b(n) \rceil}))) = Yes\right] = 0$$

but

$$\Pr\left[\mathsf{A}(\mathbf{U}_{m(n)}) = Yes\right] = \Pr\left[\mathsf{A}(\mathbf{U}_{m(n)}) \oplus (G_{NT}(0^{\lceil c \log b(n) \rceil}) \circ \cdots \circ G_{NT}(1^{\lceil c \log b(n) \rceil}))) = Yes\right]$$

Which implies that either A distinguishes between $G(\mathcal{D}_n)$ and the uniform distribution, or $\mathsf{A}'(y) = \mathsf{A}(y \oplus (G_{NT}(0^{\lceil c \log b(n) \rceil}) \circ \cdots \circ G_{NT}(1^{\lceil c \log b(n) \rceil})))$ does. □

The next corollary, which is the main theorem of this part, shows that DisjNP is hard if $\mathrm{Gap_pMK^tP}$ is hard.

Corollary 5.2. *Assume that there exists a constant $\epsilon > 0$ such that $\mathrm{E} \not\subseteq \mathrm{ioNTIME}[2^{kn}]/2\epsilon n$ for every $k \in \mathbb{N}$. Then there exists a constant c such that the following holds. Assume that for some constant $\alpha > 0$, $\mathrm{Gap_pMK^tP}[n^{\alpha/4c}, n^{\alpha}] \notin \mathsf{ioP/poly}$ for any polynomial p. Then $\mathsf{DisjNP} \not\subseteq \mathrm{BPP}$*

We note that here the same conclusion holds from the assumption that $\mathrm{Gap_pMK^tP}[n^{\alpha/4c}, n^{\alpha}] \not\subseteq \mathrm{P/poly}$.

Proof. By Lemma 2.9, the hardness of $\mathrm{Gap_pMK^tP}$ implies the hardness of $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})$ with a small loss of $O(\log n)$ in the gap. By Theorem 4.1, there exists a weak conditional PRG $G \colon \{0,1\}^{2n^{\alpha/4(c+1)}} \to \{0,1\}^{n^{\alpha/2}/2}$ against $\mathsf{BPP}_{\mathsf{U}}/(n^{\alpha}/2)$. By Theorem 5.1, this implies that $\mathsf{DisjNP} \not\subseteq \mathsf{BPP}_{\mathsf{U}}/\omega(1)$. □

Remark 5.3 (On Turing reductions from $\mathrm{Gap_pMK^tP}$ to DisjNP)**.** *We note that our proof gives a candidate poly-time Turing reduction from* $\mathrm{Gap_pMK^tP}[n^{\alpha/4c}, n^{\alpha}]$ *to* DisjNP*. Indeed, given an oracle to an algorithm that decides the above defined promise problem* Π*, and an input* x *to* $\mathrm{Gap_pMK^tP}$*, the algorithm can query the oracle on either* $G(x)$ *or on* $G(x) \oplus (G_{NT}(0^{\lceil c \log b(n) \rceil}) \circ \cdots \circ G_{NT}(1^{\lceil c \log b(n) \rceil}))$*, and to check if the oracle returns the right answer.*

However, the correctness of the reduction is only promised assuming the oracle can be implemented by an efficient algorithm.

5.1 DisjNP, TFUP, and worst-case hardness

The next lemma, states that the hardness of TFUP implies the hardness of DisjNP.

Lemma 5.4 (TFUP to DisjNP)**.** *Assume that* $\mathsf{TFUP} \nsubseteq \mathsf{ioP/poly}$*. Then* $\mathsf{DisjNP} \nsubseteq \mathsf{ioP/poly}$*.*

Proof. Let $\mathcal{L} \in \mathsf{TFUP}$ be a language such that $\mathcal{L} \notin \mathsf{ioP/poly}$. Let

$$\mathcal{Y} = \{(x, i) \colon (x, w) \in \mathcal{R}_{\mathcal{L}}, w_i = 1\},$$

and

$$\mathcal{N} = \{(x, i) \colon (x, w) \in \mathcal{R}_{\mathcal{L}}, w_i = 0\}.$$

We claim that $\Pi = (\mathcal{Y}, \mathcal{N}) \notin \mathsf{ioP/poly}$. Indeed, given an algorithm A that decides Π, and an input $x \in \{0,1\}^n$ we can efficiently find the witness w for x. Moreover, by construction $\Pi \in \mathsf{DisjNP}$. □

Finally, we prove that the worst-case hardness of DisjNP implies its average-case hardness on inefficiently samplable distribution.

Lemma 5.5. *There exists a constant* c *such that the following holds. Let* $\epsilon \colon \mathbb{N} \to (0, 1/2]$ *and* $a \colon \mathbb{N} \to \mathbb{N}$ *be functions, and let* $r(n) = n/\epsilon^2(n)$*. Assume that for a promise problem* Π *it holds that* $\Pi \notin \mathrm{io}\,\mathsf{BPP_U}/(a(n) \cdot r(n) + c)$*. Then there exists a distribution ensemble* $\mathcal{P} = \{\mathcal{P}_n\}_{n \in \mathbb{N}}$ *over* $\mathcal{Y} \cup \mathcal{N}$*, such that*

$$\Pr_{x \leftarrow \mathcal{P}_n}[\mathsf{D}(x) = \Pi(x)] \leq 1/2 + \epsilon(n)$$

for any $\mathsf{BPP_U}/a(n)$ *and for any large enough* $n \in \mathbb{N}$*.*

Proof. We prove the existence of an hard distribution against $\mathsf{TIME_U}[n^c]/a(n)$, for any constant $c \in \mathbb{N}$. The lemma then follows by a similar proof to the one of Corollary 3.4. We use the min-max theorem. For $n \in \mathbb{N}$, consider a game (A, B) in which A chooses a distinguisher in $\mathsf{TIME_U}[n^c]/a(n)$, and B chooses an input $x \in \{0,1\}^n$. The value of the game is 1 if $\mathsf{D}(x) = \Pi(x)$, or 0 otherwise, and A wants to maximize this value.

In the following we want to show that for a large enough $n \in \mathbb{N}$, when A plays first with a mixed strategy, there is a strategy for B such that B wins with probability at least $1/2 + \epsilon$. Formally, we claim that for any distribution $\mathcal{D}$ over algorithms in $\mathsf{TIME_U}[n^c]/a(n)$, there exists an input $x \in \{0,1\}^n$, such that,

$$\Pr_{\mathsf{D} \leftarrow \mathcal{D}}[\mathsf{D}(x) = \Pi(x)] \leq 1/2 + \epsilon. \tag{4}$$

Indeed, if Equation (4) holds, then by the minimax theorem, there exists a distribution $\mathcal{P}$ such that for any $\mathsf{D} \in \mathsf{TIME_U}[n^c]/a(n)$, it holds that,

$$\Pr_{x \leftarrow \mathcal{P}}[\mathsf{D}(x) = \Pi(x)] \leq 1/2 + \epsilon. \tag{5}$$

To show Equation (4), let $\mathcal{D}$ be the distribution over $\mathsf{TIME}_{\mathsf{U}}[n^c]/a(n)$ that A chooses. Our goal is to construct a distinguisher $\widehat{\mathsf{D}} \in \mathsf{BPP}_{\mathsf{U}}/(a(n) \cdot r(n) + c)$, such that

$$\left| \Pr\left[\widehat{\mathsf{D}}(x) = 1\right] - \Pr_{\mathsf{D} \leftarrow \mathcal{D}}[\mathsf{D}(x) = 1] \right| < \epsilon/2$$

for any $x \in \mathcal{Y} \cup \mathcal{N}$. In this case, since by the assumption on the hardness of Π B can find an input x for which $\widehat{\mathsf{D}}(x) \neq \Pi(x)$, it must holds that $\Pr_{\mathsf{D} \leftarrow \mathcal{D}}[\mathsf{D}(x) = \Pi(x)] \leq 1/2 + \epsilon$, as otherwise using standard amplification we get that $\Pi \in \mathsf{BPP}_{\mathsf{U}}/(a(n) \cdot r(n) + c)$.

We next construct $\widehat{\mathsf{D}}$. Let $r = 10n/\epsilon^2$, and let $\mathsf{D}_1, \ldots, \mathsf{D}_r$ be r independent random samples from $\mathcal{D}$. Let $\widehat{\mathsf{D}}(x) = \mathsf{Majority}(\mathsf{D}_1(x), \ldots, \mathsf{D}_r(x))$ (breaking ties arbitrarily). By the Chernoff bound we have that for a fixed $x \in \{0,1\}^n$

$$\Pr_{\mathsf{D}_1, \ldots \mathsf{D}_r \leftarrow \mathcal{D}}\left[\left| \mathrm{E}_{i \leftarrow [r]}[\mathsf{D}_i(x) = 1] - r(n) \cdot \Pr_{\mathsf{D} \leftarrow \mathcal{D}}[\mathsf{D}(x) = 1] \right| \geq \epsilon/2 \right] < 2^{-n}.$$

Thus, by the union bound, there is a fixing of $\mathsf{D}_1, \ldots \mathsf{D}_r$, such that for every $x \in \{0,1\}^n$,

$$\left| \mathrm{E}_{i \leftarrow [r]}[\mathsf{D}_i(x) = 1] - r(n) \cdot \Pr_{\mathsf{D} \leftarrow \mathcal{D}}[\mathsf{D}(x) = 1] \right| < \epsilon/2.$$

Thus we can take $\widehat{\mathsf{D}}$ to be the algorithm that on input x, chooses $i \leftarrow [r]$ at random, and answer according to $\mathsf{D}_i(x)$. □

6 Distribution-Aided Zero-Knowledge

In this section we define and construct distribution-aided zero-knowledge proofs. We start with the definition of distribution-aided commitments scheme.

Definition 6.1 (Distribution-aided commitment scheme)**.** *an* ϵ-distribution-aided commitment scheme *with respect to a (possibly not efficiently samplable) distribution* $\mathcal{D}$ *against class* $\mathcal{C}$ *of receivers is a two-party protocol* $(\mathsf{Com}^{\mathcal{D}}, \mathsf{Rec})$ *between efficient oracle-aided algorithm* Com *and an efficient algorithm* Rec*. The protocol on input* 1^n *has two stages.*

- **Commit**. Com *has a private input* $b \in \{0,1\}$*, a private randomness* $r \in \{0,1\}^*$*, and a sample* $w \leftarrow \mathcal{D}_n$*, and* Rec *has private randomness. At the end of the stage, both parties output a commitment* z *as a common output.*

- **Reveal**. *Both parties receive as input a commitment* z*.* Com *additionally gets its input and randomness* b, r, w*.* Com *sends to* Rec *a message* m*, and* Rec *either outputs a bit* b *or reject and outputs* $\perp$*.*

A commitment scheme is statistically binding if the following hold.

- **Completeness.** *If both parties are honest, then for any bit* $b \in \{0,1\}$ *and with probability* $1 - \mathrm{neg}(n)$ *over the randomness and samples of* Com *and* Rec*, at the end of the reveal stage* Rec *outputs* b*.*

  If the above holds with probability 1*, we say* $(\mathsf{Com}, \mathsf{Rec})$ *have perfect completeness.*

- **Statistical binding**. *Any (possibly inefficient) algorithm* $\widehat{\mathsf{Com}}$ *wins the following game with negligible probability:*
  - $\widehat{\mathsf{Com}}(1^n)$ *interact with* $\mathsf{Rec}(1^n)$ *in the commit stage to get a common output* z.
  - $\widehat{\mathsf{Com}}$ *wins if its generates two messages* m_0 *and* m_1, *where* $\mathsf{Rec}(m_0, z) = 0$ *and* $\mathsf{Rec}(m_1, z) = 1$
- ϵ**-Hiding against** $\mathcal{C}$. *For any large enough* n, *and* $\widehat{\mathsf{Rec}} \in \mathcal{C}_n$ *the following holds. For* $b \in \{0,1\}$, *let* p_b *be the probability that at the end of the commit stage of the interaction* $(\mathsf{Com}(b), \widehat{\mathsf{Rec}})(1^n)$, $\widehat{\mathsf{Rec}}$ *outputs* 1. *Then* $|p_0 - p_1| \leq \epsilon(n)$.

We next define distribution-aided zero-knowledge proofs. For a two-party protocol (P, V) we use $\mathrm{View}_{\mathsf{V}}((\mathsf{P}, \mathsf{V})(x))$ to denote the view of V in a random execution of (P, V) with common input x. In the following for an algorithm $\widehat{V}$ that get auxiliary-input z, we let $\widehat{V}_z$ be the algorithm $\widehat{V}$ that gets z as auxiliary-input.

Definition 6.2 (Distribution-aided Zero-Knowledge)**.** *A two party protocol* (P, V) *is a* distribution-aided zero-knowledge with respect to a distribution $\mathcal{D}$ *for an* NP *relation* $\mathcal{R}$ *if* V *is efficient, and:*

- α**-Completeness**. *For every* $(x, w) \in \mathcal{R}$ *and security parameter* $\lambda \in \mathbb{N}$, V *accepts with probability at least* $1 - \alpha(\lambda)$ *on the interaction* $(\mathsf{P}^{\mathcal{D}}(w), \mathsf{V})(1^\lambda, x)$.
- β**-Soundness**. *For every* $x \notin \mathcal{L}(\mathcal{R})$ *and every malicious prover* $\widehat{\mathsf{P}}$, V *rejects with probability at least* $\beta(\lambda)$ *on the interaction* $((\widehat{\mathsf{P}}, \mathsf{V})(1^\lambda, x)$.
- (a, ϵ)**-Zero-Knowledge against** $\mathcal{C}$. *For every algorithm* $\widehat{\mathsf{V}} \in \mathcal{C}$, *there exists an efficient oracle-aided simulator* $\mathsf{Sim}_{\mathsf{V}}$, *such that for any* $(x, w) \in \mathcal{R}$ *and auxiliary-input* $z \in \{0,1\}^{a(\lambda)}$, $\mathsf{Sim}_{\mathsf{V}}^{\mathcal{D}}(1^\lambda, x, z)$ *uses at most* $\mathrm{poly}(|x|, \lambda)$ *samples from* $\mathcal{D}$. *Moreover, for every ensemble* $\{(x_\lambda, w_\lambda, z_\lambda) \in \mathcal{R}\}_{\lambda \in \mathbb{N}}$, $\left\{\mathsf{Sim}_{\mathsf{V}_z}^{\mathcal{D}}(1^\lambda, x_\lambda, z_\lambda)\right\}_\lambda$ *is* $\epsilon(\lambda)$*-indistinguishable against any distinguisher* $\mathsf{Dist} \in \mathcal{C}$ *from*

$$\left\{\mathrm{View}_{\widehat{\mathsf{V}}}((\mathsf{P}^{\mathcal{D}}(w_\lambda), \widehat{\mathsf{V}}_z)(1^\lambda, x_\lambda))\right\}_\lambda.$$

We say that (P, V) *has an* efficient prover *if* P *is efficient.*

6.1 From Commitment to Zero-Knowledge

The main result in this part is a construction of distribution-aided zero-knowledge proof for NP from distribution-aided commitments, stated in the next theorem.

Theorem 6.3. *Let* $\epsilon \in (0,1)$ *be a constant. Assume that there exists an* $1/(n^\epsilon d(n))$*- distribution-aided commitment scheme* $(\mathsf{Com}, \mathsf{Rec})$ *with respcet to a distribution* $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ *over* $\{0,1\}^{d(n)}$ *that is secure against* $\mathsf{BPP}_{\mathsf{U}}/a(n)$ *for* $a(n) \geq n^\epsilon d(n)$. *Then for any* $q \in \mathrm{poly}$ *there exists a distribution-aided* $1 - \mathrm{neg}(\lambda)$*-correct,* $2^{-\lambda}$*-sound, efficient-prover,* $(q(\lambda), 1/q(\lambda))$*-zero knowledge proof for* NP *against distinguishers in* $\mathsf{BPP}_{\mathsf{U}}/q(\lambda)$.

Moreover, if $(\mathsf{Com}, \mathsf{Rec})$ *have perfect completeness then so do the zero-knowledge proof.*

To prove Theorem 6.3, we implement the zero-knowledge proof for Hamiltonian Cycle ($\mathsf{HamiltonianCycle}$) from [Blu86]. In the following we identify a graph G over k vertices with its adacency matrix M^G, such that $M^G_{i,j} = 1$ if (i,j) is an edge in G. A witness that a graph $G \in \mathsf{HamiltonianCycle}$ is a vector $(v_1, \ldots, v_k) \in [k]^k$ such that for every $i \in [k]$, $M^G_{v_i, v_{i+1 \bmod k}} = 1$, and for every $j \in [k]$ there is $i \in [k]$ such that $v_i = j$. We start with the description of the protocol.

Protocol 6.4 (Zero-Knowledge for Hamiltonian Cycle)**.**

Common input: Graph $G \in \{0,1\}^{k \times k}$*, security parameter* 1^λ*.*

Prover's input: A witness $w = (v_1, \ldots, v_k)$ *representing an Hamiltonian cycle in* G*.*

Oracle: A distribution-aided commitment scheme $(\mathsf{Com}^{\mathcal{D}}, \mathsf{Rec})$*.*

Operation:

1. *Set* $n = (10q(\lambda) \cdot 2\lambda \cdot k^2)^{1/\epsilon}$*.*
2. *Repeat for* 2λ *rounds:*
   (a) *The prover chooses a random permutation* $\pi : [k] \to [k]$*, and compute on* $G' = \pi(G)$*.*
   (b) *The prover and verifier interact using* $(\mathsf{Com}^{\mathcal{D}}, \mathsf{Rec})$*. The prover commit to the adjacency matrix of* G'*, each entry separately.*
   (c) *The verfier randomly choose* $b \leftarrow \{0,1\}$*, and sends* b *to the prover.*
   (d) *If* $b = 0$*, the prover opens all of the commitments, and sends* π *to the verifier. The verifier verifies that the commitment is a valid commitment to* $G' = \pi(G)$*. If not, the verifier rejects and abort.*
   (e) *If* $b = 1$*, the prover opens* k *bits corresponds to edges on an Hamiltonian cycle in* G'*. The verifier verifies that all the bits are opened to* 1*, and the induced graph is indeed an Hamiltonian cycle. If not, the verifier rejects.*
3. *The verifier accepts.*

We next analyze the completeness, soundness and zero-knowledge of the protocol.

6.1.1 Completeness.

Claim 6.5. *When* $(G, w) \in \mathcal{R}_{\mathsf{HamiltonianCycle}}$*, the verifier accepts with probability* $1 - \mathrm{neg}(\lambda)$ *in Protocol 6.4.*

When $(\mathsf{Com}, \mathsf{Rec})$ *have perfect completeness the verifier accepts with probability* 1*.*

Proof. The completeness follows by construction. When $b = 0$, the verifier always accepts the honest prover message by the up to the correctness error of the commitment scheme. When $b = 1$, the prover given an Hamiltonian cycle $(v_1, v_2, \ldots, v_k)$ in G, can open the cycle $(\pi(v_1), \ldots, \pi(v_k))$ in G'. By construction of G', there is always an edge between $\pi(v_i)$ to $\pi(v_{i+1})$. □

6.1.2 Soundness.

Claim 6.6. *For any* $G \notin \mathsf{HamiltonianCycle}$*, for any malicious prover* $\widehat{\mathsf{P}}$*,* V *accepts with probability at most* $(1/2 - \mathrm{neg}(\lambda))^{2\lambda} \leq 2^{-\lambda}$ *in the interaction* $(\widehat{\mathsf{P}}, \mathsf{V})(1^\lambda, G)$*.*

Proof. By the statistical binding property of the commitment, any malicious prover cannot convince the verfier to accept with probability better than $1/2 + \mathrm{neg}(\lambda)$ in each round of the protocol. Indeed, a random execution of the first step of the protocol above. With probability at least $1 - k^2 \cdot \mathrm{neg}(n) \geq 1 - \mathrm{neg}(\lambda)$ over the randomness of the prover and the verifier, all of the values the

prover open to the commitments when $b = 1$ are consistent with the values it opens to when $b = 0$. In this case, if the verifier is convinced when both $b = 0$ and $b = 1$, then the opened values in the case $b = 1$ are an Hamiltonian cycle in the graph $G' = \pi(G)$, which translates to an Hamiltonian cycle in G. □

6.1.3 Zero-Knowledge.

Finally, we prove that the protocol is zero knowledge. In the following we ignore the auxiliary-input of the verifier, and simply look on it as part of the advice. This implies our theorem as the simulator we construct next is actually black-box (that is, the same simulator work for any verifier $\widehat{V}$ given oracle-access to $\widehat{V}$). Let $\widehat{\mathsf{V}}$ be an efficient (possibly malicious) verifier, and let $\mathsf{Sim}_{\widehat{\mathsf{V}}}^{\mathcal{D}}$ be the following algorithm:

Algorithm 6.7 (Sim)**.**

Input: Graph $G \in \{0,1\}^{k \times k}$*, security parameter* 1^λ*.*

Oracle: Distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$*, malicious verifier* $\widehat{\mathsf{V}}$*.*

Operation:

1. *Repeat for* 2λ *rounds* $i = 1, \ldots, 2\lambda$*:*
   (a) *Repeat* 2λ *times:*
      i. *Choose a random bit* $b \leftarrow \{0,1\}$*, and let* $\pi \colon [k] \to [k]$ *be a random permutation.*
      ii. *If* $b = 0$*, let* $G' = \pi(G)$*. If* $b = 1$*, let* G'' *be the graph that (only) contains the Hamiltonian cycle* $(1, \ldots, k)$*, and let* $G' = \pi(G'')$*.*
      iii. Sim *simulates an interaction with* $\widehat{\mathsf{V}}$ *in the* i*-th round to commit on the adjacency matrix of* G'*.*
      iv. $\widehat{\mathsf{V}}$ *sends a bit* b' *to the prover. If* $b' \neq b$*, the simulator rewind* $\widehat{V}$ *to the beginning of the round, and return to Step 1a. Otherwise, if* $b = 0$*, the provers open the commitment to* G'*, and if* $b = 1$*, the prover opens the commitment to the edges of the Hamiltonian cycle* $(\pi(1), \ldots, \pi(k))$*.*
      v. *The simulation continue to the next round.*
   (b) *If no iteration succeed, abort.*
2. Sim *outputs the transcript and the randomness of* $\widehat{\mathsf{V}}$ *in the end of the interaction above.*

To analyze Sim, we also consider a second simulator, $\widehat{\mathsf{Sim}}$ that gets the witness as an input.

Algorithm 6.8 ($\widehat{\mathsf{Sim}}$)**.**

Input: Graph $G \in \{0,1\}^{k \times k}$*, security parameter* 1^λ*, and a witness* $w = (v_1, \ldots, v_k)$ *representing an Hamiltonian cycle in* G*..*

Oracle: Distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$*, malicious verifier* $\widehat{\mathsf{V}}$*.*

Operation:

1. *Repeat for* 2λ *rounds* $i = 1, \ldots, 2\lambda$*:*

(a) Repeat 2λ times:

i. Let $b \leftarrow \{0,1\}$ be a random bit. If $b = 1$, the simulator rewind $\widehat{V}$ to the beginning of the round, and return to Step 1a. Otherwise:

ii. Let $\pi\colon [k] \to [k]$ be a random permutation.

iii. Let $G' = \pi(G)$.

iv. Simulate an interaction with $\widehat{\mathsf{V}}$ to commit on the adjacency matrix of G', where:
$\widehat{\mathsf{V}}$ *sends a bit b' to the prover. If $b' = 0$, the provers open the commitment to G', and if $b' = 1$, the prover opens the commitment to the edges of the Hamiltonian cycle $(\pi(v_1), \ldots, \pi(v_k))$.*

v. The simulation continues to the next round.

(b) If no iteration succeed, abort.

2. $\widehat{\mathsf{Sim}}$ outputs the transcript and the randomness of $\widehat{\mathsf{V}}$ in the end of the interaction above.

We first observe that the output of $\widehat{\mathsf{Sim}}$ is statistically close to the view of $\widehat{V}$ in the real interaction.

Claim 6.9. *For every $\lambda > 1, G, w$, $\mathrm{SD}(\widehat{\mathsf{Sim}}(G, 1^\lambda, w), \mathrm{View}_{\widehat{\mathsf{V}}}((\mathsf{P}(w), \mathsf{V})(1^\lambda, G)) \leq 2^{-\lambda}$.*

Proof. We note that when $\widehat{\mathsf{Sim}}$ does not abort, the distribution of the output of $\widehat{\mathsf{Sim}}$ is exactly as in the real interaction. The probability of $\widehat{\mathsf{Sim}}$ to abort at the i-th round is $(1/2)^{2\lambda}$, and thus the probability to abort in any round is $2\lambda \cdot (1/2)^{2\lambda} \leq 2^{-\lambda}$. □

The zero-knowledge proof continues with showing that the output of Sim is indistinguishable from this of $\widehat{\mathsf{Sim}}$. In the following, fix a security parameter λ, an input G and a witness w, and assume towards a contradiction there is a distinguisher $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/q(\lambda)$ that can distinguish $\mathsf{Sim}(G)$ from $\widehat{\mathsf{Sim}}(G, w)$ with advantage $1/q(\lambda)$. The proof is by an hybrid argument over the total $(2\lambda)^2$ iterations. We define $4\lambda^2 + 1$ algorithms Sim_i, where Sim_0 is $\widehat{\mathsf{Sim}}$, $\mathsf{Sim}_{4\lambda^2}$ is Sim, and for every $i \in [4\lambda^2]$, Sim_i is the algorithm that acts like Sim in the first i iterations, and acts like $\widehat{\mathsf{Sim}}$ in iterations $i + 1, \ldots, 4\lambda^2$.

By a simple hybrid argument, there is an iteration i, such that Dist can distinguish between $\mathsf{Sim}_{i-1}(1^\lambda, G, w)$ and $\mathsf{Sim}_i(1^\lambda, G, w)$ with advantage $1/(4\lambda^2 q(\lambda))$.

We next consider 3 additional hybrids, in all of them all of the iterations but the i-th are defined exactly like in $\mathsf{Sim}_{i-1}(1^\lambda, G, w)$. We describe the i-th iteration in each of them.

H_i^1. In the first hybrid, we let Sim commit to the real Hamiltonian cycle when $b = 1$. In particular,the i-th iteration of H_i^1 is defined as follows.

- Choose a random bit $b \leftarrow \{0,1\}$, and let $\pi\colon [k] \to [k]$ be a random permutation.
- If $b = 0$, let $G' = \pi(G)$. If $b = 1$, let G'' be the graph that (only) contains the Hamiltonian cycle $(v_1, \ldots, v_k)$, and let $G' = \pi(G'')$.
- Simulate an interaction with $\widehat{\mathsf{V}}$ to commit on the adjacency matrix of G'.

- $\widehat{\mathsf{V}}$ sends a bit b' to the prover. If $b' \neq b$, abort and output $\perp$. Otherwise, if $b = 0$, open the commitment to G', and if $b = 1$, the open the commitment to the edges of the Hamiltonian cycle $(\pi(v_1), \ldots, \pi(v_k))$.
- Output the transcript and the randomness of $\widehat{\mathsf{V}}$ in the interaction above.

Claim 6.10. $\mathsf{Sim}_{i-1}(1^\lambda, G, w) \equiv H_i^1$

Proof. Since π is a random permutation, the distribution of $(\pi(v_1), \ldots, \pi(v_k))$ is exactly the same as $(\pi(1), \ldots, \pi(k))$. □

H_i^2. In the second hybrid, we let the prover commit to the real graph G both when $b = 0$ and when $b = 1$. In particular, the ith iteration of H_i^2 is the output distribution of the following process:

- Choose a random bit $b \leftarrow \{0, 1\}$, and let $\pi \colon [k] \to [k]$ be a random permutation.
- Let $G' = \pi(G)$.
- Simulate an interaction with $\widehat{\mathsf{V}}$ to commit on the adjacency matrix of G'.
- $\widehat{\mathsf{V}}$ sends a bit b' to the prover. If $b' \neq b$, abort and output $\perp$. Otherwise, if $b = 0$, the prover open the commitment to G', and if $b = 1$, the prover opens the commitment to the edges of the Hamiltonian cycle $(\pi(v_1), \ldots, \pi(v_k))$.
- Output the transcript and the randomness of $\widehat{\mathsf{V}}$ in the interaction above.

Claim 6.11. *For any* $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/q(\lambda)$,

$$\left|\Pr\left[\mathsf{Dist}(H_i^1) = 1\right] - \Pr\left[\mathsf{Dist}(H_i^2) = 1\right]\right| \leq 1/(n^\epsilon d(n)) \cdot k^2 \cdot d(n) \leq 1/(40\lambda^2 q(\lambda)).$$

Proof. Fix $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/q(\lambda)$. First, observe that when $b = 0$, the distribution of H_i^1 and H_i^2 are exactly the same. Thus, it is enough to show that

$$\left|\Pr\left[\mathsf{Dist}(H_i^1) = 1 \mid b = 1\right] - \Pr\left[\mathsf{Dist}(H_i^2) = 1 \mid b = 1\right]\right| \leq 1/(n^\epsilon d(n)) \cdot k^2 \cdot d(n).$$

Morerover, by data processing we may assume that instead aborting, when $b \neq b'$ Sim simply ignore b' and behave as in the case that $b' = 1$ (note that b' is part of the output of Sim).

This means that for some graph G, Dist can distinguish between an interaction in which Sim commit to $\pi(G)$, and an interaction in which Sim commit to $\pi(G'')$, given the transcript (that contains transcript of other rounds and the commitments and a partial opening in the i-th round) and the verifier randomness.

Fix G and its Hamiltonian cycle $w = (v_1, \ldots, v_k)$. Let $Z_{<i}$ be the random variable taking the value of the verifier view in the first $i-1$ iterations of H_i^1. For a graph $\widehat{G}$, Let $Z_i^{\widehat{G}}$ be the distribution of the i-th iteration of H_i^1, in which the simulator simulate the prover by committing the graph $\Pi(\widehat{G})$ and using the samples $D_1, \ldots, D_{k^2}$. Let $Z_{>i}^{\widehat{G}}$ be a random continuation of the last $4\lambda^2 - i$ iterations of H_i^1 from this iteration (using the algorithm Sim_i). Then we get that, $Z_{<i}, Z_i^{G''}, Z_{>i}^{G''}$ distributed exactly according to the distribution of H_i^1, and $Z_{<i}, Z_i^G, Z_{>i}^G$ distributed exactly according to the distribution of H_i^2.

Assume that Dist can distinguish between H_i^1 and H_i^2 with advantage $1/(n^\epsilon d(n)) \cdot k^2 \cdot d(n)$. By construction, we have that Dist can distinguish between

$$Z_{<i}, Z_i^{G''}, Z_{>i}^{G''}$$

and

$$Z_{<i}, Z_i^{G}, Z_{>i}^{G}.$$

By a standard hybrid argument, there are two matrices G^0, G^1 which are different in exactly one index (i,j) (where $G^0_{i,j} = 0$ and $G^1_{i,j} = 1$), such that Dist can distinguish between

$$Z_{<i}, Z_i^{G^0}, Z_{>i}^{G^0}$$

and

$$Z_{<i}, Z_i^{G^1}, Z_{>i}^{G^1}$$

with advantage $1/(n^\epsilon d(n)) \cdot d(n)$.

We want to construct an algorithm $\widehat{\mathsf{Rec}}$ with a short advice, that simulated the distribution of $Z_{<i}, Z_i^{G^0}, Z_{>i}^{G^0}$ or $Z_{<i}, Z_i^{G^1}, Z_{>i}^{G^1}$ given a single commitment, and uses Dist to break the commitment scheme. For this, we give $\widehat{\mathsf{Rec}}$ the following advice: The description of $\widehat{\mathsf{V}}$ and of Dist, The graph G and the witness w, $4\lambda^2 k^2$ random samples from $\mathcal{D}_n$ (that are chosen to maximize the success probability) and the graph G^0 together with the index (i,j).

$\widehat{\mathsf{Rec}}$ acts as follows: It first uses the first $(i-1)k^2$ samples from $\mathcal{D}$ in its advice, and the witness w, to simulate the first $(i-1)$ iterations of $\mathsf{Sim}_i(1^\lambda, G, w)$. This interaction is distributed exactly like the first $Z_{<i}$.

Next, $\widehat{\mathsf{Rec}}$ simulates $Z_i^{G^0}$ or $Z_i^{G^1}$. It simulates the interaction of $\widehat{\mathsf{V}}$ with P, where it samples a random permutation π, and when committing on the graph, on any location other than $(\pi(i), \pi(j))$, $\widehat{\mathsf{Rec}}$ uses the fixed sample from $\mathcal{D}$ taken from its advice to simulate the commitment from the honest prover. To simulate the commitment on location $(\pi(i), \pi(j))$, $\widehat{\mathsf{Rec}}$ interact with $\mathsf{Com}^{\mathcal{D}}(c)$ instead. Then, given the simulated prover view, $\widehat{\mathsf{Rec}}$ simulates the opening of every location except location (i,j). The resulting interaction distributed exactly according to $Z_i^{G^c}$.

Next, $\widehat{\mathsf{Rec}}$ simulates $Z_{>i}^{G^c}$, by simply continue the execution as described in the simulator, using fresh samples from its advice. Lastly, $\widehat{\mathsf{Rec}}$ executes Dist on the resulting simulated view, and outputs its output.

It is not hard to see that when the samples in the advice are randomly chosen, and when the committer uses $c \in \{0,1\}$, the above interaction is exactly distributed as $Z_{<i}, Z_i^{G^c}, Z_{>i}^{G^c}$. Therefore, distinguishing between $Z_{<i}, Z_i^{G^0}, Z_{>i}^{G^0}$ and $Z_{<i}, Z_i^{G^1}, Z_{>i}^{G^1}$ implies that $\widehat{\mathsf{Rec}}$ breaks the commitment.

Moreover, $\widehat{\mathsf{Rec}}$ can be implemented in polynomial time given advice that contains the description of $\widehat{\mathsf{V}}, \mathsf{Dist}, G, w, , G_t, (i,j)$ and $4\lambda^2 k^2$ random samples from $\mathcal{D}_n$. Thus, $\widehat{\mathsf{Rec}} \in \mathsf{BPP}_{\mathsf{U}}/a(n)$ if $\widehat{\mathsf{V}}, \mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/q(\lambda)$, for $a(n) \geq n^\epsilon d(n) \geq O(q(\lambda) + k^2\lambda^2(d(n)+1))$ □

H_i^3. Next, we change the order of events, letting Sim to choose the value of b and abort in the end of the protocol. In particular, the ith iteration in H_i^3 is the output distribution of the following process:

- Let $\pi \colon [k] \to [k]$ be a random permutation.

- Let $G' = \pi(G)$.
- Simulate an interaction with $\widehat{\mathsf{V}}$ to commit on the adjacency matrix of G'.
- $\widehat{\mathsf{V}}$ sends a bit b' to the prover. If $b' = 0$, the provers open the commitment to G', and if $b' = 1$, the prover opens the commitment to the edges of the Hamiltonian cycle $(\pi(v_1), \ldots, \pi(v_k))$.
- Choose a random bit $b \leftarrow \{0,1\}$. If $b \neq b'$, abort and output $\perp$. Otherwise, output the transcript and the randomness of $\widehat{\mathsf{V}}$ in the interaction above.

Claim 6.12. *$H_i^2 \equiv H_i^3$.*

Proof. The bit b is independent of all other random variables. □

H_i^4**.** Next, we change the order of events again, letting Sim to choose the value of b and abort in the beginning of the protocol. In particular, let H_4 be the output distribution of the following process:

- Let $b \leftarrow \{0,1\}$ be a random bit. If $b = 1$, abort and output $\perp$. Otherwise:
- Let $\pi \colon [k] \to [k]$ be a random permutation.
- Let $G' = \pi(G)$.
- Simulate an interaction with $\widehat{\mathsf{V}}$ to commit on the adjacency matrix of G'.
- $\widehat{\mathsf{V}}$ sends a bit b' to the prover. If $b' = 0$, the provers open the commitment to G', and if $b' = 1$, the prover opens the commitment to the edges of the Hamiltonian cycle $(\pi(v_1), \ldots, \pi(v_k))$.
- Output the transcript and the randomness of $\widehat{\mathsf{V}}$ in the interaction above.

Claim 6.13. *$H_3 \equiv H_4$ and $H_4 \equiv \mathsf{Sim}_{i+1}(1^\lambda, G, w)$.*

Proof. The first part follows since in H_i^3, Sim aborts with probability $1/2$ independently of all others random variables. The second part follows by definition of H_i^4 and $\mathsf{Sim}_{i+1}(1^\lambda, G, w)$. □

6.1.4 Proving Theorem 6.3

We are now ready to prove Theorem 6.3.

Proof of Theorem 6.3. The completeness and soundness follows by Claims 6.5 and 6.6. For zero-knowledge, assume there is a distinguisher Dist that distinguishes between the real interaction and the simulator with advantage $1/q(\lambda)$. By Claim 6.9, Dist can distinguish between $\mathsf{Sim}(1^\lambda, G)$ and $\widehat{\mathsf{Sim}}(1^\lambda, G, w)$ with advantage $1/2q(\lambda)$. Let $\mathsf{Sim}_0, \ldots, \mathsf{Sim}_{4\lambda^2}$ be the algorithms as described above. By a simple hybrid argument, there is an iteration i, such that Dist can distinguish between $\mathsf{Sim}_{i-1}(1^\lambda, G, w)$ and $\mathsf{Sim}_i(1^\lambda, G, w)$ with advantage $1/(8\lambda^2 q(\lambda))$.

However, by a simple hybrid argument using Claims 6.10 to 6.13, we have that $\mathsf{Sim}_{i-1}(1^\lambda, G, w)$ and $\mathsf{Sim}_i(1^\lambda, G, w)$ are $1/40\lambda^2 q(\lambda))$ indistinguishable for algorithms in $\mathsf{BPP}_\mathsf{U}/q(\lambda)$, , which implies that $\mathsf{Dist} \notin \mathsf{BPP}_\mathsf{U}/q(\lambda)$. □

6.2 Constructing Distribution-Aided Commitments

In this part we construct distribution-aided commitments. We present two constructions. One from strong conditional PRG, and the second from a hard language in DisjNP.

6.2.1 Distribution-Aided from Conditional PRGs

We start with the construction of commitments scheme from conditional PRGs. For this, we simply use Naor's construction of commitments from PRGs [Nao91]. We get the following theorem.

Theorem 6.14. *Assume the existence of an $\epsilon/2$-strong conditional PRG $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{2d(n)+\omega(\log n)}$ against $\mathsf{BPP}_{\mathsf{U}}/a(n)$. Then there exists an ϵ-distribution aided commitment scheme with perfect completeness against $\mathsf{BPP}_{\mathsf{U}}/a'(n)$ where $a'(n) = a(n) - O(1)$, with respect to a distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{d(n)}$.*

Proof. To prove Theorem 6.14 we consider the following protocol. Let $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ be an ϵ-strong conditional PRG, with respect to a distribution $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$, for $m(n) = 2d(n) + \omega(\log n)$.

- **Commit Stage**:
  - Rec chooses at random $r \leftarrow \{0,1\}^{3d(n)}$ and sends r to Com.
  - Com samples $w \leftarrow \mathcal{P}_n$. To commit on 0, Com sends $y = G(w)$. To commit on 1, Com sends $y = G(w) \oplus r$.
  - The commitment is set to be $z = (y, r)$.
- **Reveal Stage**:
  - To open a commitment $z = (y, r)$ for $b \in \{0,1\}$ where $y \in \{G(w), G(w) \oplus r\}$, Com sends b, w to Rec.
  - If $b = 0$, Rec verifies that $y = G(w)$. If $b = 1$, Rec verifies that $y = G(w) \oplus r$. If the verification passes, Rec outputs b. Otherwise, Rec outputs $\perp$.

The efficiency and completeness of the protocol follow by the construction and the efficiency of G. For binding, we claim that with probability at least $1 - \mathrm{neg}(n)$ over the choice of r, it holds that the sets $\mathcal{S}_0^r = \mathcal{S}_0 = \left\{G(w)\colon w \in \{0,1\}^{d(n)}\right\}$ and $\mathcal{S}_1^r = \left\{G(w) \oplus r\colon w \in \{0,1\}^{d(n)}\right\}$ are disjoint. The soundness follows as to open a (y, r) to a bit b, $\widehat{\mathsf{Com}}$ needs to show a witness that $y \in \mathcal{S}_b^r$, which means that no y can be opened to both $b = 0$ and $b = 1$.

To see that the sets are indeed disjoint with high probability, fix $w \in \{0,1\}^{d(n)}$. Then

$$\Pr_{r\leftarrow\{0,1\}^{m(n)}}[(G(w) \oplus r) \in \mathcal{S}_0] \leq |\mathcal{S}_0|/2^{m(n)} \leq 2^{d(n)}/2^{m(n)} \leq 2^{-d(n)+\omega(\log n)} = 2^{-d(n)} \cdot \mathrm{neg}(n).$$

The claim now follows by taking union bound on all possible $w \in \{0,1\}^{d(n)}$.

Finally we prove that the scheme is hiding. Fix $n \in \mathbb{N}$, $\widehat{\mathsf{Rec}} \in \mathsf{BPP}_{\mathsf{U}}/a(n)$, and assume that $\widehat{\mathsf{Rec}}$ can distinguish between Com having input $b = 0$ to Com having input $b = 1$ with advantage ϵ.

We construct two algorithms $\mathsf{Dist}_0, \mathsf{Dist}_1$ that at least one of them breaks the security of G. Let Dist_0 be the algorithm that given input $y = v$ (which is either sampled from $G(\mathcal{P}_n)$ or from $U_{m(n)}$), first computes the first message r of $\widehat{\mathsf{Rec}}$. Then gives back $\widehat{\mathsf{Rec}}$ the message v and outputs $\widehat{\mathsf{Rec}}$'s output. Similarly, let Dist_1 be the algorithm that given input v first computes the first message r of $\widehat{\mathsf{Rec}}$. Then gives back $\widehat{\mathsf{Rec}}$ the message $y = v \oplus r$ and outputs $\widehat{\mathsf{Rec}}$'s output.

We first observe that $\Pr[\mathsf{Dist}_0(G(\mathcal{P}_n)) = 1]$ is exactly the probability that $\widehat{\mathsf{Rec}}$ outputs 1 in an interaction with $\mathsf{Com}(0)$, while $\Pr[\mathsf{Dist}_1(G(\mathcal{P}_n)) = 1]$ is exactly the probability that $\widehat{\mathsf{Rec}}$ outputs 1 in an interaction with $\mathsf{Com}(1)$. Moreover, it holds that $\Pr\big[\mathsf{Dist}_0(G(U_{m(n)})) = 1\big] = \Pr\big[\mathsf{Dist}_1(U_{m(n)}) = 1\big]$ as $U_{m(n)} \equiv U_{m(n)} \oplus r$ for any r. We thus conclude that

$$\begin{aligned}
\epsilon &\leq |\Pr[\mathsf{Dist}_0(G(\mathcal{P}_n)) = 1] - \Pr[\mathsf{Dist}_1(G(\mathcal{P}_n)) = 1]| \\
&= \big|\Pr[\mathsf{Dist}_0(G(\mathcal{P}_n)) = 1] - \Pr\big[\mathsf{Dist}_0(U_{m(n)}) = 1\big] + \Pr\big[\mathsf{Dist}_1(U_{m(n)}) = 1\big] - \Pr[\mathsf{Dist}_1(G(\mathcal{P}_n)) = 1]\big| \\
&\leq \big|\Pr[\mathsf{Dist}_0(G(\mathcal{P}_n)) = 1] - \Pr\big[\mathsf{Dist}_0(U_{m(n)}) = 1\big]\big| + \big|\Pr\big[\mathsf{Dist}_1(U_{m(n)}) = 1\big] - \Pr[\mathsf{Dist}_1(G(\mathcal{P}_n)) = 1]\big|
\end{aligned}$$

Which implies that either $\mathcal{D}_0$ or $\mathcal{D}_1$ break the security of the PRG. □

6.2.2 Distribution-Aided Commitments from Hardness of DisjNP

We prove the following theorem.

Theorem 6.15. *Assume that there is a promise problem $\Pi \in \mathsf{DisjNP}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ such that*

$$\Pr_{(x,w)\leftarrow P_n}[\mathsf{Dist}(x) = \Pi(x)] \leq (1+\epsilon)/2$$

for any $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/a(n)$. Then there exists an an ϵ-distribution-aided commitment scheme with perfect completeness against $\mathsf{BPP}_{\mathsf{U}}/a'(n)$ for $a'(n) = a(n) - O(1)$, with respect to a distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{d(n)}$ where $d \in \mathrm{poly}$, and $d(n) - n$ is the bound on the witness length for instances of length n.

Proof. In the following, for $w \in \mathcal{Y} \cup \mathcal{N}$, let $\Pi(x) \in \{0,1\}$ be 1 if $x \in \mathcal{Y}$ and 0 otherwise. Consider the following protocol:

- **Commit Stage**:
  - Com samples $(x, w) \leftarrow \mathcal{P}_n$. To commit to a bit b, Com sends $z = (x, b \oplus \Pi(x))$. The commitment is set to be z.
- **Reveal Stage**:
  - To open a commitment $z = (x, c)$, the prover sends b and a witness w for x.
  - Rec verifies that $c = b \oplus \Pi(x)$. If the verification passes, Rec outputs b. Otherwise, Rec outputs $\perp$.

The efficiency and completeness follows by construction. The binding holds since $\mathcal{Y}$ and $\mathcal{N}$ are disjoint. The hiding holds since an malicious receiver $\widehat{\mathsf{Rec}}$ that distinguishes between commitment to 1 to commitment of 0 can be used to decide Π. Indeed, assume without loss of generality that $\widehat{\mathsf{Rec}}$ outputs 1 with larger probability on commitment to 1 than on commitment to 0. Let Dist be

the algorithm that given $x \in \mathcal{Y} \cup \mathcal{N}$, chooses a random bit $c \leftarrow \{0,1\}$, and executes $\widehat{\mathsf{Rec}}(x,c)$. Then Dist outputs c if $\widehat{\mathsf{Rec}}$ outputs 1, or $1-c$ otherwise. Then,

$$\begin{aligned}
\Pr_{(x,w)\leftarrow\mathcal{P}_n}[\mathsf{Dist}(x)=\Pi(x)] &= \Pr_{(x,w)\leftarrow\mathcal{P}_n, c\leftarrow\{0,1\}}\left[\widehat{\mathsf{Rec}}(x,c)=1_{c=\Pi(x)}\right] \\
&= 1/2\cdot\Pr_{(x,w)\leftarrow\mathcal{P}_n}\left[\widehat{\mathsf{Rec}}(x,\Pi(x))=1\right] + 1/2\cdot\Pr_{(x,w)\leftarrow\mathcal{P}_n}\left[\widehat{\mathsf{Rec}}(x,1\oplus\Pi(x))=0\right] \\
&= 1/2 + 1/2(\Pr_{(x,w)\leftarrow\mathcal{P}_n}\left[\widehat{\mathsf{Rec}}(x,\Pi(x))=1\right] - \Pr_{(x,w)\leftarrow\mathcal{P}_n}\left[\widehat{\mathsf{Rec}}(x,1\oplus\Pi(x))=1\right]) \\
&= 1/2 + 1/2(\Pr_{z\leftarrow\mathsf{Com}^{\mathcal{D}}(0)}\left[\widehat{\mathsf{Rec}}(z)=1\right] - \Pr_{z\leftarrow\mathsf{Com}^{\mathcal{D}}(1)}\left[\widehat{\mathsf{Rec}}(z)=1\right]) \\
&\geq 1/2+\epsilon/2
\end{aligned}$$

with contradiction to the assumption on $\mathcal{P}$. □

7 From Distribution-Aided Zero-Knowledge to ϵ-Zero-Knowledge

In this section we use Theorem 6.3 to construct ϵ-Zero-Knowledge proof. We first define ϵ-Zero-Knowledge.

Definition 7.1 ((a,ϵ) Zero-Knowledge with non-uniform simulator)**.** *A two party protocol* (P,V) *is an* (a,ϵ)-zero-knowledge proof with non-uniform simulator *for an* NP *relation* $\mathcal{R}$ *if* V *is efficient and:*

- α**-Completeness**. *For every* $(x,w)\in\mathcal{R}$ *and security parameter* $\lambda\in\mathbb{N}$, V *accepts with probability at least* $1-\alpha(\lambda)$ *on the interaction* $(\mathsf{P}(w),\mathsf{V})(1^\lambda,x)$.
- β**-Soundness**. *For every* $x\notin\mathcal{L}(\mathcal{R})$ *and every malicious prover* $\widehat{\mathsf{P}}$, V *rejects with probability at least* $\beta(\lambda)$ *on the interaction* $(\widehat{\mathsf{P}},\mathsf{V})(1^\lambda,x)$.
- (a,ϵ)**-Zero-Knowledge**. *For every algorithm* $\widehat{\mathsf{V}}\in$ BPP*, there exists an efficient,* non-uniform *simulator* $\mathsf{Sim}_{\mathsf{V}}$*, such that for every ensemble* $\{(x_\lambda,w_\lambda,z_\lambda)\in\mathcal{R}\}_{\lambda\in\mathbb{N}}$ *with* $z_\lambda\in\{0,1\}^{a(\lambda)}$, $\left\{\mathsf{Sim}_{\mathsf{V}}^{\mathcal{D}}(z,1^\lambda,x_\lambda)\right\}_\lambda$ *is* $\epsilon(\lambda)$*-indistinguishable against any distinguisher* $\mathsf{Dist}\in\mathrm{BPP}/a(\lambda)$ *from* $\left\{\mathrm{View}_{\widehat{\mathsf{V}}_z}((\mathsf{P}^{\mathcal{D}}(w_\lambda),\widehat{\mathsf{V}}_z)(1^\lambda,x_\lambda))\right\}_\lambda$.

When $a(n)=0$ *we simply say that* (P,V) *is* ϵ*-zero-knowledge with non-uniform simulator.*

We prove the following theorem.

Theorem 7.2. *Let* $\epsilon\in 1/\mathrm{poly}(\lambda)$ *be a function. Assume there exists a distribution-aided* α*-correct,* β*-sound* (q,ϵ)*-zero-knowledge proof for* NP *against distinguishers in* $\mathsf{BPP}_{\mathsf{U}}/\log n$*. Then there exists an (inefficient prover)* α*-correct,* β*-sound* $(q,2\epsilon)$*-zero-knowledge proof for* NP *with non-uniform simulator.*

Moreover, there exists an efficient, non uniform prover protocol which is $(\alpha+\epsilon)$*-correct,* β*-sound* $(q-\log n,2\epsilon)$*-zero-knowledge proof for* NP *with non-uniform simulator.*

Proof. Let (P,V) be a distribution-aided α-correct, β-sound (q,ϵ)-zero-knowledge proof for NP against distinguishers in $\mathsf{BPP}_{\mathsf{U}}/\log n$. We claim that the same protocol is $(q,2\epsilon)$-zero-knowledge, when P (inefficiently) samples from the distribution $\mathcal{D}$.

We need to construct the simulator. Fix a malicious verifier

$$\widehat{V} \in \mathrm{BPP} \subseteq \mathsf{BPP}_{\mathsf{U}}/\log n,$$

and let $\mathsf{Sim}^{\mathcal{D}}_{\widehat{V}}$ be the simulation of (P,V). Assume $\mathsf{Sim}^{\mathcal{D}}_{\widehat{V}}(1^\lambda, x)$ uses at most $t(\lambda,|x|)$ samples from $\mathcal{D}$. We will construct a new simulator Sim' that has $r(\lambda) = 10q(\lambda)/\epsilon^2(\lambda)$ sequences of $t(\lambda,|x|)$ samples each hard-coded to its advice. On input $(1^\lambda, x)$ Sim' will simply simulate Sim, where in the begining of the execution Sim' will sample a random sequence from its advice, and whenever Sim samples the i-th sample from $\mathcal{D}$, Sim' will feed it the i-th sample from the sequence.

Let $\mathcal{S}$ be the distribution of $t(\lambda,|x|)$ samples from $\mathcal{D}$. For a sequence $S = (D_1, \ldots, D_{t(\lambda,|x|)})$, let $\mathsf{Sim}^{S}_{\widehat{V}}(1^\lambda, x)$ be the execution of Sim where the i-th sample from $\mathcal{D}$ is hard-coded to be D_i.

Then for every distinguisher Dist,

$$\mathrm{E}_{S\leftarrow(\mathcal{D}_n)^{t(\lambda,|x|)}}\Big[\mathsf{Dist}(\mathsf{Sim}^{S}_{\widehat{V}}(1^\lambda,x))=1\Big] = \Pr\Big[\mathsf{Dist}(\mathsf{Sim}^{\mathcal{D}_n}_{\widehat{V}}(1^\lambda,x))=1\Big].$$

Using Chernoff, we get that for every distinguisher Dist,

$$\Pr_{S_1,\ldots,S_r\leftarrow(\mathcal{D}_n)^{t(\lambda,|x|)}}\left[\left|\Pr_{i\leftarrow[r]}\Big[\mathsf{Dist}(\mathsf{Sim}^{S_i}_{\widehat{V}}(1^\lambda,x))=1\Big] - \Pr\Big[\mathsf{Dist}(\mathsf{Sim}^{\mathcal{D}_n}_{\widehat{V}}(1^\lambda,x))=1\Big]\right| \geq \epsilon(\lambda)\right] < 2^{-q(\lambda))}.$$

By taking union bound over all $2^{q(\lambda)}$ algorithms in $\mathsf{BPP}_{\mathsf{U}}/q(\lambda)$, we get that there is a fixing of $S_1,\ldots,S_r$, such that $\mathsf{Sim}'_{\widehat{V}}$ is ϵ-indistinguishable from $\mathsf{Sim}^{\mathcal{D}}_{\widehat{V}}$ against $\mathsf{BPP}_{\mathsf{U}}/q(\lambda)$. The theorem now follows using an hybrid argument.

For the moreover part, we simply give the efficient prover the same advice as for the simulator. By the same proof, no distinguisher $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/q(\lambda)$ can ϵ-distinguish between the real prover to the efficient one. □

7.1 Witness Hiding

We next define uniform witness-hiding for non-uniform hard languages, and prove that our notion of zero-knowledge captures it.

Definition 7.3 (Witness Hiding)**.** *Let $\mathcal{L} \in \mathrm{NP}$ be a language with witness-relation $\mathcal{R}$. Let $\mathcal{X} = \{\mathcal{X}_n\}_{n\in\mathbb{N}}$ be an ensemble of distributions over $\mathcal{L}_n$. An interactive proof (P,V) is* ϵ-uniform Witness Hiding *with respect to $\mathcal{X}$ if the following holds: If for every poly-size circuit C and for every large enough $n \in \mathbb{N}$ it holds that*

$$\Pr_{x\leftarrow\mathcal{X}_n}[(x, C(x)) \in \mathcal{R}] = \mathrm{neg}(n),$$

Then for every PPT $\widehat{\mathsf{V}}$*, every large enough n and every witness distribution (jointly distributed with $\mathcal{X}$) W_n,*

$$\Pr\Big[(\mathcal{X}_n, \langle \mathsf{P}(\mathcal{X}_n, W), \widehat{\mathsf{V}}(\mathcal{X}_n)\rangle) \in \mathcal{R}\Big] \leq \epsilon(n).$$

We say that (P,V) is ϵ-uniform Witness Hiding with respect to non-uniform hard distributions *if the above holds for any $\mathcal{X}$.*

We prove the following lemma.

Lemma 7.4. *Let $\epsilon \in 1/\mathrm{poly}$, and let (P,V) be an ϵ-zero-knowledge proof for $\mathcal{L}$ with non-uniform simulator. Then (P,V) is 2ϵ-uniform Witness Hiding with respect to non-uniform hard distributions.*

We remark that when the proof (P,V) is ZK given bounded auxiliary-input, so does the witness-hiding.

Proof. Assume that for some distribution $\mathcal{X}$, there is PPT $\widehat{\mathsf{V}}$ such that

$$\Pr\Big[(\mathcal{X}_n, \langle \mathsf{P}(\mathcal{X}_n, W), \widehat{\mathsf{V}}(\mathcal{X}_n)\rangle) \in \mathcal{R}\Big] > 2\epsilon(n).$$

We will show that

$$\Pr\big[(\mathcal{X}_n, \mathsf{Sim}_{\widehat{V}}) \in \mathcal{R}\big] > \epsilon(n), \tag{6}$$

which means that $\mathcal{X}$ is not hard for poly-size circuits. Assume that Equation (6) does not hold. Then,

$$\mathrm{E}_{x\leftarrow\mathcal{X}_n}\Big[\Pr\Big[(x, \langle \mathsf{P}(x, W), \widehat{\mathsf{V}}(x)\rangle) \in \mathcal{R}\Big] - \Pr\big[(x, \mathsf{Sim}_{\widehat{V}}) \in \mathcal{R}\big]\Big] > \epsilon.$$

Fix $x \in \mathcal{L}$ that maximize the above expectation. Then we have that

$$\Pr\Big[(x, \langle \mathsf{P}(x, W), \widehat{\mathsf{V}}(x)\rangle) \in \mathcal{R}\Big] - \Pr\big[(x, \mathsf{Sim}_{\widehat{V}}) \in \mathcal{R}\big] > \epsilon,$$

with contradiction to the zero-knowledge property. □

8 Putting it All Together

Theorem 8.1. *Assume that for some $t \in \mathrm{poly}$ with $t(n) \geq n$ and some constant $1/3 > \epsilon > 0$, $(\mathrm{K}^t \text{ v.s. } \mathsf{rK}^{p(t)})[n^\epsilon, n - \log n] \notin \mathsf{ioP}/\mathrm{poly}$, for any $p \in \mathrm{poly}$. Then for any $q \in \mathrm{poly}$, there exists a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof for NP with non-uniform simulator, with perfect completeness and soundness error $2^{-\lambda}$.*

Proof. By the assumption and Theorem 4.1, there exists an $1/3n^\epsilon$-weak conditional PRG $G\colon \{0,1\}^{n^\epsilon + O(\log^4 n)} \to \{0,1\}^{5n^\epsilon}$ secure against $\mathrm{BPP}/(n/2)$.

By Corollary 3.4, this implies a $1/n^\epsilon$-strong conditional PRG against $\mathsf{BPP}_{\mathsf{U}}/(n^{1-2\epsilon}/18m(n))$.

By Theorem 6.14, this implies an $2/n^\epsilon$-distribution-aided commitment scheme with perfect completeness secure against $\mathsf{BPP}_{\mathsf{U}}/(n^{1-2\epsilon}/18m(n) - O(1))$ with respect to a distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{n^\epsilon + \log^4 n}$.

Finally, by Theorem 6.3, such a commitment implies distribution-aided $1/2q(\lambda)$-zero-knowledge proof against $\mathsf{BPP}_{\mathsf{U}}/2q(\lambda)$, and by Theorem 7.2 this implies a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof. □

Theorem 8.2. *Assume that $\mathsf{DisjNP} \not\subseteq \mathrm{io}\,\mathsf{BPP}_{\mathsf{U}}/\mathrm{poly}$. Then for any $q \in \mathrm{poly}$, there exists there exists a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof for NP with non-uniform simulator, with perfect completeness and soundness error $2^{-\lambda}$.*

Proof. Let $\delta = \epsilon/4$, and assume $\mathsf{DisjNP} \not\subseteq \mathrm{io}\,\mathsf{BPP}_{\mathsf{U}}/\mathrm{poly}$. By the assumption and Lemma 5.5, there exists a distribution ensemble $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ such that

$$\Pr_{(x,w)\leftarrow\mathcal{P}_n}[\mathsf{Dist}(x) = \Pi(x)] \leq (1+\delta)/2$$

for any $\mathsf{Dist} \in \mathsf{BPP}_{\mathsf{U}}/\mathrm{poly}$.

By Theorem 6.15, this implies an distribution-aided commitment scheme with perfect completeness secure against $\mathsf{BPP}_{\mathsf{U}}/\mathrm{poly}$.

Finally, by Theorem 6.3, such a commitment implies distribution-aided $1/2q(\lambda)$-zero-knowledge proof against $\mathsf{BPP}_{\mathsf{U}}/2q(\lambda)$, and by Theorem 7.2 this implies a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof. □

9 Weak Zero-Knowledge from Hardness of $\mathrm{Gap_pMrK^tP}$

In this section we sketch the construction of the weak ZK proof starting from the worst-case hardness of $\mathrm{Gap_pMrK^tP}$. Recall that for $p, t \in \mathrm{poly}$, $\mathrm{Gap_pMrK^tP}[s, \ell]$ is the following promise problem:

- $\mathcal{Y} = \left\{ x \in \{0,1\}^n \colon \mathsf{rK}^{t(n)}(x) \leq s(n) \right\}$
- $\mathcal{N} = \left\{ x \in \{0,1\}^n \colon \mathsf{rK}^{p(t(n))}(x) > \ell(n) \right\}$

We next explain how to change our results and proof to derive the construction from $\mathrm{Gap_pMrK^tP}$. We prove the following theorem.

Theorem 9.1. *Assume that for some $t \in \mathrm{poly}$ with $t(n) \geq n$ and some constant $1/3 > \epsilon > 0$, $\mathrm{Gap_pMrK^tP}[n^\epsilon, n - \log n] \notin \mathsf{ioP}/\mathrm{poly}$, for any $p \in \mathrm{poly}$. Then for any $q \in \mathrm{poly}$, there exists a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof with non-uniform simulator, with $1 - \mathrm{neg}$-completeness and soundness error $2^{-\lambda}$.*

9.1 Randomized Conditional PRG

We start with the definition of a randomized version of conditional PRG.

Definition 9.2 (Conditional PRG)**.** *Let $\epsilon \colon \mathbb{N} \to \mathbb{N}$ be a function. An efficiently computable randomized function $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ is an ϵ-*randomized conditional PRG against distinguishers class *$\mathcal{C}$ if for every $n \in \mathbb{N}$ $m(n) > d(n)$, and for every distinguisher $\mathsf{Dist} \in \mathcal{C}$ there exist a distribution ensemble $\mathcal{P} = \{\mathcal{P}_n\}_{n \in \mathbb{N}}$ over $\{0,1\}^{d(n)}$, such that*

$$\left| \Pr[\mathsf{Dist}(G(\mathcal{P}_n)) = 1] - \Pr\left[\mathsf{Dist}(\mathbf{U}_{m(n)}) = 1\right] \right| \leq \epsilon(n)$$

for every large enough $n \in \mathbb{N}$. Moreover, for every $x \in \mathrm{Supp}(\mathcal{P}_n)$ there exists $y \in \{0,1\}^{m(n)}$ such that $\Pr[G(x) = y] \geq 2^{-n}$.

*If there exists a fixed distribution ensemble $\{\mathcal{P}_n\}_{n \in \mathbb{N}}$ such that the above holds for every distinguisher $\mathsf{Dist} \in \mathcal{C}$, we say that G is an ϵ-*strong conditional PRG *against $\mathcal{C}$.*

Note that the choice of 2^{-n} in the above definition is arbitrary, and whenever G outputs y with probability larger than say, $2/3$, we can amplify it using repetitions. We want the indistinguishability to hold when G outputs the correct value y.

By essentially the same proof of Corollary 3.4, we derive the following weak-to-strong amplification for randomized PRG.

Corollary 9.3. *Assume that G is an $\epsilon(n)$-randomized conditional PRG against $\mathsf{BPP}_{\mathsf{U}}/a(n)$ for a constant $c \in \mathbb{N}$. Then G is an $3\epsilon(n)$-strong randomized conditional PRG against $\mathsf{BPP}_{\mathsf{U}}/a'(n)$, for $a'(n) = \Omega(a(n)/(n^{2c} m(n)))$.*

And by the proof of Corollary 4.2, we get the following corollary.

Corollary 9.4. *Assume that for some efficiently computable functions $\ell, s\colon \mathbb{N} \to \mathbb{N}$ and $t \in \text{poly}$ with $t(n) > n$,*

$$\text{Gap}_{\text{p}}\text{MrK}^{\text{t}}\text{P}\, tp(t)[s(n), \ell(n)] \notin \mathsf{ioP}/\text{poly}$$

for any polynomial p. Then for every efficiently computable function $m(n) \in \text{poly}$ there exists a function $G\colon \{0,1\}^{d(n)=s(n)+\log^2 m(n)} \to \{0,1\}^{3d(n)}$ which is an ϵ-conditional PRG secure against $\text{BPP}/a(n)$ for $a(n) = \Omega((\ell(n)/d(n))^{1/3})$ and $\epsilon = 1/a(n)$.

Finally, by the same construction of conditional PRG from Section 4 (with amplification to make G output the same answer with high probability), we get the following theorem.

Theorem 9.5. *Assume that for some efficiently computable functions $\ell, s, : \mathbb{N} \to \mathbb{N}$ and for $t \in \text{poly}$ with $t(n) \geq n$, $\text{Gap}_{\text{p}}\text{MrK}^{\text{t}}\text{P}[s(n), \ell(n)] \notin \mathsf{ioP}/\text{poly}$, for any polynomials p. Then for every efficiently computable function $m(n) \in \text{poly}$ there exists a function $G\colon \{0,1\}^{s(n)+\log^4 n} \to \{0,1\}^{m(n)}$ which is an ϵ-randomized conditional PRG secure against $\mathsf{BPP}_{\mathsf{U}}/a(n)$ for $a(n) = \ell(n) - O(m^2(n) + \log^4 n)$ and for any $\epsilon \in 1/\text{poly}$.*

9.2 Distribution-Aided Commitments

We use the same construction of commitments from PRGs as in the proof of Theorem 6.14, but we make the verifier check that the PRG outputs the same output with high probability. We get the following theorem.

Theorem 9.6. *Assume the existence of an $\epsilon/2$-randomized strong conditional PRG $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{2d(n)+\omega(\log n)}$ against $\mathsf{BPP}_{\mathsf{U}}/a(n)$. Then there exists an ϵ-distribution aided commitment scheme against $\mathsf{BPP}_{\mathsf{U}}/a'(n)$ where $a'(n) = a(n) - O(1)$, with respect to a distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{d(n)}$.*

Proof. To prove Theorem 9.6 we consider the following protocol. Let $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ be an ϵ-randomized strong conditional PRG, with respect to a distribution $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$, for $m(n) = 2d(n) + \omega(\log n)$.

- **Commit Stage**:
  - Rec chooses at random $r \leftarrow \{0,1\}^{3d(n)}$.
  - Com samples $w \leftarrow \mathcal{P}_n$. To commit on 0, Com sends $y = G(w)$. To commit on 1, Com sends $y = G(w) \oplus r$.
  - The commitment is set to be $z = (y, r)$.
- **Reveal Stage**:
  - To open a commitment $z = (y, r)$ for $b \in \{0,1\}$ where $y \in \{G(w), G(w) \oplus r\}$, Com sends b, w to Rec.
  - Rec computes $G(w)$ n times with fresh randomness. If not all output are the same, Rec rejects.
  - If $b = 0$, Rec verifies that $y = G(w)$. If $b = 1$, Rec verifies that $y = G(w) \oplus r$. If the verification passes, Rec outputs b. Otherwise, Rec outputs $\perp$.

For completeness, we notice that for every $w \in \mathrm{Supp}(\mathcal{D})$, G outputs the same output with probability 2^{-n}. Thus, the probability that Rec reject when Com is honest is negligible.

For binding, for every $w \in \{0,1\}^{d(n)}$, let y_w the image y that maximize $\Pr[G(w) = y]$ (breaking ties arbitrary). Notice that for every $y \neq y_w$, $\Pr[G(w) = y] \leq 1/2$. Thus, the probability that the prover opens a commitment (y, r) with w such that $y \neq y_w$, and Rec does not abort in the second step of the reveal stage is negligible.

We can therefore assume that Com always sends w such that $y_w = y$ in the reveal stage, and continue the proof as in the proof of Theorem 6.14. □

9.3 Putting It All Together

We next prover Theorem 9.1

Proof of Theorem 9.1. By the assumption and Theorem 9.5, there exists an $1/3n^\epsilon$-conditional PRG $G\colon \{0,1\}^{n^\epsilon + O(\log^4 n)} \to \{0,1\}^{5n^\epsilon}$ secure against $\mathrm{BPP}/(n/2)$.

By Corollary 9.3, this implies a $1/n^\epsilon$-strong conditional PRG against $\mathsf{BPP}_{\mathsf{U}}/(n^{1-2\epsilon}/18m(n))$.

By Theorem 9.6, this implies an $2/n^\epsilon$-distribution-aided commitment scheme with perfect completeness secure against $\mathsf{BPP}_{\mathsf{U}}/(n^{1-2\epsilon}/18m(n) - O(1))$ with respect to a distribution $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{n^\epsilon + \log^4 n}$.

Finally, by Theorem 6.3, such a commitment implies distribution-aided $1/2q(\lambda)$-zero-knowledge proof against $\mathsf{BPP}_{\mathsf{U}}/q(\lambda)$, and by Theorem 7.2 this implies a $(q(\lambda), 1/q(\lambda))$-zero-knowledge proof. □

Acknowledgment

We are grateful to anonymous reviewers for valuable comments, and for pointing us to the literature on DisjNP.

References

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *stoc29*. See also ECCC TR96-065. 1997, pp. 284–293 (cit. on p. 3).

[AH19] Eric Allender and Shuichi Hirahara. "New insights on the (non-) hardness of circuit minimization and related problems". In: *ACM Transactions on Computation Theory (ToCT)* 11.4 (2019), pp. 1–27 (cit. on p. 13).

[Bar01] Boaz Barak. "How to Go Beyond the Black-Box Simulation Barrier." In: *focs42*. 2001, pp. 106–115 (cit. on p. 5).

[BC20] Nir Bitansky and Arka Rai Choudhuri. "Characterizing deterministic-prover zero knowledge". In: *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part I 18*. Springer. 2020, pp. 535–566 (cit. on p. 5).

[BDS25] Marshall Ball and Dana Dachman-Soled. "(Inefficient Prover) ZAPs from Hard-to-Invert Functions". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2025, pp. 274–302 (cit. on p. 14).

[Bey06] Olaf Beyersdorff. "Disjoint NP-Pairs and Propositional Proof Systems". PhD thesis. Humboldt-Universität zu Berlin, 2006 (cit. on p. 4).

[BKPRV24] Nir Bitansky, Chethan Kamath, Omer Paneth, Ron D Rothblum, and Prashant Nalini Vasudevan. "Batch proofs are statistically hiding". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 435–443 (cit. on p. 15).

[BLM05] Harry Buhrman, Troy Lee, and Dieter van Melkebeek. "Language compression and pseudorandom generators". In: *computational complexity* 14.3 (2005), pp. 228–255 (cit. on pp. 6, 7).

[Blu83] Manuel Blum. "Coin flipping by telephone a protocol for solving impossible problems". In: *ACM SIGACT News* 15.1 (1983), pp. 23–27 (cit. on p. 4).

[Blu86] Manuel Blum. "How to prove a theorem so no one else can claim it". In: *International Congress of Mathematicians, 1986*. 1986 (cit. on pp. 11, 29).

[BM82] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo Random Bits". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 112–117 (cit. on pp. 4, 6).

[BOV03] Boaz Barak, Shien Jin Ong, and Salil Vadhan. "Derandomization in Cryptography." In: *crypto03*. 2003, pp. 299–315 (cit. on p. 12).

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. "Computational analogues of entropy". In: *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques (APPROX)*. 2003, pp. 200–215 (cit. on pp. 21, 22).

[Cha69] Gregory J. Chaitin. "On the Simplicity and Speed of Programs for Computing Infinite Sets of Natural Numbers". In: *J. ACM* 16.3 (1969), pp. 407–422 (cit. on p. 3).

[CT21] Lijie Chen and Roei Tell. "Simple and fast derandomization from very hard functions: eliminating randomness at almost no cost". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 283–291 (cit. on pp. 5, 12).

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* (1976), pp. 644–654 (cit. on pp. 3, 4).

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry Stockmeyer. "Magic functions: In memoriam: Bernard m. dwork 1923–1998". In: *Journal of the ACM (JACM)* 50.6 (2003), pp. 852–921 (cit. on p. 6).

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. "Concurrent zero-knowledge". In: *Journal of the ACM (JACM)* 51.6 (2004), pp. 851–898 (cit. on p. 5).

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *Annual International Cryptology Conference (CRYPTO)*. 1984, pp. 10–18 (cit. on p. 3).

[ESY84] Shimon Even, Alan L Selman, and Yacov Yacobi. "The complexity of promise problems with applications to public-key cryptography". In: *Information and control* 61.2 (1984), pp. 159–173 (cit. on pp. 4, 16).

[FS86] Amos Fiat and Adi Shamir. "How to prove yourself: Practical solutions to identification and signature problems". In: *Conference on the theory and application of cryptographic techniques*. Springer. 1986, pp. 186–194 (cit. on p. 4).

[FS90] Uriel Feige and Adi Shamir. "Witness Indistinguishable and Witness Hiding Protocols". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. Ed. by Harriet Ortiz. ACM, 1990, pp. 416–426 (cit. on p. 6).

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions". In: *Journal of the ACM* (1986), pp. 792–807 (cit. on p. 4).

[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, pp. 16–1 (cit. on p. 17).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*. 1989, pp. 25–32 (cit. on pp. 9, 24).

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: *Journal of Computer and System Sciences* (1984), pp. 270–299 (cit. on p. 4).

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* (1989). Preliminary version in *STOC'85*, pp. 186–208 (cit. on p. 5).

[GO94] Oded Goldreich and Yair Oren. "Definitions and properties of zero-knowledge proof systems". In: *Journal of Cryptology* 7.1 (1994), pp. 1–32 (cit. on p. 12).

[GS98] Venkatesan Guruswami and Madhu Sudan. "Improved decoding of Reed-Solomon and algebraic-geometric codes". In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*. IEEE. 1998, pp. 28–37 (cit. on pp. 9, 24).

[GSZ06] Christian Glaßer, Alan L Selman, and Liyu Zhang. "Survey of disjoint NP-pairs and relations to propositional proof systems". In: *Theoretical Computer Science: Essays in Memory of Shimon Even*. Springer, 2006, pp. 241–253 (cit. on p. 4).

[Har83] J. Hartmanis. "Generalized Kolmogorov complexity and the structure of feasible computations". In: *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*. 1983, pp. 439–445. DOI: 10.1109/SFCS.1983.21 (cit. on p. 3).

[HI25] Shuichi Hirahara and Rahul Ilango. "NP-hardness of the Minimum Circuit Size Problem from Well-Studied Assumptions". In: *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2025, pp. 1648–1664 (cit. on p. 13).

[HILL99] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A pseudorandom generator from any one-way function". In: *SIAM Journal on Computing* (1999), pp. 1364–1396 (cit. on pp. 4, 6).

[Hir18] Shuichi Hirahara. "Non-black-box worst-case to average-case reductions within NP". In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2018, pp. 247–258 (cit. on pp. 3, 7, 8, 18).

[Hir22] Shuichi Hirahara. "NP-hardness of learning programs and partial MCSP". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 968–979 (cit. on p. 13).

[Hir23] Shuichi Hirahara. "Capturing one-way functions via np-hardness of meta-complexity". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 1027–1038 (cit. on p. 5).

[HIR23] Yizhi Huang, Rahul Ilango, and Hanlin Ren. "NP-Hardness of Approximating Meta-Complexity: A Cryptographic Approach". In: *Cryptology ePrint Archive* (2023) (cit. on p. 13).

[HN23] Shuichi Hirahara and Mikito Nanashima. "Learning in pessiland via inductive inference". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 447–457 (cit. on pp. 4, 5, 13, 14).

[HR07] Iftach Haitner and Omer Reingold. "Statistically-hiding commitment from any one-way function". In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. 2007, pp. 1–10 (cit. on p. 4).

[HS22] Shuichi Hirahara and Rahul Santhanam. "Errorless versus error-prone average-case complexity". In: (2022) (cit. on p. 4).

[IKV23] Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. "The power of natural properties as oracles". In: *computational complexity* 32.2 (2023), p. 6 (cit. on p. 13).

[IL89] Russell Impagliazzo and Michael Luby. "One-way Functions are Essential for Complexity Based Cryptography". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1989, pp. 230–235 (cit. on p. 4).

[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)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2020 (cit. on p. 13).

[Ila23] Rahul Ilango. "SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 733–742 (cit. on p. 13).

[ILCO20] Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. "NP-hardness of circuit minimization for multi-output functions". In: *CCC'20: Proceedings of the 35th Computational Complexity Conference*. 2020, pp. 1–36 (cit. on p. 13).

[IW97] Russell Impagliazzo and Avi Wigderson. "P= BPP if E requires exponential circuits: Derandomizing the XOR lemma". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 1997, pp. 220–229 (cit. on pp. 5, 7).

[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*. 2000, pp. 73–79 (cit. on pp. 8, 13).

[KM99] Adam R. Klivans and Dieter van Melkebeek. "Graph Nonisomorphism has Subexponential Size Proofs unless the Polynomial-time Hierarchy Collapses". In: *stoc31*. 1999, pp. 659–667 (cit. on p. 5).

[Ko86] Ker-I Ko. "On the Notion of Infinite Pseudorandom Sequences". In: *Theor. Comput. Sci.* 48.3 (1986), pp. 9–33. DOI: 10.1016/0304-3975(86)90081-2. URL: https://doi.org/10.1016/0304-3975(86)90081-2 (cit. on p. 3).

[Ko91] Ker-I Ko. "On the complexity of learning minimum time-bounded Turing machines". In: *SIAM Journal on Computing* 20.5 (1991), pp. 962–986 (cit. on p. 13).

[Kol68] A. N. Kolmogorov. "Three approaches to the quantitative definition of information". In: *International Journal of Computer Mathematics* 2.1-4 (1968), pp. 157–168 (cit. on p. 3).

[LMP24] Yanyi Liu, Noam Mazor, and Rafael Pass. "A Note on Zero-Knowledge for NP and One-Way Functions". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 5, 18).

[LOS21] Zhenjian Lu, Igor C Oliveira, and Rahul Santhanam. "Pseudodeterministic algorithms and the structure of probabilistic time". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 303–316 (cit. on pp. 6, 7).

[LP20] Yanyi Liu and Rafael Pass. "On one-way functions and Kolmogorov complexity". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2020, pp. 1243–1254 (cit. on pp. 3, 6, 7).

[LP21a] Yanyi Liu and Rafael Pass. "Cryptography from Sublinear Time Hardness of Time-bounded Kolmogorov Complexity". In: *STOC*. 2021 (cit. on p. 4).

[LP21b] Yanyi Liu and Rafael Pass. "On the Possibility of Basing Cryptography on EXP $\neq$ BPP EXP$\neq$ BPP". In: *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*. Springer. 2021, pp. 11–40 (cit. on p. 18).

[LP22] Yanyi Liu and Rafael Pass. "On One-Way Functions from NP-Complete Problems". In: *37th Computational Complexity Conference*. 2022 (cit. on p. 13).

[LP23] Yanyi Liu and Rafael Pass. "Leakage-resilient hardness vs randomness". In: (2023) (cit. on p. 22).

[LP24] Yanyi Liu and Rafael Pass. "On One-Way Functions, the Worst-Case Hardness of Time-Bounded Kolmogorov Complexity, and Computational Depth". In: *Theory of Cryptography Conference*. Springer. 2024, pp. 222–252 (cit. on pp. 4, 13, 14).

[LP25] Yanyi Liu and Rafael Pass. "Hardness along the boundary: Towards one-way functions from the worst-case hardness of time-bounded kolmogorov complexity". In: *Annual International Cryptology Conference*. Springer. 2025, pp. 617–650 (cit. on pp. 4, 5, 12–14, 16, 17).

[MP24] Noam Mazor and Rafael Pass. "Gap MCSP Is Not (Levin) NP-Complete in Obfustopia". In: *39th Computational Complexity Conference (CCC 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2024, pp. 36–1 (cit. on pp. 3, 13).

[MV99] P. B. Miltersen and N. V. Vinodchandran. "Derandomizing Arthur-Merlin Games Using Hitting Sets". In: *focs40*. 1999, pp. 71–80 (cit. on p. 5).

[MW17] Cody D Murray and R Ryan Williams. "On the (non) NP-hardness of computing circuit complexity". In: *Theory of Computing* 13.1 (2017), pp. 1–22 (cit. on p. 13).

[Nao91] Moni Naor. "Bit commitment using pseudorandomness". In: *Journal of cryptology* 4.2 (1991), pp. 151–158 (cit. on pp. 4, 10, 25, 36).

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs randomness". In: *Journal of computer and System Sciences* 49.2 (1994), pp. 149–167 (cit. on pp. 8, 23).

[NY89] Moni Naor and Moti Yung. "Universal One-Way Hash Functions and their Cryptographic Applications". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*. 1989, pp. 33–43 (cit. on p. 4).

[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 für Informatik. 2019, pp. 32–1 (cit. on pp. 6, 7).

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Functions are Essential for Non-Trivial Zero-Knowledge". In: *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems*. IEEE Computer Society, 1993, pp. 3–17 (cit. on p. 5).

[Pud03] Pavel Pudlák. "On reducibility and symmetry of disjoint NP pairs". In: *Theoretical Computer Science* 295.1-3 (2003), pp. 323–339 (cit. on p. 4).

[Pud17] Pavel Pudlák. "Incompleteness in the finite domain". In: *Bulletin of Symbolic Logic* 23.4 (2017), pp. 405–441 (cit. on p. 4).

[Raz94] Alexander A Razborov. *On provably disjoint NP-pairs*. Aarhus Universitet. Basic Research in Computer Science [BRICS], 1994 (cit. on pp. 4, 10, 16).

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40 (cit. on p. 3).

[Rom90] John Rompel. "One-Way Functions are Necessary and Sufficient for Secure Signatures". In: 1990, pp. 387–394 (cit. on p. 4).

[RS22] Hanlin Ren and Rahul Santhanam. "A relativization perspective on meta-complexity". In: *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2022 (cit. on p. 13).

[San20] Rahul Santhanam. "Pseudorandomness and the minimum circuit size problem". In: *LIPIcs* 151 (2020) (cit. on p. 18).

[Sha04] Ronen Shaltiel. "Recent developments in explicit constructions of extractors". In: *Current Trends in Theoretical Computer Science: The Challenge of the New Century Vol 1: Algorithms and Complexity Vol 2: Formal Models and Semantics* (2004), pp. 189–228 (cit. on pp. 21, 22).

[Sip83] Michael Sipser. "A Complexity Theoretic Approach to Randomness". In: 1983, pp. 330–335 (cit. on p. 3).

[Sol64] R.J. Solomonoff. "A formal theory of inductive inference. Part I". In: *Information and Control* 7.1 (1964), pp. 1 –22. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(64)90223-2 (cit. on p. 3).

[SS20] Michael Saks and Rahul Santhanam. "Circuit lower bounds from NP-hardness of MCSP under Turing reductions". In: *LIPIcs* 169 (2020) (cit. on p. 13).

[SS22] Michael Saks and Rahul Santhanam. "On randomized reductions to the random strings". In: *37th Computational Complexity Conference (CCC 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2022 (cit. on p. 13).

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. "Pseudorandom generators without the XOR lemma". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 1999, pp. 537–546 (cit. on p. 8).

[SU05] Ronen Shaltiel and Christopher Umans. "Simple extractors for all min-entropies and a new pseudorandom generator". In: *Journal of the ACM (JACM)* 52.2 (2005), pp. 172–216 (cit. on p. 5).

[Tra84] Boris A Trakhtenbrot. "A survey of Russian approaches to perebor (brute-force searches) algorithms". In: *Annals of the History of Computing* 6.4 (1984), pp. 384–400 (cit. on pp. 3, 8).

[Tre99] Luca Trevisan. "Construction of extractors using pseudo-random generators". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 1999, pp. 141–148 (cit. on p. 22).

[Vad+12] Salil P Vadhan et al. "Pseudorandomness". In: *Foundations and Trends® in Theoretical Computer Science* 7.1–3 (2012), pp. 1–336 (cit. on pp. 9, 24).

[Yao82] Andrew C. Yao. "Theory and Applications of Trapdoor Functions". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 80–91 (cit. on p. 6).

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il