Electronic Colloquium on Computational Complexity, Report No. 322 (2026)

Complexity-Theoretic Consequences of Zero-Knowledge with Inefficient Simulation

Gal Arnon[1], Noam Mazor[2], Rafael Pass[3], and Jad Silbak[4]

[1]NTT Research, galarnon42@gmail.com
[2]New York University, noammaz@gmail.com
[3]Technion, Cornell Tech, and Tel Aviv University, rafael@cs.cornell.edu
[4]MIT, jadsilbak@gmail.com

November 19, 2026

Abstract

Zero-knowledge (ZK) protocols are one of the most fundamental cryptographic primitives. Classic results in the 1980s established that one-way functions (OWFs) suffice for constructing ZK for NP (Goldreich-Micali-Wigderson [JACM'91], Naor [CRYPTO'89], Hastad et al. [SICOMP'99]). Moreover, OWFs are, in fact, necessary for ZK for NP, due to the classic result of Ostrovsky and Wigderson [ISTCS'93], which was recently strengthened by Hirahara and Nanashima [STOC'24].

But for many applications of ZK, it suffices to consider a slight relaxation of the ZK property where the simulator is allowed to run in super-polynomial time (Pass [Eurocrypt'03]). Surprisingly, the results of Ostrovsky and Wigderson break once we allow even a slightly inefficient simulator. Indeed, a recent result by Baril and Haitner [CRYPTO'26] shows that, without efficient simulation, ZK does not imply one-way functions in a black-box manner.

We show that two-message arguments with a (non-trivial) inefficient simulator imply the errorless average-case hardness of the Minimum Time-Bounded Kolmogorov Complexity Problem (MK^tP) over the uniform distribution. The hardness of the same problem against *two-sided* error algorithms is known to imply OWFs (Liu and Pass, [FOCS'20]).

As far as we know, this result is the first complexity-theoretic consequence of ZK with even just slightly super-polynomial-time simulation.

ISSN 1433-8092

Contents

1 Introduction **1**
- 1.1 Technical Overview . . . 3

2 Preliminaries **6**
- 2.1 Complexity Classes and NP Relations . . . 7
- 2.2 Kolmogorov Complexity . . . 7
- 2.3 Auxiliary-Input One-Way Functions . . . 8
- 2.4 Computational Extraction . . . 8
- 2.5 Interactive Protocols . . . 9

3 Conditional PRG and Next-Bit Unpredictability **10**
- 3.1 Conditional PRG . . . 10
- 3.2 Next-Bit Unpredictability . . . 10

4 Simulatable Two-Message Arguments to Next-Bit Unpredictability **11**
- 4.1 Proving Claim 4.4 . . . 13
- 4.2 Proving Lemma 4.5 . . . 14

5 Dealing with Private Coins **15**

6 Unpredictability to Pseudorandomness: Proving Theorem 3.5 **16**
- 6.1 Proving Lemma 6.2 . . . 17
- 6.2 Proving Lemma 6.3 . . . 19

A Unique Succinct Non-interactive Arguments **23**

B Deferred Proofs **24**
- B.1 Proving Lemma 2.4 . . . 24
- B.2 Proving Lemma 2.11 . . . 25

1 Introduction

Establishing the minimal complexity assumptions required for basic cryptographic primitives, as well as the connections between these primitives, is a foundational question in cryptography. Most basic cryptographic objects (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), rely on the existence of one-way functions (OWFs) [DH76], which are functions that are easy to compute but hard to invert. Additionally, all of those primitives are known to imply the existence of OWFs [IL89].

Zero-knowledge (ZK). In this work, we focus on the fundamental notion of a zero-knowledge (ZK) proof [GMR89; BCC88]: ZK proofs are paradoxical constructs that enable a Prover to convince a Verifier that some instance x belongs to a language $\mathcal{L}$ 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 "simulated" indistinguishably by an *efficient*, polynomial-time, simulator. The polynomial-time requirement on the simulator formalizes the intuition that "anything an efficient verifier can learn in the interaction, could have been generated without interaction (by simply running the simulator)". Classic results from the 1980s established that OWFs imply ZK proof for all of NP [GMW91; Nao91; HILL99]. Additionally, Ostrovsky and Wigderson [Ost91; OW93] show that auxiliary-input one-way functions[1] are necessary for ZK arguments assuming worst-case hardness of NP.[2] Recently, Hirahara and Nanashima [HN24] (see also [LMP24]) extend these results to show that, assuming worst-case hardness of NP, ZK arguments imply (standard) one-way functions, and a more recent line of work extended this result to arguments satisfying a notion of "weak"[3] ZK [CHK25; CLV26; CHKT26].

Zero-knowledge with Inefficient Simulation. In most applications of ZK, it suffices to consider a slight variant of ZK where the simulator is *inefficient*. For instance, if the goal is to protect some information that is poly(T)-hard to recover before interaction, then it suffices to employ a ZK proof where the simulator runs in time T. Indeed, [Pas03] introduced such a notion of ZK with inefficient simulators—referred to as super-polynomial-time simulation (SPS) ZK—and showed that many barriers of ZK protocols (e.g., [GO94a]) can be overcome once we relax the running-time requirement on the simulation.

The Complexity of ZK with Inefficient Simulators. While both WI and SPS-ZK have been shown to overcome classic barriers of ZK proofs, the complexity-theoretic assumptions needed to construct them do not differ from those needed to construct ZK. In fact, essentially all constructions of these weaker primitives start off with some "underlying" ZK protocol and thus inherit the complexity-theoretic assumptions (notably OWFs).

Yet, surprisingly, the result of Ostrovsky and Wigderson [OW93] (and subsequent improvements) breaks down once we consider even just a slightly super-polynomial-time simulator. The reason for this is that the approach is to use the assumed ZK simulator as a one-way function: roughly speaking, the core idea is that if a malicious prover can efficiently invert the simulator, then it can convince the verifier on a false statement, breaking the (computational) soundness of the protocol (unless the language is already trivial). The fact that the simulator "is the OWF" means that a super-poly simulator would at best yield a super-polynomial-time computable OWF. But for technical reasons, even that does not work![4]

Indeed, as shown by recent work of Baril and Haitner [BH26], the general proof approach of Ostrovsky and Wigderson *cannot* be extended to work for ZK with inefficient simulators: they showed that

[1] An auxiliary-input one-way function is a function family $\{f_a\}$ so that for every efficient adversary A there exists a value a so that f_a is hard for A to invert (even knowing a).

[2] Ostrovsky and Wigderson additionally show that, assuming the *average-case* hardness of NP, *standard* one-way functions are necessary for zero-knowledge.

[3] Weak ZK refers to zero-knowledge protocol where the simulation is only "weakly" indistinguishably from real executions.

[4] It would seem that to get any consequences of [OW93] for the setting of inefficient simulators, we need to consider a notion of "strong T-simulation" [Pas03; Pas04] (where the simulator's output also is indistinguishable w.r.t. poly(T)-distinguishers, and additionally consider soundness against poly(T) attackers. Strong SPS simulation, however, is a significantly stronger notion than just SPS—in fact, it is orthogonal to even just plain ZK.

there is no *black-box construction* of (auxiliary-input) OWF from one-message perfect zero-knowledge, as long as the reduction does not have access to the simulator, even under complexity assumptions such as $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. This gives a strong barrier for establishing the necessity of one-way functions for ZK with inefficient simulation.

Kolmogorov Complexity. To present our results, recall that in the *Time-Bounded Kolmogorov Complexity Problem (*$\mathrm{MK^tP}$*)* [Kol68; Sol64; Cha69; Ko86; Har83; Sip83], the goal is to determine 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 $t = \mathrm{poly}(|x|)$. Formally, we define $\mathrm{MK^tP}[s] = \{x\colon \mathrm{K}^t(x) \leq s(|x|)\}$. The hardness of computing K^t is strongly connected to one-way functions. Indeed, Liu and Pass [LP20] showed that the hardness of solving $\mathrm{MK^tP}$ on the uniform distribution for algorithms with two-sided error characterizes the existence of one-way functions.

Our Results. We prove that two-message arguments (i.e., there is a verifier message and then a prover message) with inefficient but nontrivial simulation imply the average-case hardness of $\mathrm{MK^tP}$ on the uniform distribution, for *errorless* algorithms. Recall that an algorithm A is errorless for a language $\mathcal{L}$ if, for every input x, $\mathbf{Pr}\left[A(x) \in \{\mathcal{L}(x), \perp\}\right] \geq 2/3$. We say that A decides $\mathcal{L}$ over the uniform distribution if $\mathbf{Pr}_{x \leftarrow \{0,1\}^n}\left[A(x) = \perp\right] \leq 1/3$. We prove the following theorem:

Theorem 1.1 (informal, Theorem 5.1)**.** *Assume a language* $\mathcal{L} \notin \mathsf{SIZE}[T_L]$ *has a two-message interactive argument with negligible completeness and soundness errors and a simulator* Sim *computable by a circuit of size* T_S *such that* $T_S(n)/T_L(n) = \mathsf{negl}(n)$*. Then there exists* $t \in \mathrm{poly}$ *such that* $\mathrm{MK^tP}[n-2]$ *is average-case hard on the uniform distribution for efficient errorless algorithms.*[5]

Our result applies to the weakest notion of security with regard to just an *honest verifier*, and applies even just to interactive *arguments* (i.e., where soundness holds only against non-uniform efficient adversaries). Moreover, it is enough for us to consider a simulator that simulates only the transcript of the protocol rather than the verifier's view. In the above theorem, we assume that the honest prover is efficient given an instance-witness pair (i.e., we only consider *efficient-prover* protocols); this is the only relevant case for cryptographic purposes.[6]

We note that Hirahara [Hir18] showed that the worst-case hardness of a gap version of $\mathrm{MK^tP}$ implies the average-case hardness of $\mathrm{MK^tP}$ for errorless algorithms. In this work we prove the average-case hardness directly by constructing a weak version of a PRG. See Section 1.1 for details.

As an application of the above theorem, we observe that succinct non-interactive arguments that have a unique proof for each instance (unique SNARGs, [FW26]) can be simulated in time proportional to 2^c, where c is the length of the proof. As a result we directly get the following corollary.

Corollary 1.2 (informal, Corollary A.4)**.** *Assume a language* $\mathcal{L} \notin \mathsf{SIZE}[T_L]$ *has a Unique SNARG with message length* $c(n) = \log T_L - \omega(\log n)$*. Then there exists* $t \in \mathrm{poly}$ *such that* $\mathrm{MK^tP}[n-2]$ *is average-case hard on the uniform distribution for efficient errorless algorithms.*

Identifying the minimal assumptions for succinct proof systems such as SNARGs is an important long-standing open problem in cryptography. We view Corollary 1.2 as an indication that Theorem 1.1 suggests a promising path toward this goal.[7]

Discussion. We see our results as a step toward proving that OWFs, or so-called auxiliary-input OWFs (AI-OWFs) [OW93], are necessary for constructing such arguments. Indeed, while it is known that OWFs imply the hardness of $\mathrm{MK^tP}$ [RR94; KC00], and also even just AI-OWFs [AH19; LMP24], as far as we know no other assumptions are known to imply hardness of $\mathrm{MK^tP}$ (that do not themselves already imply AI-OWFs).[8] Notably, the question whether $\mathrm{MK^tP}[n-2]$ is NP-complete is still open,

[5] In the main body of this paper we prove the theorem assuming $\mathcal{L}$ is not contained within the infinitely-often variant of $\mathsf{SIZE}[T_L]$. This is purely for convenience, and all of the techniques readily translate to the case of $\mathcal{L} \notin \mathsf{SIZE}[T_L]$.

[6] Additionally, note that inefficient simulation might be trivial for inefficient provers who can "canonicalize" the witness.

[7] See Remark 1.5 for discussion on simulating general SNARGs.

[8] One exception is *hitting-set generator* against poly-time algorithms, but the existence of this primitive is almost directly equivalent to the hardness of $\mathrm{MK^tP}$. Indeed, assuming $\mathrm{MK^tP}[s]$ is hard, it is not hard to see that $T(x) = \mathrm{U}(x, 1^t)$ is an hitting set generator. Similarly, if T is an hitting set computed in time t and with seeds of length s, then $\mathrm{MK^tP}[s + O(1)]$ is worst-case hard.

and, while there has been a lot of recent progress showing that related problems are NP-complete [Ila20; ILCO20; LP22; Hir22; HIR23; Ila23; IKV23; HI25], there is strong evidence that it is not NP-complete [AH19; Ko91; RS22; MW17; KC00; SS20; SS22; MP24; LMP26].

As such, Theorem 1.1 (and Corollary 1.2) give the first primitives known to imply the hardness of $\mathrm{MK^tP}$ but not known to imply AI-OWFs. We also point out that Theorem 1.1 shows that, in order to prove that two-message simulatable arguments imply OWFs, it is enough to show that the errorless average-case hardness of $\mathrm{MK^tP}$ implies its two-sided error average-case hardness. While bridging the gap between errorless average-case hardness and two-sided average-case hardness of $\mathrm{MK^tP}$ is already a well-motivated important open question, it also suggests a path toward bypassing the barrier of [BH26]. Indeed, non-black-box techniques were developed in the literature on Kolmogorov complexity and "meta-complexity" and used to bypass existing black-box barriers (e.g. [Hir18]).

Relationship to [HI25]. It is worthwhile to compare Theorem 1.1 with the recent (and independently of the current work), breakthrough result of Hirahara and Ilango [HI25] showing that the Minimum Circuit Size Problem (MCSP) is NP-complete with respect to quasi-polynomial non-adaptive reductions based on the following assumptions: (a) Subexponentially-secure non-interactive witness-indistinguishable proofs (NIWI) for NP, (b) coNP requires subexponential-size non-deterministic circuits, and (c) $\mathsf{P}^{\mathsf{NP}}/\mathrm{poly}$ requires circuits of size $\Omega(2^n/n)$. This result can be interpreted as showing that, under these assumptions, subexponentially-secure NIWI implies hardness of MCSP.

In contrast, we *unconditionally* show that ZK with non-trivial inefficient simulators yields hardness of $\mathrm{MK^tP}$.

1.1 Technical Overview

In this section, we outline the proof of Theorem 1.1. Our goal is to prove the hardness of approximating K^t using an argument system with inefficient honest-verifier view simulation. Let (P, V) be a two-message argument system (i.e., a verifier message and then a prover message) for a hard language $\mathcal{L}$, and let Sim be an inefficient simulator for the verifier's view. For simplicity, in this overview, we assume that the argument has perfect completeness, meaning that, for every $(x, w) \in \mathcal{R}_{\mathcal{L}}$, $P(x, w)$ convinces $V(x)$ with probability 1. We let T_S be the size of the circuit computing Sim. T_S is allowed to be large, but we assume that it is significantly smaller than the circuit size $T_{\mathcal{L}}$ needed to decide $\mathcal{L}$. We further assume the existence of distributions $\mathcal{D}^Y$ and $\mathcal{D}^N$ where $\mathcal{D}^Y$ is over pairs $(x, w) \in \mathcal{R}_{\mathcal{L}}$, and $\mathcal{D}^N$ is over $x \notin \mathcal{L}$, such that $(x, \cdot) \leftarrow \mathcal{D}^Y$ and $x \leftarrow \mathcal{D}^N$ are indistinguishable for $T_{\mathcal{L}}$ size-circuits. Such (inefficiently sampleable) distributions exist assuming the worst-case hardness of $\mathcal{L}$ (see Lemma 2.4).

Conditional PRGs. Our approach is based on the notion of conditional PRG [LMP26]. A conditional PRG is an efficiently computable function $G\colon \{0,1\}^m \to \{0,1\}^\ell$ (for $\ell > m$) where there is some distribution $\mathcal{D}$ over $\{0,1\}^m$, such that $G(\mathcal{D})$ is pseudorandom. [LMP26] show that the existence of such a PRG implies the (errorless) average-case hardness of $\mathrm{MK^tP}[n-2]$ over the uniform distribution, for errorless algorithms.

Theorem 1.3 ([LMP26], see Theorem 3.2)**.** *Assume the existence of a conditional PRG. Then* $\mathrm{MK^tP}[n-2]$ *is average-case hard on the uniform distribution for errorless algorithms.*

Indeed, every possible output $y = G(x)$ of G has time-bounded Kolmogorov complexity at most $|x| + O(1) = m + O(1)$ (since G and x can be used to describe y, and G is a uniform algorithm), but pseudorandomness implies that it is hard to distinguish between images of G and uniform strings of length ℓ (that have Kolmogorov complexity close to ℓ with high probability).

Following Theorem 1.3, our goal is to construct a conditional PRG from the existence of two-message arguments with inefficient simulation. Liu and Pass [LP20] showed that an entropy-preserving conditional PRG (that is, the entropy of the output is close to the seed length) implies the average-case hardness of $\mathrm{MK^tP}[n-2]$ for two-sided error algorithms, which in turn implies the existence of a one-way function. Unfortunately, our conditional PRG will be far from being entropy-preserving.

Below we explain how to construct conditional PRGs from public-coin two-message arguments with inefficient simulation of the honest verifier's view.[9] The generalization to private-coin protocols

[9]Recall that in a public-coin protocol all of the verifier's messages are uniformly and independently sampled random strings.

follows from the observation that two-message protocols can be made public-coin if there are no one-way functions.

1.1.1 Conditional PRG from Arguments with Deterministic Prover

We explain how to use P to construct a conditional PRG. As a first step, we make the simplifying assumption that P is deterministic (we will discuss how to remove this assumption in Section 1.1.2). Let $a(n)$ be the length of the message sent by the verifier to the prover, and $b(n)$ the length of the message sent back from the prover.

We construct a weak pseudorandom function family F, where the keys are inefficiently sampleable. In more detail, the keys for F are sampled from $\mathcal{D}^Y$. For $(x,w) \leftarrow \mathcal{D}^Y$, we let $F_{x,w} \colon \{0,1\}^{a(n)+b(n)} \to \{0,1\}$ be the function defined by

$$F_{x,w}(\rho, r) = \mathsf{GL}(P(x,w,\rho), r),$$

where ρ is the message from the verifier and $\mathsf{GL}(P(x,w,\rho),r)$ is the Goldriech-Levin hard-core predicate [GL89]. We note that, given x, w and an input ρ, r, the function $F_{x,w}(\rho, r)$ can be efficiently computed. Below, in Claim 1.4, we show that F is a weak PRF, meaning that for a random key $(x,w) \leftarrow \mathcal{D}^Y$, and for every $k \in \text{poly}$, it is hard to distinguish between k random input-output pairs of $F_{x,w}$, $((\rho_1, r_1, F_{x,w}(\rho_1, r_1)), \ldots, (\rho_k, r_k, F_{x,w}(\rho_k, r_k)))$ and a uniform string of length $k(a+b+1)$.

Observe that showing that F is a weak PRF suffices for constructing a conditional PRG by letting

$$G(x, w, \rho_1, r_1, \ldots, \rho_k, r_k) = ((\rho_1, r_1, F_{x,w}(\rho_1, r_1)), \ldots, (\rho_k, r_k, F_{x,w}(\rho_k, r_k))),$$

and $\mathcal{D} = \mathcal{D}^Y || \mathbf{U}_{k(a+b)}$. It is not hard to see that $G(\mathcal{D})$ is indeed pseudorandom if F is a weak PRF, and that G is stretching when $k > |x| + |w|$.

Claim 1.4. *F is a weak PRF.*

Proof sketch. Assume towards a contradiction that this is not the case. Then, by a simple hybrid argument, there is some index i such that the following distributions are distinguishable:

- $((\rho_1, r_1, F_{x,w}(\rho_1, r_1)), \ldots, (\rho_{i-1}, r_{i-1}, F_{x,w}(\rho_{i-1}, r_{i-1})), (\rho_i, r_i, F_{x,w}(\rho_i, r_i)))$, and
- $((\rho_1, r_1, F_{x,w}(\rho_1, r_1)), \ldots, (\rho_{i-1}, r_{i-1}, F_{x,w}(\rho_{i-1}, r_{i-1})), (\rho_i, r_i, \mathbf{U}_1))$.

Note that r_i is uniformly random and, as the protocol is public-coin, this is also true for ρ_i. Recall that $F_{x,w}(\rho_i, r) = \mathsf{GL}(P(x,w,\rho_i), r)$ is a Goldreich-Levin hard-core bit of $P(x,w,\rho)$. This means that using the assumed distinguisher, there is an efficient algorithm Rec, that given samples

$$((\rho_1, r_1, F_{x,w}(\rho_1, r_1)), \ldots, (\rho_{i-1}, r_{i-1}, F_{x,w}(\rho_{i-1}, r_{i-1}))) \tag{1}$$

and a uniformly random verifier message ρ_i, "recovers" with a noticeable probability the message $m = P(x,w,\rho_i)$, which is a convincing prover message such that $V(x, \rho_i, m)$ accepts. Our goal is to use Rec to fool the verifier on a No instance, thus violating the soundness of the protocol (P, V).

Towards this goal, we first notice that we can replace the samples that Rec is given (i.e., as in Eq. (1)) with simulated proofs. That is, we claim that for a random $(x,w) \leftarrow \mathcal{D}^Y$, and given

$$(\text{Sim}(x; r_1), \ldots, \text{Sim}(x; r_{i-1}))$$

and a random verifier message ρ_i, Rec still outputs a convincing proof m with $V(x, \rho_i, m) = 1$ with noticeable probability. Indeed, otherwise we could construct a small circuit that distinguishes between $\text{Sim}(x)$ and the real interaction.

Finally, we show that Rec outputs a convincing proof even when x is sampled from $\mathcal{D}^N$. Indeed, assume that for $x \leftarrow \mathcal{D}^N$, and given $(\text{Sim}(x; r_1), \ldots, \text{Sim}(x; r_{i-1}))$ and a random verifier message ρ_i, Rec does not output a convincing proof m with $V(x, \rho_i, m) = 1$. We construct a circuit C of size $< T_L$ that distinguishes between $(x, \cdot) \leftarrow \mathcal{D}^Y$ and $x \leftarrow \mathcal{D}^N$. Let C be the circuit that given x, computes i random simulated proofs on x $(\text{Sim}(x; r_1), \ldots, \text{Sim}(x; r_{i-1}))$, and executes Rec on a uniformly random verifier message ρ_i to get a prover message m. Then, if $V(x, \rho_i, m)$ accepts then C outputs 1, and otherwise C outputs 0. The circuit C outputs 1 on a sample from $\mathcal{D}^Y$ with noticeable probability.

It follows that if C does not output 1 with noticeable probability on samples from $\mathcal{D}^N$, then C can distinguish between $\mathcal{D}^Y$ and $\mathcal{D}^N$. But the size of C is at most

$$\text{poly}(n) \cdot (|\mathsf{Sim}| + |\mathsf{Rec}| + |V|) = \text{poly}(n) \cdot T_S \ll T_{\mathcal{L}},$$

which is a contradiction to the hardness of $\mathcal{L}$. Thus, Rec is able to generate accepting proofs for No instances, breaking the soundness of (P, V). □

1.1.2 Dealing with a Randomized Prover

The approach described above fails when the prover algorithm is randomized. The problem is that P gets additional (potentially large) randomness as input, and so in order to get a stretch for the conditional PRG, it does not suffice to output one bit per proof as we did earlier. At the very least, we need to extract the randomness used by the prover.

A natural approach is to output more hardcore bits. For example, if the min-entropy of the output of P is k, we can output $k+1$ GL hard-core bits that will appear to be pseudorandom. However, we do not necessarily have a bound on the min-entropy of P, and so we cannot do this.

We resolve this issue by using ideas from constructions of PRGs from one-way functions. Specifically, we use the notion of bits unpredictability, defined in [MP23]. Roughly, for a randomized prover we define

$$F_{x,w}(\rho, r_1, \ldots, r_b) = \mathsf{GL}(P(x, w, \rho), r_1), \ldots, \mathsf{GL}(P(x, w, \rho), r_b).$$

While we do not know whether the $b(n)$-bit output of F is pseudorandom, we are able to show that for an online algorithm that reads the output bit-by-bit, more bits appear to be random than the number of random bits in the input. In the language of [MP23], this is called a bits-unpredictability generator, and such a generator can be manipulated to construct a (conditional) PRG. We highlight that this part of the proof is highly non-trivial and, interestingly, the final construction is similar to the constructions of PRG from one-way functions. See Section 3 for more detail.

Remark 1.5 (Inefficient Simulation and SNARGs)**.** *At first glance it might seem that our result can be extended to Succinct Non-Interactive Arguments (SNARGs). Indeed, by the result of [GW11], such an argument can be simulated by a simulator of size roughly 2^ℓ, where ℓ is the length of the proof. When 2^ℓ is small enough compared to the time needed to decide the language, our result should yield hardness of* $\mathrm{MK^tP}$.

However, the simulation implied by [GW11] is weaker than what we need. [GW11] promises that for every distribution X over Yes-instances of the language, there is a simulator, such that $(X, \mathrm{Sim}(X))$ is indistinguishable from $(X, \Pi(X))$, where $\Pi(X)$ is a valid proof for X. Note that here indistinguishability only holds on average. On the other hand, for our result, we need indistinguishability to hold for any *Yes-instance x. In particular, we need to be able to claim that k samples from $\mathrm{Sim}(x)$ are indistinguishable from k valid proofs, for $k > |x|$.*

Additional Related Works

As mentioned above, [OW93; HN24] prove that one-way functions are necessary for zero-knowledge arguments for NP, and [LMP24] extended this result to the uniform setting. [BOV07] showed that under assumptions from the literature on derandomization [NW94], it is possible to derandomize the verifier of a ZAP and construct Non-Interactive WI (NIWI). Goldreich and Oren [GO94b] showed that languages outside BPP do not have two-message ZK arguments, or that auxiliary-input ZK with a deterministic prover. [CLMP12] consider two-message arguments with super-polynomial simulator (SPS), and showed a barrier to proving soundness of such arguments using black-box reductions. [BC20] showed that without auxiliary-input, ZK with deterministic prover for NP can be constructed from strong cryptographic assumptions. Additionally, [BC20] showed that such protocols imply witness encryption. [Wee06] presented an oracle with respect to which there exists a succinct argument for a hard language in NP, but one-way functions do not exist. Finally, In related work to this paper, [AMPS26] show that, assuming NP is hard, the existence of auxiliary-input one-way functions is implied by (constant-round) WI arguments for NP that satisfy *knowledge soundness*, roughly meaning that an extractor, given black-box access to a highly-convincing prover, can derive a valid witness.

Paper Organization

- In Section 2 we present the notation and definitions we use in this paper.
- Section 3 is dedicated to describing the tools we use in our proof. In Section 3.1 we define conditional PRGs, and explain why it is enough to construct a conditional PRG to derive Theorem 1.1. In Section 3.2 we define unpredictable bits generators, and in Section 6 we show it is enough to construct a conditional PRG, using the approach of [MP23].
- In Section 4 we construct unpredictable bits generator from public-coin protocols.
- In Section 5 we generalize the result for private-coin protocols, and prove Theorem 1.1.
- Finally, the definitions and proof for our observation on unique SNARGs is given in Section A.

2 Preliminaries

General notation. We use the following conventions throughout the paper:

- All logarithms are taken in base 2.
- Calligraphic letters denote sets and distributions. Uppercase letters denote random variables; lowercase letters denote values and functions.
- poly denotes the set of polynomially bounded functions. PPT denotes probabilistic polynomial-time algorithms. A non-uniform polynomial-time algorithm A is equipped with a fixed polynomial-size advice sequence $\{z_n\}_{n\in\mathbb{N}}$ (typically omitted from notation), and we write A_n for A with advice z_n on inputs of length n.
- For a randomized algorithm A, we write $A(\cdot; r)$ for A with randomness fixed to $r \in \{0,1\}^*$. We write negl for a negligible function.
- For $n \in \mathbb{N}$, we let $[n] \coloneqq \{1, \ldots, n\}$. For $i \leq j$, we let $[i : j] \coloneqq \{i, \ldots, j\}$.
- For a vector $v \in \Sigma^n$, we write v_i for its i-th entry, $v_{<i} = (v_1, \ldots, v_{i-1})$, and $v_{\leq i} = (v_1, \ldots, v_i)$. For $\mathcal{I} \subseteq [n]$, we write $v_{\mathcal{I}}$ for the ordered tuple $(v_i)_{i\in\mathcal{I}}$.
- For strings $x, y \in \{0,1\}^*$, we write $x||y$ for their concatenation.

Probability notation. When unambiguous, we will naturally identify a random variable with its distribution. The support of a finite distribution $\mathcal{P}$ is defined by $\mathrm{Supp}(\mathcal{P}) \coloneqq \{x\colon \mathbf{Pr}_{\mathcal{P}}[x] > 0\}$. 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). For $p \in [0,1]$ we use U_p to denote the random variable that takes 1 with probability p and 0 otherwise.

A collection of distributions $\{\mathcal{D}_a\}_{a\in\{0,1\}^{\lambda(n)},n\in\mathbb{N}}$ is samplable if there exists a $\mathrm{poly}(n)$-time randomized algorithm S such that for every $n \in \mathbb{N}$ and every $a \in \{0,1\}^{\lambda(n)}$ $S(a) \equiv \mathcal{D}_a$.

Statistical distance. The statistical distance of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P}, \mathcal{Q}) \coloneqq \max_{\mathcal{S}\subseteq\mathcal{X}} |\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{X}} |\mathcal{P}(x) - \mathcal{Q}(x)|$.

Entropy. For a random variable X, let $H(X) = \mathbf{E}_{x\leftarrow X}\left[\log \frac{1}{\mathbf{Pr}[X=x]}\right]$ denote the **(Shannon) entropy** of X, and let $\mathrm{H}_\infty(X) = \min_{x\in Supp(X)}\left\{\log \frac{1}{\mathbf{Pr}[X=x]}\right\}$ denote the **min-entropy** of X.

We will make use of the following facts in probability.

Lemma 2.1 (Coupling)**.** *Let $\mathcal{D}_A, \mathcal{D}_B$ be two distributions. Then there exists a triplet of jointly distributed random variables (A, B, E), such that the following hold:*

- *The marginal distribution of A is $\mathcal{D}_A$ and the marginal distribution of B is $\mathcal{D}_B$.*
- $\mathbf{Pr}[E \neq 1] = \mathrm{SD}(\mathcal{D}_A, \mathcal{D}_B)$*, and,*
- $\mathbf{Pr}[A = B | E = 1] = 1$.

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

Lemma 2.2 (Chernoff bound)**.** *Let $A_1, ..., A_n$ be independent random variables s.t. $A_i \in \{0,1\}$. Let $\widehat{A} = \Sigma_{i=1}^{n} A_i$ and $\mu = \mathbf{E}\left[\widehat{A}\right]$. For every $\epsilon \in [0,1]$ It holds that:*

$$\mathbf{Pr}\left[|\widehat{A} - \mu| \geq \epsilon \cdot \mu\right] \leq 2 \cdot e^{-\epsilon^2 \cdot \mu/3}.$$

2.1 Complexity Classes and NP Relations

For a language $\mathcal{L} \subseteq \{0,1\}^*$, let $\mathcal{L}_n = \mathcal{L} \cap \{0,1\}^n$, and for $\mathcal{L} \in \mathsf{NP}$ let $\mathcal{R}(\mathcal{L}) \subseteq \mathcal{L} \times \{0,1\}^*$ be a canonical NP relation. Similarly, for an NP relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, we let $\mathcal{L}(\mathcal{R}) = \{x|\ \exists w : (x,w) \in \mathcal{R}\}$. For a string x, we let $\mathcal{L}(x)$ be the indicator function that outputs 1 if $x \in \mathcal{L}$ and 0 otherwise.

$\mathsf{SIZE}[T(n)]$. We use $\mathsf{SIZE}[T(n)]$ to denote the class of languages $\mathcal{L}$ that, for every n, have a circuit of size at most $T(n)$ that decides $\mathcal{L}_n$. We use $\mathsf{ioSIZE}[T(n)]$ to denote the class of languages $\mathcal{L}$ that have such a circuit of infinitely many $n \in \mathbb{N}$.

The SAT language and relation. We define the canonical relation for the NP complete language SAT to be $\mathcal{R}_{\mathsf{SAT}} = \{(\varphi, v)\colon \varphi$ is a Boolean formula and v is an assignment such that $\varphi[v] = 1\}$. The language OR-SAT consists of all instances of the form $\varphi := (\varphi_0 \vee \varphi_1)$ where $|\varphi_0| = |\varphi_1|$ and φ is a satisfiable SAT formula. The canonical relation $\mathcal{R}_{\mathsf{OR\text{-}SAT}}$ is defined similarly to $\mathcal{R}_{\mathsf{SAT}}$.

Definition 2.3 ($\mathsf{AvgBPP/poly}$)**.** *A pair $(\mathcal{L}, \mathcal{D})$ where $\mathcal{L}$ is a language and $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ is a samplable distribution is in* $\mathsf{AvgBPP/poly}$ *if there exists a non-uniform PPT A such that the following holds for every large enough $n \in \mathbb{N}$:*

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

The following lemma is proven in Section B.1

Lemma 2.4. *Assume that $\mathcal{L} \notin \mathsf{ioSIZE}[T(n)]$. Then, there exists a pair of distributions ensemble $\mathcal{D}^Y = \{\mathcal{D}_n^Y\}_{n\in\mathbb{N}}, \mathcal{D}^N = \{\mathcal{D}_n^N\}_{n\in\mathbb{N}}$ with $\mathcal{D}_n^Y \subseteq \mathcal{L}_n$ and $\mathcal{D}_n^N \subseteq \{0,1\}^n \setminus \mathcal{L}_n$, such that*

$$\left|\mathop{\mathbf{Pr}}_{x \leftarrow \mathcal{D}^Y}\left[A(x) = 1\right] - \mathop{\mathbf{Pr}}_{x \leftarrow \mathcal{D}^N}\left[A(x) = 1\right]\right| \leq \epsilon(n)$$

for any $A \in \mathsf{SIZE}[\frac{\epsilon^2(n)(T(n) - \mathrm{poly}(n, 1/\epsilon(n)))}{160n}]$ and any large enough n.

2.2 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 an 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 Π 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 $\mathrm{U}(\Pi, 1^t)$ 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*

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

where $|\Pi|$ is referred to as the description length *of Π.*

2.3 Auxiliary-Input One-Way Functions

Definition 2.6 (AI-OWF)**.** *An efficiently computable function family* $\{f_a\colon \{0,1\}^{m(n)} \to \{0,1\}^{m(n)}\}_{a\in\{0,1\}^{\lambda(n)}}$ *is an* auxiliary-input one-way function (AI-OWF) *if for every PPT* $\mathcal{A}$ *there exists a negligible function* μ *such that for every* $n \in \mathbb{N}$*, there exists some* $a \in \{0,1\}^{\lambda(n)}$ *such that*

$$\Pr_{x\leftarrow\{0,1\}^{m(n)}}\left[\mathcal{A}(a, f_a(x)) \in f_a^{-1}(f_a(x))\right] \leq \mu(n) \tag{2}$$

If a *is simply* $0^{\lambda(n)}$*, we refer to the family as simply a* one-way function (OWF)*.*

We say that $\{f_a\}$ *is* non-uniform-secure *if the above holds for every non-uniform PPT* $\mathcal{A}$*. We say that* $\{f_a\}$ *is* infinitely often auxiliary-input one-way function (ioAI-OWF) *if for every* $\mathcal{A}$ *the above holds for infinitely many* $n \in \mathbb{N}$*.*

We now define the notion of distributional auxiliary-input one-way functions.

Definition 2.7 (Distributional Auxiliary-input One-way functions, [IL89])**.** *An efficiently computable function family* $\{f_a\colon \{0,1\}^{m(n)} \to \{0,1\}^{m(n)}\}_{a\in\{0,1\}^{\lambda(n)}}$ *is an* ε-distributional auxiliary-input one-way function (distributional AI-OWF) *if for every* PPT $\mathcal{A}$ *and for every* $n \in \mathbb{N}$*, there exists some* $a \in \{0,1\}^{\lambda(n)}$ *such that*

$$\mathrm{SD}\Big(\big(\mathcal{A}(a, f_a(X)), f_a(X)\big), \big(X, f_a(X)\big)\Big) \geq \varepsilon$$

where X *is uniformly sampled from* $\{0,1\}^{m(n)}$*. The family* $\{f_a\}$ *is a simply a* distributional auxiliary-input one-way function *if there exists a constant* $0 < c$ *such that* $\{f_a\}$ *is* $1/n^c$*-distributional AI-OWF.*

We say that $\{f_a\}$ *is* infinitely often (ϵ-)distributional auxiliary-input one-way function *if for every* $\mathcal{A}$ *the above holds for infinitely many* $n \in \mathbb{N}$*.*

Theorem 2.8 (Distributional one-way functions imply one-way functions, [IL89])**.** *If there exists an auxiliary-input distributional one-way function then there exists an auxiliary-input one-way function.*

2.4 Computational Extraction

We will make use of the leftover hash lemma and Goldreich-Levin. In the following, we identify a matrix $M \in \{0,1\}^{n\times m}$ with a function $M\colon \{0,1\}^n \to \{0,1\}^m$ by $M(x) := x \cdot M \bmod 2$, thinking of $x \in \{0,1\}^n$ as a vector with dimension n.

Lemma 2.9 (Leftover hash lemma, [HILL99])**.** *Let* $n \in \mathbb{N}$*,* $\varepsilon \in [0,1]$*, and let* X *be a random variable over* $\{0,1\}^n$*. Let* $M \leftarrow \{0,1\}^{n\times\ell}$ *be a random matrix for* $\ell \leq \mathrm{H}_\infty(X) - 2\log 1/\varepsilon$*. Then,*

$$\mathrm{SD}((M, M(X)), (M, \mathbf{U}_\ell)) \leq \varepsilon$$

for $\mathbf{U}_\ell$ *being the uniform distribution over* $\{0,1\}^\ell$*.*

Lemma 2.10 (Goldreich-Levin [GL89])**.** *There exists an oracle-aided* PPTP *such that the following holds. Let* $n \in N$ *be a number, and* $\mathcal{D}$ *a distribution over* $\{0,1\}^n \times \{0,1\}^*$*, and let* A *be an algorithm such that*

$$\Pr_{(x,z)\leftarrow\mathcal{D},r\leftarrow\{0,1\}^n}[A(z, r, GL(x, r)) = 1] - \Pr_{(x,z)\leftarrow\mathcal{D},r\leftarrow\{0,1\}^n}[A(z, r, \mathbf{U}) = 1] \geq \alpha$$

for some α*, where* $GL(x, r) := \langle x, r\rangle$ *is the Goldreich-Levin predicate. Then*

$$\Pr_{(x,z)\leftarrow\mathcal{D}}\left[P^A(1^n, 1^{\lceil 1/\alpha\rceil}, z) = x\right] \geq \alpha^3/8n.$$

We also prove the following corollary of Goldreich-Levin in Section B.2.

Lemma 2.11 (Multi-bit Goldreich-Levin, [MP23])**.** *There exists an oracle-aided* PPT$\widehat{P}$ *such that the following holds. Let* $\ell, n \in N$ *be numbers, and* $\mathcal{D}$ *a distribution over* $\{0,1\}^n \times \{0,1\}^*$*. and let* A *be an algorithm such that*

$$\Pr_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z, M, M(x)_{\leq\ell}) = 1] - \Pr_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z, M, \mathbf{U}_\ell) = 1] \geq \alpha$$

for some α*, where* $GL(x, r) := \langle x, r\rangle$ *is the Goldreich-Levin predicate. Then*

$$\Pr_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}\left[\widehat{P}^A(1^n, 1^{\lceil 1/\alpha\rceil}, \ell, z, M, M(x)_{\leq\ell}) = x\right] \geq \alpha^3/8n\ell^4.$$

2.5 Interactive Protocols

Interactive Protocol. Let $P = (P_1, \ldots, P_k)$ and $V = (V_1, \ldots, V_k, V^*)$ be interactive algorithms (the prover and verifier, respectively). We assume $k = k(|x|) \in \text{poly}$.

For inputs x, y and random coins r_P, r_V, we denote by

$$\mathsf{tr} \leftarrow \langle P(x; r_P), V(y; r_V) \rangle$$

the transcript of their interaction, and let

$$\mathsf{View}_V^P(x, y) := (r_V, \langle P(x; r_P), V(y; r_V) \rangle)$$

denote the view of V in the interaction. When we omit r_P, r_V, they are sampled uniformly and independently.

The transcript has the form $\mathsf{tr} = (\rho_1, m_1, \ldots, \rho_k, m_k)$, where for $i = 1, \ldots, k$:

$$(\rho_i, \mathsf{st}_{V,i}) = V_i(y, \mathsf{st}_{V,i-1}, m_{i-1}; r_{V,i}),$$
$$(m_i, \mathsf{st}_{P,i}) = P_i(x, \mathsf{st}_{P,i-1}, \rho_i; r_{P,i}),$$

with $\mathsf{st}_{V,0} = \mathsf{st}_{P,0} = \emptyset$ and $m_0 = \emptyset$.

We define the verifier's output as

$$\mathsf{Out}(\langle P(x), V(y) \rangle) \coloneqq V^*(y, \mathsf{st}_{V,k}, m_k).$$

(During this paper, sometimes we allow the verifier's decision procedure to be randomized. We do not count this randomness towards the message count.)

We write $\mathsf{tr}_{<i} = (\rho_1, m_1, \ldots, \rho_{i-1}, m_{i-1})$ and $\mathsf{tr}_{\leq i} = (\rho_1, m_1, \ldots, \rho_i, m_i)$.

Structural properties.

- The protocol is **public-coin** if for every i, $\rho_i = r_{V,i}$ and $\mathsf{st}_{V,i} = \emptyset$.
- The protocol is γ**-unique** for $\mathcal{R}$ if for every x, w_0, w_1 with $(x, w_0), (x, w_1) \in \mathcal{R}$,

$$\Pr_{r_P, r_V}\left[\langle P(x, w_0; r_P), V(x; r_V) \rangle \neq \langle P(x, w_1; r_P), V(x; r_V) \rangle\right] \leq \gamma.$$

- The protocol is γ**-succinct** if for every $(x, w) \in \mathcal{R}$,

$$|\langle P(x, w), V(x) \rangle| \leq (|x| + |w|)^\gamma.$$

Definition 2.12 (Interactive Proofs and Arguments, Completeness and Soundness)**.** *An* interactive protocol *for a relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is a pair of PPT interactive machines (P, V) such that:*

- ***Completeness:*** *For every $(x, w) \in \mathcal{R}$,*

$$\mathbf{Pr}\left[\mathsf{Out}(\langle P(x, w), V(x) \rangle) = 1\right] \geq 1 - \alpha(|x|).$$

- ***Computational Soundness:*** *For every $x \notin L_\mathcal{R}$ and every non-uniform PPT prover $\widehat{P}$,*

$$\mathbf{Pr}\left[\mathsf{Out}(\langle \widehat{P}, V(x) \rangle) = 1\right] \leq \beta(|x|).$$

- ***Statistical Soundness:*** *For every $x \notin L_\mathcal{R}$ and every (possibly inefficient) prover $\widehat{P}$,*

$$\mathbf{Pr}\left[\mathsf{Out}(\langle \widehat{P}, V(x) \rangle) = 1\right] \leq \beta(|x|).$$

We say that (P, V) is an interactive argument (IA) *for $\mathcal{R}$ if it satisfies completeness and computational soundness.*

We say that (P, V) is an interactive proof (IP) *for $\mathcal{R}$ if it satisfies completeness and statistical soundness.*

For an oracle $\mathcal{O}$, we say that a scheme is an IA with respect to $\mathcal{O}$ *(resp., an* IP with respect to $\mathcal{O}$*) if the above holds when (P, V) and the relevant cheating prover have oracle access to samples from $\mathcal{O}$. When α and β are not specified, they are assumed to be negligible functions.*

2.5.1 Interactive Arguments with Simulators

We formally define simulators for interactive argument.

Definition 2.13 (Simulator)**.** *An algorithm* Sim *is a (honest-verifier)* ϵ*-simulator for an interactive argument* (P, V) *if for every poly-size distinguisher* $D = \{D_n\}_{n\in\mathbb{N}}$*, and for every ensemble* $\{(x_n, w_n) \in \mathcal{R}_{L_n}\}_{n\in\mathbb{N}}$

$$\left|\mathbf{Pr}\left[D(\mathrm{Sim}(x_n)) = 1\right] - \mathbf{Pr}\left[D\left(\mathsf{View}_V^P(x_n, w_n)\right) = 1\right]\right| \leq \epsilon(n).$$

An external-view ϵ-simulator *is a algorithm such that*

$$\left|\mathbf{Pr}\left[D(\mathrm{Sim}(x_n)) = 1\right] - \mathbf{Pr}\left[D(\langle P(x_n, w_n), V(x_n)\rangle) = 1\right]\right| \leq \epsilon(n).$$

If Sim *is* ϵ*-simulator for every* $\epsilon \in 1/\mathrm{poly}$*, we simply call* Sim *a simulator. We say that* Sim *is* inefficient *if its running time is super-polynomial. We say that* Sim *is* unbounded *if its runtime is unbounded.*[10]

3 Conditional PRG and Next-Bit Unpredictability

In this part we describe the tools we need to prove Theorem 1.1. We define conditional PRG and unpredictable bits unpredictability generator.

3.1 Conditional PRG

To prove Theorem 1.1, we use the definition of conditional PRG, taken from [LMP26].

Definition 3.1 (Conditional PRG)**.** *An efficiently computable function* $G\colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ *is* ϵ*-conditional PRG if* $\ell(n) > m(n)$*, and there exists a distribution* $\mathcal{D}_n$ *over* $\{0,1\}^{m(n)}$ *such that for every poly-size distinguisher* $D = \{D_n\}_{n\in\mathbb{N}}$*, and for every large enough* n*,*

$$\left|\mathbf{Pr}\left[D(\mathbf{U}_{\ell(n)}) = 1\right] - \mathbf{Pr}\left[D(G(\mathcal{D}_n)) = 1\right]\right| \leq \epsilon(n).$$

[LMP26] prove the following theorem.

Theorem 3.2 ([LMP26])**.** *Assume the existence of a* $1/10$*-conditional PRG* $G\colon \{0,1\}^{d(n)} \to \{0,1\}^{d(n)+\log n}$ *computable in time* $t(d(n) + \log n)$*. Then* $(\mathrm{MK}^{\mathrm{t}}\mathrm{P}[n-2], \mathbf{U}_n) \notin \mathsf{AvgBPP}/\mathrm{poly}$*.*

In the rest of this section we construct a conditional PRG. For this we introduce the notion of next-bit unpredictability.

3.2 Next-Bit Unpredictability

In the following we construct a conditional PRG by first constructing unpredictable bits generator. We adjust the definitions from [MP23] to the case in which the input distribution is not uniform. Recall that we identify a matrix $M \in \{0,1\}^{n\times m}$ with a function $M\colon \{0,1\}^n \to \{0,1\}^m$ by $M(x) := x \cdot M \bmod 2$, thinking of $x \in \{0,1\}^n$ as a vector with dimension n.

Definition 3.3 (Unpredictable bits)**.** *Let* $m = m(n)$*,* $\ell = \ell(n)$*,* $\lambda = \lambda(n)$ *and* $k = k(n)$ *be integer functions, and let* $\epsilon = \epsilon(n) \in [0,1]$*. We say that a function family* $g = \{g_a\colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a\in\{0,1\}^{\lambda(n)}}$ *has* (k,ϵ)-bits-unpredictability *with respect to a distribution* $\mathcal{D} = \{\mathcal{D}_n \subseteq \{0,1\}^{m(n)}\}_{n\in\mathbb{N}}$ *if for every* $n \in \mathbb{N}$ *and* $x \in \{0,1\}^{m(n)}$*, there exists a set* $\mathcal{S}(x) \subseteq [\ell(n)]$*, such that, for* $X_n \leftarrow \mathcal{D}_n$ *and* $A \leftarrow \{0,1\}^{\lambda(n)}$*:*

1. *For every* n*,* $\mathbf{E}\left[|\mathcal{S}(X_n)|\right] \geq k(n)$*, and,*
2. *for every sequence* $\{i_n\}_{n\in\mathbb{N}}$ *such that* $i_n \in \bigcup_{x\in\{0,1\}^{m(n)}} \mathcal{S}(x)$*,*

$$\{(A, g_A(X_n)_{<i_n}, g_A(X_n)_{i_n})|_{i_n\in\mathcal{S}(X_n)}\}_{n\in\mathbb{N}} \overset{c}{\approx}_\epsilon \{(A, g_A(X_n)_{<i_n}, \mathbf{U})|_{i_n\in\mathcal{S}(X_n)}\}_{n\in\mathbb{N}}.$$

We say that g *has* k-bits-unpredictability *if it has* (k, n^{-c})*-bits-unpredictability for every* $c \in \mathbb{N}$*.*

[10] Although we don't use it in our work, it can be assumed without lose of generality that the runtime of an unbounded simulator is at most $2^{(|w_n|)} \cdot \mathrm{poly}(n)$.

As in [MP23], we will also consider a stronger notion of unpredictability—called *k-random-bit unpredictability*, that requires each individual bit to be unpredictable with probability k/ℓ where ℓ is the output length.

Definition 3.4 (Random bits unpredictability)**.** *Let $m = m(n)$, $\ell = \ell(n)$ and $k = k(n)$ be integer functions, and let $\epsilon = \epsilon(n) \in [0,1]$. We say that a function family $g = \{g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a \in \{0,1\}^{\lambda(n)}}$ has (k,ϵ)-*random-bits-unpredictability *with respect to $\mathcal{D}$ if it satisfies Definition 3.3 except that condition (1) is replaced by:*

1. *For every $i \in [\ell(n)]$, $\mathbf{Pr}\left[i \in \mathcal{S}(X_n)\right] \geq k(n)/\ell(n)$.*

*We say that g has k-*random-bits-unpredictability *if it has (k, n^{-c})-random-bits-unpredictability for every $c \in \mathbb{N}$.*

We will need the following theorem, proven in Section 6.

Theorem 3.5. *Assume there is an efficiently computable function family $\{f_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a \in \{0,1\}^{\lambda(n)}}$ and a distribution ensemble $\mathcal{D}$, with respect to f has $k(n)$-random unpredictable bits, for $k(n) \geq \mathrm{H}(f(\mathcal{D}_n)) + (m(n) - \mathrm{H}(\mathcal{D}_n)) + 1$. Then there exists an ϵ-conditional PRG $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{d(n)+\log n}$ for a negligible function ϵ.*

4 Simulatable Two-Message Arguments to Next-Bit Unpredictability

In this section we consider *public-coin* two-message protocol with honest-verifier inefficient simulator. Recall that $\mathrm{MK^tP}[s] = \{x \colon \mathrm{K}^t(x) \leq s(|x|)\}$. We prove the following theorem.

Theorem 4.1. *Assume a language $L \notin \mathsf{ioSIZE}[T_L]$ has a two-message, public-coin argument (P,V) with negligible completeness and soundness errors, and an external-view simulator* Sim *computable by a circuit of size $T_S(n) < T_L(n) \cdot negl(n)$. Then for some $t \in \mathrm{poly}$ $(\mathrm{MK^tP}[n-2], \mathbf{U}_n) \notin \mathsf{AvgBPP}$.*

Theorem 4.1 is a direct corollary of the following lemma, which is our main result in this section.

Lemma 4.2. *Assume a language $L \notin \mathsf{ioSIZE}[T_L]$ has a two-message, public-coin argument (P,V) with negligible completeness and soundness errors, and an external-view simulator* Sim *computable by a circuit of size $T_S(n) < T_L(n) \cdot negl(n)$. Then there is an efficiently computable function family $\{f_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a \in \{0,1\}^{\lambda(n)}}$ and a distribution ensemble $\mathcal{D}$, with respect to f has $k(n)$-random unpredictable bits, for $k(n) \geq \mathrm{H}(f(\mathcal{D}_n)) + (m(n) - \mathrm{H}(\mathcal{D}_n)) + 1$*

Proof of Theorem 4.1. Immediate from Lemma 4.2 and Theorems 3.2 and 3.5. □

The rest of this section is dedicated to prove Lemma 4.2. Let (P,V) be a two-message argument system with inefficient external-view simulator Sim.

We refer the message of the verifier as a crs. Assume the crs is of length $\ell(n)$, and that the prover uses randomness of length $p(n)/2$, and generates proofs of length $p(n)$. Let $\mathcal{D}^Y, \mathcal{D}^N$ be the distribution ensembles promised by Lemma 2.4, with $\epsilon(n) = (T_S \cdot n / T_L)^{1/4} = \mathsf{negl}(n)$. It follows by Lemma 2.4 that circuits of size $T_S \cdot \mathrm{poly}(n)$ cannot distinguish between $\mathcal{D}^Y$ and $\mathcal{D}^N$ with a noticeable advantage. In the following we assume that $\mathcal{D}^Y$ is a distribution over pairs $(x,w) \in \mathcal{R}_L$ (note that we do not assume $\mathcal{D}^Y$ is efficiently samplable). Our goal is to show that for large enough $t(n) \in \mathrm{poly}$ the function

$$f_A(x, w, crs_1, \ldots, crs_t, r_1, \ldots, r_t, j) = [(crs_1, A(P_{x,w}(crs_1; r_1)), \ldots, (crs_t, A(P_{x,w}(crs_t; r_t))]_{[j:(t-1)(\ell+p)+j-1]}$$

has sufficient amount of random bits unpredictability, when (x,w) is generated from the distribution $\mathcal{D}^Y_n$, $j \leftarrow [\ell(n) + p(n)]$, and $crs_1, \ldots, crs_t, r_1, \ldots, r_t$ are chosen uniformly at random, and $A \in \{0,1\}^{p(n) \times p(n)}$.

We start by showing that

$$G_A(x, w, crs_1, \ldots, crs_t, r_1, \ldots, r_t) = (crs_1, A(P_{x,w}(crs_1; r_1)), \ldots, (crs_t, A(P_{x,w}(crs_t; r_t))$$

has sufficient amount of bits unpredictability, and more over, in each block of length $\ell(n)+p(n)$ the expected number of unpredictable bits is large. For this, we define

$$\gamma(x,w,crs,r) = \lceil -\log \mathbf{Pr}\left[P_{x,w}(crs) = P_{x,w}(crs;r)\right]\rceil,$$

(note that $\gamma(x,w,crs,r) \leq |r| = p(n)/2$ by definition). For a fixed ϵ, let

$$\widehat{\gamma}(x,w,crs,r) = \min(\gamma(x,w,crs,r), T),$$

where $T \in [p(n)]$ is the largest such that $\mathbf{Pr}\left[\gamma(x,w,crs,r) \geq T\right] \geq \epsilon/(p(n)t^2)$ for $(x,w) \leftarrow \mathcal{D}_n^Y$, and random crs, r. Observe that

$$\mathbf{Pr}\left[\widehat{\gamma}(x,w,crs,r) \neq \gamma(x,w,crs,r)\right] \leq \epsilon/p(n)t^2.$$

Finally, let

$$\mathcal{S}_\epsilon(x,w,crs_1,\ldots,crs_t,r_1,\ldots,r_t) = \bigcup_{i\in[t(n)]} \{j+(i-1)(\ell(n)+p(n)) \colon j \leq \ell(n)+\widehat{\gamma}(x,w,crs_i,r_i)+2\log n\},$$

and

$$\mathcal{S}(x,w,crs_1,\ldots,crs_t,r_1,\ldots,r_t) = \bigcup_{i\in[t(n)]} \{j+(i-1)(\ell(n)+p(n)) \colon j \leq \ell(n)+\gamma(x,w,crs_i,r_i)+2\log n\},$$

In the following, $\mathcal{S}_\epsilon$ will be the set with respect to G_A have bits-unpredictability, but it will be easier to analyze $\mathcal{S}$.

Fix $n \in \mathbb{N}$, and consider the following random variables. Let $(X,W) \leftarrow \mathcal{D}_n^Y$, $CRS \leftarrow \mathbf{U}_{\ell(n)}$ and $P_{X,W}(CRS)$ be a random proof of P generated with uniformly random randomness. Let $Z \leftarrow \mathbf{U}_{t(n)(\ell(n)+p(n))}$. We prove the following claims. The first shows that $\mathcal{S}_\epsilon$ is large.

Claim 4.3. *For each $i \in [t(n)]$,*

$$\mathbf{E}\left[\mathcal{S}_\epsilon(X,W,Z) \cap [(i-1)(\ell(n)+p(n))+1, i(\ell(n)+p(n))]\right] \geq \ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + \log n.$$

Proof. This is easy to see for the set $\mathcal{S}$ by the definition of $\mathcal{S}$ and γ. Since $\mathbf{Pr}\left[\mathcal{S} \neq \mathcal{S}_\epsilon\right] \leq \epsilon/p(n)t$, and since $|\mathcal{S}| - |\mathcal{S}_\epsilon| < tp(n)$, we get a similar bound on $\mathcal{S}_\epsilon$. □

The second claim shows that G_A indeed fulfills the bits-unpredictability requirement with respect to the set $\mathcal{S}_\epsilon$.

Claim 4.4. *Let $G_A, \mathcal{S}_\epsilon$ be as defined above. Then G_A fulfills the definition of ϵ-bits unpredictability with respect to the distribution $(\mathcal{D}^Y, \mathbf{U}_{t(n)(\ell(n)+p(n))})$ and the set $\mathcal{S}_\epsilon$.*

We prove Claim 4.4 in Section 4.1, but first, let us use it to prove Lemma 4.2.

Proof of Lemma 4.2. Let $t(n) = |x,w| + \ell(n) + p(n)\log(p(n)+\ell(n))$, and let G_A, f_A be the functions defined above. Fix ϵ, and define

$$\begin{aligned}&\mathcal{S}'_\epsilon(x,w,crs_1,\ldots,crs_t,r_1,\ldots,r_t,j)\\ &= \{v-j+1 \colon v \in \mathcal{S}_\epsilon(x,w,crs_1,\ldots,crs_t,r_1,\ldots,r_t), j \leq v \leq (t(n)-1)(\ell(n)+p(n))+j-1\}.\end{aligned}$$

By the definition of f_A and by data processing, it is clear that f_A fulfills the definition of bits unpredictability with respect to the distribution $(\mathcal{D}^Y, \mathbf{U}_{t(n)(\ell(n)+p(n))})$ and the set $\mathcal{S}'_\epsilon$. It is thus enough to show that for any $v' \in [t(n)(\ell(n)+p(n))]$ it holds that for $J \leftarrow [\ell(n)+p(n)]$

$$\mathbf{Pr}\left[v' \in \mathcal{S}'_\epsilon(X,W,Z,J)\right] \geq \frac{H(f_A(X,W,Z,J)) + |X,W,Z,J| - H(X,W,Z,J) + 1}{|f_A(X,W,Z,J)|}$$

where we have

$$\begin{aligned}
&\frac{H(f_A(X,W,Z,J)) + |X,W,Z,J| - H(X,W,Z,J) + 1}{|f_A(X,W,Z,J)|} \\
&\leq \frac{H(f_A(X,W,Z,J)) + |x,w| + 1}{|f_A(X,W,Z,J)|} \\
&\leq \frac{t(n)\mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + t(n)\ell(n) + |J| + |x,w| + 1}{(t(n)-1)(\ell(n)+p(n))} \\
&= \frac{\ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS)}{(\ell(n)+p(n))} \\
&\quad + \frac{\ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + |J| + |x,w| + 1}{(t(n)-1)(\ell(n)+p(n))} \\
&\leq \frac{\ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS)}{\ell(n)+p(n)} + \frac{(t(n)-1)\log n}{(t(n)-1)(\ell(n)+p(n))} \qquad (*) \\
&\leq \frac{\ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + \log n}{\ell(n)+p(n)}
\end{aligned}$$

where the last inequality holds since by our choice of t

$$\mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + |J| + |x,w| + 1 \leq \ell(n) + p(\ell) + |J| + |x,w| + 1 \leq (t(n)-1)\log n.$$

Finally, observe that for every v', the distribution $v' + J \bmod \ell(n)+p(n)$ is uniform. By the definition of $\mathcal{S}'$, we get that

$$\begin{aligned}
\mathbf{Pr}\left[v' \in \mathcal{S}'_\epsilon(X,W,Z,J)\right] &= \mathbf{Pr}\left[v' + J \in \mathcal{S}_\epsilon(X,W,Z,J)\right] \\
&\geq \frac{\min_i\{\mathbf{E}\left[\mathcal{S}_\epsilon(X,W,Z) \cap [i(\ell(n)+p(n))+1, (i+1)(\ell(n)+p(n))]\right]\}}{(\ell(n)+p(n))} \\
&\geq \frac{\ell(n) + \mathrm{H}(P_{X,W}(CRS) \mid X,W,CRS) + \log n}{\ell(n)+p(n)}
\end{aligned}$$

where the last inequality holds by Claim 4.3. □

4.1 Proving Claim 4.4

In this section we prove Claim 4.4. In the following, for every $i \in \mathbb{N}$, let $Q^i_{x,w}$ be a random variable distributed according to

$$((crs_1, P_{x,w}(crs_1)), \ldots ((crs_i, P_{x,w}(crs_i))$$

where $crs_1, \ldots crs_i$ are uniformly distributed. Let $A_n \leftarrow \{0,1\}^{p(n)\times p(n)}$ be a random matrix, and for A_n, crs let $A_n(P_{x,w}(crs))_{\leq \gamma+2\log n}$ be the random variable distributed according to

$$A(P_{x,w}(crs); R)_{\leq \gamma(x,w,crs,R)+2\log n},$$

for uniformly distributed R. We prove the following lemma.

Lemma 4.5. *For every polynomial-size distinguisher D, and for every $i = i_n \leq t(n)$*

$$\begin{aligned}
\Big| &\mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, Q^{i-1}_{x,w}, (crs_i, A(P_{x,w}(crs_i))_{\leq\gamma+2\log n})) = 1\right] \\
&- \mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, Q^{i-1}_{x,w}, (crs_i, \mathbf{U}_{\leq\gamma+2\log n})) = 1\right]\Big| = \mathsf{negl}(n).
\end{aligned}$$

Claim 4.4 follows immediately by Lemma 4.5.

Proof of Claim 4.4. Let $\widehat{Q}^i_{x,w}$ be a random variable distributed according to

$$((crs_1, A_n(P_{x,w}(crs_1))), \ldots ((crs_i, A_n(P_{x,w}(crs_i))).$$

By data processing, Lemma 4.5 implies that

$$\left| \Pr_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, \widehat{Q}^{i-1}_{x,w}, (crs_i, A(P_{x,w}(crs_i))_{\leq\gamma+2\log n})) = 1\right]\right.$$
$$\left. - \Pr_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, \widehat{Q}^{i-1}_{x,w}, (crs_i, \mathbf{U}_{\leq\gamma+2\log n})) = 1\right]\right| = \mathsf{negl}(n).$$

Claim 4.4 now follows by the definition of $\mathcal{S}$ and Lemma 4.5, together with the fact that crs_i is always random, and the observation that, for any $i \in \mathcal{S}_\epsilon$, the conditioning that $i \in \mathcal{S}_\epsilon$ happens with probability at least 1/poly by definition, and thus adding this conditioning does not change the distinguishing advantage to noticeable (observe that D implicitly gets γ and thus the conditioning event is recognizable by D). □

4.2 Proving Lemma 4.5

Finally, we prove Lemma 4.5.

Proof of Lemma 4.5. Assume towards a contradiction that Lemma 4.5 does not hold, and let D be a distinguisher such that

$$\left| \Pr_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, \widehat{Q}^{i-1}_{x,w}, (crs_i, A(P_{x,w}(crs_i))_{\leq\gamma+2\log n})) = 1\right]\right.$$
$$\left. - \Pr_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell}\left[D(x, A_n, \widehat{Q}^{i-1}_{x,w}, (crs_i, \mathbf{U}_{\leq\gamma+2\log n})) = 1\right]\right| \geq 1/n^c$$

For some constant $c \in \mathbb{N}$ and for infinitely many $n \in \mathbb{N}$. Fix such $n \in \mathbb{N}$.

Step I: Using D to generate accepting proofs. We start by constructing from D an algorithm $\widehat{P}$ that given x and sequence of valid proofs, output an accepting proof for a fresh CRS with noticeable probability.

Using D and by Lemma 2.11 (applied to the value of γ that maximizes the above distinguishing advantage), we get a polynomial-size algorithm Pred, that given

$$(x, h, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1})), (crs_i, A_n(P_{x,w}(crs_i))_{\leq\gamma+2\log n}))$$

and with probability $\epsilon(n) = \mathrm{poly}(1/n^c, 1/r(n))$, outputs (exactly) the proof $P_{x,w}(crs_i)$ outputted by the prover. For a fixed $z = (x, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1})), crs_i)$ and $\pi = P_{w,x}(crs_i)$, let ϵ^z_π be the probability that Pred outputs π on input $(z, A_n(\pi)_{\gamma+2\log n})$. We have that $\mathbf{E}_{z,\pi}[\epsilon^z_\pi] = \epsilon(n)$.

Let $\widehat{P}$ be the algorithm that given

$$(x, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1})), crs_i)$$

guesses the value of $\gamma' \leftarrow [p(n)]$, and executes Pred on

$$(x, A_n, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1})), (crs_i, \mathbf{U}_{\leq\gamma'+2\log n})).$$

Fix $\pi = P_{w,x}(crs)$, and $\gamma = \gamma(x, w, crs, \pi)$. Then with probability at least $2^{-\gamma(x,w,crs)-2\log n}/2p(n) = \mathbf{Pr}[P_{x,w}(crs) = \pi]/(2n^2p(n))$, $\widehat{P}$ guesses correctly the value of $\gamma, h(\pi)_{\leq\gamma(x,w,crs)+2\log n}$. We conclude that $\widehat{P}$ outputs π on input z with probability $\mathbf{Pr}[P_{x,w}(crs) = \pi]/(2n^2p(n)) \cdot \epsilon^z_\pi$. Thus, we can bound the probability that $\widehat{P}$ outputs some convincing proof with

$$\sum_{\pi\in\mathsf{GOOD}} \mathbf{Pr}[P_{x,w}(crs) = \pi]/(2n^2p(n)) \cdot \epsilon^z_\pi$$

where GOOD is the set of prover messages on which the verifier accepts with probability at least 0.9. By the completeness of the protocol, $\mathbf{Pr}[P_{x,w}(crs) \in \mathsf{GOOD}] \geq 1 - \mathsf{negl}(n)$. We conclude that the probability that $\widehat{P}$ outputs some convincing proof is

$$\sum_{\pi\in\mathsf{GOOD}} \mathbf{Pr}[P_{x,w}(crs) = \pi]/(2n^2p(n)) \cdot \epsilon^z_\pi \geq \sum_\pi \mathbf{Pr}[P_{x,w}(crs) = \pi]/(2n^2p(n)) \cdot \epsilon^z_\pi - \mathsf{negl}(n)$$
$$= 1/(2n^2p(n)) \mathop{\mathbf{E}}_\pi[\epsilon^z_\pi] - \mathsf{negl}(n).$$

taking expectation over z we get that $\widehat{P}$ outputs a convincing proof with probability at least $\epsilon'(n) = \epsilon(n)/2n^2p(n) - \mathsf{negl}(n)$.

Step II: Replacing the valid proofs with simulated proofs. Our goal is to show that $\widehat{P}$ can be used to produce an accepting proof for a No instance. Toward this we first want to show that $\widehat{P}$ succeed in outputting a convincing proof even if the input it gets is sampled using the simulator instead of the random prover.

For this, let D'' be the algorithm that on input,

$$(x, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1})))$$

samples a uniformly random crs_i, and then simulates $\widehat{P}$ to get a proof π and checks if $V(x, crs_i, \pi)$ accepts. Then we have that D'' outputs 1 with the same probability that $\widehat{P}$ outputs a valid proof. We next claim that D'' does not notice if we replace the proofs generated by the honest prover by simulated proofs. That is,

$$\begin{aligned} \Big| \mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell} [D''(x, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1}))) = 1] \\ - \mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, (crs_j,\pi_j)\leftarrow\mathrm{Sim}(x), crs_i\leftarrow\mathbf{U}_\ell} [D''(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1})) = 1] \Big| \leq \epsilon'(n)/4. \end{aligned} \tag{3}$$

Indeed, assume Eq. (3) does not hold, and fix x, w that maximize this distinguishing advantage. By a simple hybrid argument we can construct a polynomial size circuit that distinguishes $\mathrm{Sim}(x)$ from a random proof with advantage $\epsilon'(n)/i = 1/\mathrm{poly}(n)$, which is a contradiction to the security of the simulation.

Step III: Replacing x with No-instance. Lastly, we replace x with a No-instance. We claim that

$$\begin{aligned} \Big| \mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, (crs_i,\pi_i)\leftarrow\mathrm{Sim}(x)} [D''(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1})) = 1] \\ - \mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^N, (crs_i,\pi_i)\leftarrow\mathrm{Sim}(x)} [D''(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1}))) = 1] \Big| \leq \epsilon'(n)/4. \end{aligned} \tag{4}$$

Indeed, assume towards a contradiction this is not the case. So we can distinguish between $\mathcal{D}^Y$ to $\mathcal{D}^N$ with circuit of size $T_S \cdot i + \mathrm{poly}(n) < T_S \cdot \mathrm{poly}(n)$ by applying the simulator on x to get $((crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1}))$, and then executing D''. This is a contradiction to the hardness of $(\mathcal{D}^Y, \mathcal{D}^N)$.

By Eqs. (3) and (4) we conclude that

$$\begin{aligned} \Big| \mathop{\mathbf{Pr}}_{(x,w)\leftarrow\mathcal{D}^Y, crs_i\leftarrow\mathbf{U}_\ell} [D''(x, (crs_1, P_{x,w}(crs_1)), \ldots, (crs_{i-1}, P_{x,w}(crs_{i-1}))) = 1] \\ - \mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^N, (crs_i,\pi_i)\leftarrow\mathrm{Sim}(x)} [D''(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1}))) = 1] \Big| \leq \epsilon'(n)/2, \end{aligned} \tag{5}$$

which means that

$$\mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^N, (crs_j,\pi_j)\leftarrow\mathrm{Sim}(x), crs\leftarrow\{0,1\}^{\ell(n)}} \Big[V(x, crs, \widehat{P}(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1}), crs)) = 1 \Big] \geq \epsilon'(n)/2.$$

This is a contradiction to the soundness of the proof, as by averaging argument, there is a fixing of $x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1})$ such that $P'_x(crs) = \widehat{P}(x, (crs_1, \pi_1), \ldots, (crs_{i-1}, \pi_{i-1}), crs)$ outputs a convincing proof with non-negligible probability, where x is a No-instance for L. Moreover, P'_x is of polynomial size. □

5 Dealing with Private Coins

We next generalize Theorem 4.1 to hold for private coin protocols. We prove the following theorem.

Theorem 5.1. *Assume a language $L \notin \mathsf{ioSIZE}[T_L]$ has a two-message argument (P, V) with negligible completeness and soundness errors and a simulator* Sim *computable by a circuit of size $T_S(n) < T_L(n) \cdot negl(n)$. Then for some $t \in \mathrm{poly}$ $(\mathrm{MK^tP}[n-2], \mathbf{U}_n) \notin \mathsf{AvgBPP}$.*

The proof is by the observation that we can assume without loss of generality the protocol is public-coins, as otherwise there exists an ioAI-OWF. In the following we write $V = (V_1, V_2)$, where $V_1(x; r)$ is the algorithm generating the first message of the protocol, and $V_2(x, m; r)$ the algorithm deciding if to accept or reject the proof m generated by the prover. In the following, let $C(\text{Sim})$ be the circuit size needed to compute Sim.

Lemma 5.2. *Assume that ioAI-OWF do not exist. Let (P, V) be a two-message argument with negligible completeness and soundness error and* Sim *be its inefficient external-view simulator. Then there exist efficient algorithms P', V_1' such that for every $\epsilon \in 1/\text{poly}$ there exist a simulator* Sim' *computable with a circuit of size $C(\text{Sim}') \leq C(\text{Sim}) + \text{poly}$, and V_2', such that $(P', V_\epsilon' = (V_1', V_2'))$ is a public-coin, two-message argument with completeness and soundness error ϵ and external-view ϵ-simulator* Sim'.

Proof. Let $f = \{f_x\}_{x \in \{0,1\}^n}$ be the function family defined by

$$f_x(r) = V_1(x; r).$$

That is, f_x is the verifier first message function. In the following we assume f is not infinitely-often auxiliary-input distributional one-way function. That is, for every $\epsilon \in 1/\text{poly}$, there exists an efficient algorithm Inv, such that

$$\text{SD}((R, f_x(R)), (\mathsf{Inv}(f_x(R)), f_x(R))) \leq \epsilon(n),$$

where R is chosen uniformly at random.

We change (P, V) as follows. V_1' on input x and randomness r will simply send r to the prover. P' on input (x, w) and the message r, will compute $\rho = V(x; r)$, and will send $m = P(x, w, \rho)$ to the verifier. V_2' will then compute $r' = \mathsf{Inv}(f_x(r))$, and will accept if V accepts on message m and randomness r'.

Before proving soundness and completeness, we construct the simulator Sim'. To simulate the above protocol, we simply execute Sim to get $(\rho, m) \leftarrow \text{Sim}(x)$, and output $(\rho' = \mathsf{Inv}(\rho), m)$. It is not hard to see by data processing that a distinguisher distinguish between Sim' to the interaction between (P', V') can be used to distinguish between the interaction of (P, V) and Sim.

We move to analyze completeness. We notice that if Inv is a perfect inverter, that is if $(R, f_x(R)) \equiv (\mathsf{Inv}(f_x(R)), f_x(R))$, then the completeness is exactly the same as in the original protocol. Thus, by the small statistical distance, the completeness error is at most ϵ.

For soundness, let $\widehat{P}'$ be a cheating prover, that on input $x \notin \mathcal{L}$ convinces the verifier V' to accept with probability 3ϵ. We construct a prover $\widehat{P}$ that convinces V to accept with probability ϵ.

$\widehat{P}$ on input x and message ρ from the verifier, will simply compute $r = \mathsf{Inv}(\rho)$, and will execute $\widehat{P}'(x, r)$ to get a message m to send to the verifier. Again, we notice that when Inv is perfect, $\widehat{P}$ convinces V with the same probability $\widehat{P}'$ convinces V'. Since the statistical distance is at most ϵ, by a simple hybrid argument the probability to convince the verifier V is at least ϵ (where we need to account both for the error of $\widehat{P}$ to invert f_x, and the error of V'. □

We also use the following fact, followed by the fact that we can use a ioAI-OWF to construct ioAI-PRG.

Lemma 5.3 ([LMP24])**.** *Assume that ioAI-OWF exists. Then* $\text{MK}^{\text{t}}\text{P}[n-2] \notin \mathsf{AvgBPP}$.

We finally prove Theorem 5.1.

Proof of Theorem 5.1. By Lemmas 5.2 and 5.3, we can assume without loss of generality that the protocol is public-coins, and instead of negligible soundness error, it has the structure described in Lemma 5.2. The proof now follows by the same proof of Theorem 4.1, with the observation that we can choose ε to be small enough without changing the distribution of the transcript of the protocol, which is enough for the proof of Theorem 4.1. □

6 Unpredictability to Pseudorandomness: Proving Theorem 3.5

In this section we prove Theorem 3.5. We will make use of the following theorem. In [MP23], the following theorem is proven for the case $\mathcal{D}_n$ is the uniform distribution, However, same proof yields the following general version.

Theorem 6.1 (Extracting from random bits unpredictability, [MP23])**.** *Let $m = m(n)$, $\ell = \ell(n)$, $\lambda = \lambda(n)$ and $k = k(n)$ be integer functions, and let $g = \{g_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a \in \{0,1\}^{\lambda(n)}}$ be a function family with $k(n)$-random-bits-unpredictability with respect to $\mathcal{D}$. Then the following holds for every polynomial $t = t(n)$. Let $\alpha(n) = k(n)/\ell(n)$, and let $H_n \leftarrow \{0,1\}^{t(n) \times q(n)}$ be a random matrix, for $q = \lfloor \alpha t - \sqrt{\alpha t \cdot \log^2 n} - \log^2 n \rfloor$. Then for $X_n^1 \ldots, X_n^{t(n)} \leftarrow (\mathcal{D}_n)^{t(n)}$ and $A_n \leftarrow \{0,1\}^{\lambda(n)}$, the distribution ensemble*

$$\{H_n, A_n, H_n(g_{A_n}(X_n^1)_1, \ldots, g_{A_n}(X_n^{t(n)})_1), \ldots, H_n(g_{A_n}(X_n^1)_{\ell(n)}, \ldots, g_{A_n}(X_n^{t(n)})_{\ell(n)})\}_{n \in \mathbb{N}}$$

is pseudorandom.

We will use the following two lemmas, that shows it is enough to construct a function with pseudorandom output, where the output is longer than its real entropy.

Lemma 6.2. *Let $G \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ be an efficiently computable function, and let X_n be a random variable distributed over $\{0,1\}^{m(n)}$. Assume that $\{G(X_n)\}_{n \in \mathbb{N}}$ is pseudorandom, and that $\ell(n) - \mathrm{H}(G(X_n)) > m(n) - \mathrm{H}(X_n) + 1$. Then there exists a conditional PRG $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{d(n)+\log n}$.*

Lemma 6.3. *Assume there is an efficiently computable function family $\{f_a \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}\}_{a \in \{0,1\}^{\lambda(n)}}$ and a distribution ensemble $\mathcal{D}$, with respect to f has $k(n)$-random unpredictable bits, for $k(n) \geq \mathrm{H}(f(\mathcal{D}_n)) + (m(n) - \mathrm{H}(\mathcal{D}_n)) + 1$. Then there exists a efficiently computable function $G \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ and a random variable X_n distributed over $\{0,1\}^{m(n)}$, such that $\{G(X_n)\}_{n \in \mathbb{N}}$ is pseudorandom, and that $\ell(n) - \mathrm{H}(G(X_n)) > m(n) - \mathrm{H}(X_n) + 1$.*

Lemmas 6.2 and 6.3 together imply Theorem 3.5.

Proof of Theorem 3.5. Immediate by Lemmas 6.2 and 6.3. □

We prove Lemma 6.2 in Section 6.1 and Lemma 6.3 in Section 6.2.

6.1 Proving Lemma 6.2

Lemma 6.4 (Lemma 6.2,restated)**.** *Let $G \colon \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ be an efficiently computable function, and let X_n be a random variable distributed over $\{0,1\}^{m(n)}$. Assume that $\{G(X_n)\}_{n \in \mathbb{N}}$ is pseudorandom, and that $\ell(n) - \mathrm{H}(G(X_n)) > m(n) - \mathrm{H}(X_n) + 1$. Then there exists a conditional PRG $G \colon \{0,1\}^{d(n)} \to \{0,1\}^{d(n)+\log n}$.*

In the following, recall that we identify a matrix $M \in \{0,1\}^{n \times m}$ with a function $M \colon \{0,1\}^n \to \{0,1\}^m$ by $M(x) := x \cdot M \bmod 2$, thinking of $x \in \{0,1\}^n$ as a vector with dimension n.

Proof of Lemma 6.2. First, observe that if $\ell(n) > m(n)$, then we are done by taking $\log n$ repetitions. Thus, we assume that $\ell(n) \leq m(n)$. Let $t(n) = (m(n))^4$, and $A_n \colon \{0,1\}^{t(n) \cdot m(n)} \to \{0,1\}^{t(n)(m(n)-\ell(n))+\log n}$ be a random matrix. Let

$$G^{t(n)}(A, x^1, \ldots, x^{t(n)}) = A, G(x^1), \ldots, G(x^{t(n)}), A(x^1, \ldots, x^{t(n)}).$$

Let $r(n) = t(n)(m(n)-\ell(n))+1$, and let $(X_n^1, \ldots, X_n^{t(n)})$ be $t(n)$ random variables distributed according to $(X_n)^{t(n)}$. In the following we want to show that $G^{t(n)}$ is a conditional PRG. By our choice of parameters, $G^{t(n)}$ is expanding, so it is enough to prove that its output is pseudorandom.

Pseudorandomness. First, by assumption, the distribution

$$A, (G(X_n^1), \ldots, G(X_n^{t(n)}))$$

is pseudorandom. Thus, it is enough to prove that

$$A(X^1, \ldots, X^{t(n)})$$

is pseudorandom given $A, (G(X_n^1), \ldots, G(X_n^{t(n)}))$.

In the following, we will show that $(G(X_n^1),\ldots,G(X_n^{t(n)})),(X_n^1,\ldots,X_n^{t(n)})$ is statistically close to a random distribution $(G(X_n^1),\ldots,G(X_n^{t(n)})),(\widehat{X}_n^1,\ldots,\widehat{X}_n^{t(n)})$ with

$$\mathrm{H}_\infty((\widehat{X}_n^1,\ldots,\widehat{X}_n^{t(n)}) \mid (G(X_n^1),\ldots,G(X_n^{t(n)})) = (y_1,\ldots,y_{t(n)})) \geq r(n)+\omega(\log n)$$

for any $(y_1,\ldots,y_{t(n)})$. By the left-over hash lemma (Lemma 2.9), this implies that $A(X_n^1,\ldots,X_n^{t(n)})$ is statistically close to the uniform distribution given the value of $(A,G(X_n^1),\ldots,G(X_n^{t(n)}))$, which concludes the proof.

Bounding the min-entropy. It is enough to show that

$$\mathop{\mathbf{Pr}}_{\substack{x_1,\ldots,x_{t(n)}\leftarrow(X_n)^{t(n)}\\ y_1,\ldots,y_{t(n)}=G(x_1),\ldots,G(x_{t(n)})}}\left[\mathop{\mathbf{Pr}}_{\geq 2^{-r(n)-\omega(\log n)}}[(X_n^1,\ldots,X_n^{t(n)})=(x_1,\ldots,x_{t(n)})\mid(G(X_n^1),\ldots,G(X_n^{t(n)}))=(y_1,\ldots,y_{t(n)})]\right]\leq \mathsf{negl}(n). \quad (6)$$

Indeed, Eq. (6) implies that $(G(X_n^1),\ldots,G(X_n^{t(n)})),(X_n^1,\ldots,X_n^{t(n)})$ is statistically close to a random distribution $(G(X_n^1),\ldots,G(X_n^{t(n)})),(\widehat{X}_n^1,\ldots,\widehat{X}_n^{t(n)})$ with

$$\mathrm{H}_\infty((\widehat{X}_n^1,\ldots,\widehat{X}_n^{t(n)}) \mid (G(X_n^1),\ldots,G(X_n^{t(n)})) = (y_1,\ldots,y_{t(n)})) \geq r(n)+\omega(\log n).$$

Proving Eq. (6). We move to show that Eq. (6) holds. This follows standard consternation of the probabilities to sample random $X^1,\ldots,X^n$:

By the assumption on $\ell(n)$ and $m(n)$ we have that

$$r(n)=t(n)(m(n)-\ell(n))\leq t(n)(\mathrm{H}(X_n)-\mathrm{H}(G(X_n))-1)\leq t(n)\mathrm{H}(X_n\mid G(X_n))-t(n),$$

so we want to show that

$$\mathop{\mathbf{Pr}}_{\substack{x_1,\ldots,x_n\leftarrow(X_n)^{t(n)}\\ y_1,\ldots,y_t=G(x_1),\ldots,G(x_t)}}\left[\mathbf{Pr}\left[(X_n^1,\ldots,X_n^t)=(x_1,\ldots,x_t)\mid(G(X_n^1),\ldots,G(X_n^t))=(y_1,\ldots,y_t)\right]\geq 2^{-r'(n)}\right]\leq\mathsf{negl}(n)$$

for $r'(n)=t(n)\mathrm{H}(X_n\mid G(X_n))-t(n)/2$. For each $i\in[t(n)]$, let L_i be a random variable jointly distributed with X_n^i, taking the value $L_i=-\log\mathbf{Pr}_{x\leftarrow X_n}\left[x=X_n^i\mid G(x)=G(X_n^i)\right]$. Then by definition of entropy, $\mathbf{E}\left[L_i\right]=\mathrm{H}(X_n\mid G(X_n))\leq m(n)$. Let $\widetilde{L_i}=\min(L_i,2m(n))$, and observe that $L_i\geq\widetilde{L_i}$. Moreover,

$$\mathbf{E}\left[L_i\right]-\mathbf{E}\left[\widetilde{L_i}\right]\leq\sum_{\ell>2m(n)}(\ell+1)\,\mathbf{Pr}\left[L_i\geq\ell\right]=\sum_{\ell>2m(n)}(\ell+1)\mathop{\mathbf{Pr}}_{X_n^i}\left[\mathop{\mathbf{Pr}}_{x\leftarrow X_n}\left[x=X_n^i\mid G(x)=G(X_n^i)\right]\leq 2^{-\ell}\right]$$

$$\leq\sum_{\ell>2m(n)}(\ell+1)2^{-\ell}\cdot 2^{m(n)}=\sum_{\ell>0}(2m(n)+\ell+1)2^{-m(n)-\ell}=\mathsf{negl}(n).$$

Using Chernoff (Lemma 2.2), we get that

$$\mathbf{Pr}\left[\sum_i L_i\leq t(n)\cdot\mathrm{H}(X_n\mid G(X_n))-t(n)/2\right]\leq\mathbf{Pr}\left[\sum_i\widetilde{L_i}\leq t(n)\cdot\mathrm{H}(X_n\mid G(X_n))-t(n)/2\right]\leq\mathsf{negl}(n),$$

which implies that

$$\mathop{\mathbf{Pr}}_{\substack{x_1,\ldots,x_n\leftarrow(X_n)^{t(n)}\\ y_1,\ldots,y_{t(n)}=G(x_1),\ldots,G(x_{t(n)})}}\left[\mathbf{Pr}\left[(X_n^1,\ldots,X_n^{t(n)})=(x_1,\ldots,x_{t(n)})\mid(G(X_n^1),\ldots,G(X_n^{t(n)}))=(y_1,\ldots,y_{t(n)})\right]\geq 2^{-r'(n)}\right]$$

$$=\mathop{\mathbf{Pr}}_{x_1,\ldots,x_n\leftarrow X_n^1,\ldots,X_n^{t(n)}}\left[\prod_i 2^{-L_i}\geq 2^{-r'(n)}\right]$$

$$=\mathop{\mathbf{Pr}}_{x_1,\ldots,x_n\leftarrow X_n^1,\ldots,X_n^{t(n)}}\left[2^{-\sum_i L_i}\geq 2^{-(t(n)\mathrm{H}(X_n|G(X_n))-t(n)/2)}\right]\leq\mathsf{negl}(n)$$

as we wanted to show. □

6.2 Proving Lemma 6.3

Proof of Lemma 6.3. Let $t(n) \geq (\sqrt{(k(n)\ell(n))log^2(n)} + 2\ell(n)\log^2 n + 1)^2$ be a polynomial, and let

$$G(H_n, A_n, X_n^1, \ldots, X_n^{t(n)}) = (H_n, A_n, H_n(g_{A_n}(X_n^1)_1, \ldots, g_{A_n}(X_n^{t(n)})_1), \ldots, H_n(g_{A_n}(X_n^1)_{\ell(n)}, \ldots, g_{A_n}(X_n^{t(n)})_{\ell(n)})),$$

where H, A, X_n^i are defined as in Theorem 6.1. By Theorem 6.1, the output of G is pseudo-random. To get that the output of G is long enough, we need to have

$$q(n) \cdot \ell(n) \geq t(n)(\mathrm{H}(f(\mathcal{D}_n)) + (m(n) - \mathrm{H}(\mathcal{D}_n))) + 1,$$

where $q(n)$ is defined as in Theorem 6.1. Since

$$t(n)(k(n) - 1) + 1 \geq t(n)(\mathrm{H}(f(\mathcal{D}_n)) + (m(n) - \mathrm{H}(\mathcal{D}_n))) + 1,$$

it is enough to show that

$$q(n) \cdot \ell(n) \geq t(n)(k(n) - 1) + 1.$$

Indeed,

Then

$$\begin{aligned} q(n)\ell(n) &\geq k(n)t(n) - \sqrt{(k(n)\ell(n))t(n)log^2(n)} - 2\ell(n)\log^2 n \\ &= t(n)(k(n) - 1) + 1 + t(n) - \sqrt{(k(n)\ell(n))t(n)log^2(n)} - 2\ell(n)\log^2 n - 1 \end{aligned}$$

Thus, it is enough to show that

$$t(n) \geq \sqrt{(k(n)\ell(n))t(n)log^2(n)} + 2\ell(n)\log^2 n + 1$$

Or,

$$\sqrt{t(n)} \geq \sqrt{(k(n)\ell(n))log^2(n)} + 2\ell(n)\log^2 n + 1,$$

which holds by our choice of t. □

Acknowledgments

This work was initiated and much of it was conducted while Gal Arnon, Noam Mazor and Jad Silbak were research fellows at the Simons Institute for the Theory of Computing.

Gal Arnon is supported by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme (Grant agreement No. 101019547) and by a Cariplo CRYPTONOMEX grant. The majority of this work was done while Gal Arnon was a postdoc at Bocconi University. Noam Mazor is supported by the Simons Foundation. Rafael Pass is 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, the AFOSR, the European Union or the European Research Council Executive Agency.

AI Use Statement

The results and proofs in this work were found by humans. ChatGPT was used solely for writing assistance. The authors take full responsibility for all claims and proofs made in this paper.

References

[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.

[AMPS26] Gal Arnon, Noam Mazor, Rafael Pass, and Jad Silbak. *Witness-Indistinguishable Arguments of Knowledge and One-Way Functions*. Cryptology ePrint Archive, Paper 2026/682. 2026. URL: https://eprint.iacr.org/2026/682.

[BC20] Nir Bitansky and Arka Rai Choudhuri. "Characterizing deterministic-prover zero knowledge". In: *Theory of Cryptography Conference*. Springer. 2020, pp. 535–566.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. "Minimum Disclosure Proofs of Knowledge". In: *J. Comput. Syst. Sci.* 37.2 (1988), pp. 156–189.

[BH26] Idan Baril and Iftach Haitner. *On the Complexity of Interactive Arguments*. Cryptology ePrint Archive, Paper 2026/272. 2026. URL: https://eprint.iacr.org/2026/272.

[BM82] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo Random Bits". In: 1982, pp. 112–117.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil Vadhan. "Derandomization in cryptography". In: *SIAM Journal on Computing* 37.2 (2007), pp. 380–400.

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

[CHK25] Suvradip Chakraborty, James Hulett, and Dakshita Khurana. "On Weak NIZKs, One-Way Functions and Amplification". In: *Annual International Cryptology Conference*. Springer. 2025, pp. 580–610.

[CHKT26] Suvradip Chakraborty, James Hulett, Dakshita Khurana, and Kabir Tomer. "Non-Trivial Zero-Knowledge Implies One-Way Functions". In: *arXiv preprint arXiv:2602.17651* (2026).

[CLMP12] Kai-Min Chung, Edward Lui, Mohammad Mahmoody, and Rafael Pass. "Unprovable security of 2-message zero knowledge". In: *Cryptology ePrint Archive* (2012).

[CLV26] Rohit Chatterjee, Yunqi Li, and Prashant Nalini Vasudevan. "Weak Zero-Knowledge and One-Way Functions". In: *arXiv preprint arXiv:2602.16156* (2026).

[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.

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: (1976), pp. 644–654.

[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.

[FW26] Cody Freitag and Daniel Wichs. "Unique SNARGs with Adaptive Security: Constructions and Black-Box Separations". In: *Advances in Cryptology – CRYPTO 2026*. Ed. by Nadia Heninger and Mike Rosulek. Vol. 16800. Lecture Notes in Computer Science. Springer, 2026, pp. 190–217. DOI: 10.1007/978-3-032-35367-2_7.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions". In: (1986), pp. 792–807.

[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*. 1989, pp. 25–32.

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: (1984), pp. 270–299.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM J. Comput.* 18.1 (1989), pp. 186–208.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems". In: *Journal of the ACM (JACM)* 38.3 (1991), pp. 690–728.

[GO94a] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *J. Cryptol.* 7.1 (1994), pp. 1–32.

[GO94b] Oded Goldreich and Yair Oren. "Definitions and properties of zero-knowledge proof systems". In: *Journal of Cryptology* 7.1 (1994), pp. 1–32.

[GW11] Craig Gentry and Daniel Wichs. "Separating succinct non-interactive arguments from all falsifiable assumptions". In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 2011, pp. 99–108.

[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.

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

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

[HN24] Shuichi Hirahara and Mikito Nanashima. "One-Way Functions and Zero Knowledge". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM, 2024, pp. 1731–1738.

[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.

[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.

[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.

[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.

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

[IL89] Russell Impagliazzo and Michael Luby. "One-way functions are essential for complexity based cryptography". In: *30th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society. 1989, pp. 230–235.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[LMP24] Yanyi Liu, Noam Mazor, and Rafael Pass. "A note on zero-knowledge for NP and one-way functions". In: *Cryptology ePrint Archive* (2024).

[LMP26] Yanyi Liu, Noam Mazor, and Rafael Pass. "Cryptographic Implications of Worst-Case Hardness of Time-Bounded Kolmogorov Complexity". In: *Cryptology ePrint Archive* (2026).

[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.

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

[MP23] Noam Mazor and Rafael Pass. "Counting unpredictable bits: a simple PRG from one-way functions". In: *Theory of Cryptography Conference*. Springer. 2023, pp. 191–218.

[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.

[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.

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

[NY89] Moni Naor and Moti Yung. "Universal One-Way Hash Functions and their Cryptographic Applications". In: 1989, pp. 33–43.

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

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Functions are Essential for Non-Trivial Zero-Knowledge". In: *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*. IEEE Computer Society, 1993, pp. 3–17.

[Ost91] Rafail Ostrovsky. "One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs." In: *SCT*. 1991, pp. 133–138.

[Pas03] Rafael Pass. "Simulation in quasi-polynomial time, and its application to protocol composition". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2003, pp. 160–176.

[Pas04] Rafael Pass. "Alternative variants of zero-knowledge proofs". PhD thesis. 2004.

[RR94] Alexander A Razborov and Steven Rudich. "Natural proofs". In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. 1994, pp. 204–213.

[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.

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

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

[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.

[Sip83] Michael Sipser. "A complexity theoretic approach to randomness". In: *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. 1983, pp. 330–335.

[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.

[Wee06] Hoeteck Wee. "Finding pessiland". In: *Theory of Cryptography Conference*. Springer. 2006, pp. 429–442.

A Unique Succinct Non-interactive Arguments

Definition A.1 (SNARGs for NP)**.** *Let L be an NP language with witness relation R. We say that a tuple $\Pi = (G, P, V)$ is a* succinct non-interactive argument (SNARG) *for R if it has the following syntax and efficiency:*

- *$crs \leftarrow G(1^n)$: A randomized polynomial-time algorithm that on input 1^n (that is also the security parameter) outputs a common reference string crs.*
- *$\pi = P(crs, x, w)$: A deterministic polynomial-time algorithm that on input the common reference string crs, an input x, and a witness w, outputs a proof π.*
- *$b = V(crs, x, \pi)$: A deterministic polynomial-time algorithm that on input the common reference string crs, an input x, and a proof π, outputs a bit b indicating whether to accept or reject.*

We require that $\Pi = (G, P, V)$ satisfy the following properties:

- ***(Completeness:** For all $n \in \mathbb{N}$ and $(x, w) \in R \cap \{0,1\}^n$, it holds that*

$$\mathbf{Pr}\left[\begin{array}{l} crs \leftarrow G(1^n) \\ \pi = P(crs, x, w) \end{array} : V(crs, x, \pi) = 1\right] = 1 - \mathsf{negl}(\lambda).$$

- ***(Adaptive) Soundness:** For all non-uniform PPT attackers A and polynomials m, there exists a negligible function ϵ such that for all $n \in \mathbb{N}$, it holds that*

$$\mathbf{Pr}\left[\begin{array}{l} crs \leftarrow G(1^n) \\ (x, \pi) \leftarrow A(crs) \end{array} : \begin{array}{c} V(crs, x, \pi) = 1 \\ \wedge\ x \notin L \end{array}\right] \leq \mathsf{negl}(\lambda)).$$

- ***Succinct Proofs:** There exists a function $p(n) \in n^{o(1)}$ such that for all $n \in \mathbb{N}$, $(x, w) \in R \cap \{0,1\}^n$, $crs \in \mathrm{Supp}(G(1^n))$, and $\pi = P(crs, x, w)$, it holds that $|\pi| \leq p(n)$.*

We use the following definition of Freitag and Wichs.

Definition A.2 (Uniqueness [FW26])**.** *We say that a SNARG $\Pi = (G, P, V)$ for a language L with witness relation R has α-uniqueness if for all $n \in \mathbb{N}$ and $(x, w) \in R \cap \{0,1\}^n$, it holds that*

$$\mathbf{Pr}\left[crs \leftarrow G(1^n) : \exists \pi \neq \pi' \text{ s.t. } V(crs, x, \pi) = 1 \wedge V(crs, x, \pi') = 1\right] \leq \mathsf{negl}(n).$$

We observe the following simple lemma.

Lemma A.3. *Let $\Pi = (P, V)$ be a unique SNARG with message length $c(n)$. Then Π has a simulator* Sim *of size $2^{c(n)} \cdot \mathrm{poly}(n)$.*

Proof. On input $x \in \{0,1\}^n$, Sim samples $crs \leftarrow G(1^n)$ and enumerates, in a fixed order, all strings $\pi \in \{0,1\}^{\leq c(n)}$. It lets π^* be the first string for which $V(crs, x, \pi^*) = 1$, using a fixed default string if no such proof exists, and outputs (crs, π^*).

Consider an honest transcript (crs, π), where $\pi \leftarrow P(crs, x, w)$. Except in the cases that the honest proof is rejected, or there exist two distinct accepting proofs, the proof π is the unique accepting proof and hence $\pi^* = \pi$. Therefore,

$$\mathsf{SD}\big(\mathrm{Sim}(x), (crs, P(crs, x, w))\big) \leq \mathsf{err}_{\mathsf{comp}}(n) + \mathsf{err}_{\mathsf{uniq}}(n) = \mathsf{negl}(n).$$

The size of Sim is $2^{c(n)} \cdot \mathrm{poly}(n)$. □

We get the following corollary.

Corollary A.4. *Assume a language $\mathcal{L} \notin \mathsf{ioSIZE}[T_L]$ has a Unique SNARG with message length $c(n) = \log T_L - \omega(\log n)$. Then there exists $t \in \mathrm{poly}$ such that $\mathrm{MK}^{\mathrm{t}}\mathrm{P}[n-2]$ is average-case hard on the uniform distribution for efficient errorless algorithms.*

Proof. Immediate by Section A and Theorem 5.1. □

B Deferred Proofs

B.1 Proving Lemma 2.4

Lemma B.1 (Lemma 2.4, restated)**.** *Assume that $\mathcal{L} \notin \mathsf{ioSIZE}[T(n)]$. Then, there exists a pair of distributions ensemble $\mathcal{D}^Y = \{\mathcal{D}^Y_n\}_{n\in\mathbb{N}}, \mathcal{D}^N = \{\mathcal{D}^N_n\}_{n\in\mathbb{N}}$ with $\mathcal{D}^Y_n \subseteq \mathcal{L}_n$ and $\mathcal{D}^N_n \subseteq \{0,1\}^n \setminus \mathcal{L}_n$, such that*

$$\left| \Pr_{x\leftarrow\mathcal{D}^Y}[A(x)=1] - \Pr_{x\leftarrow\mathcal{D}^N}[A(x)=1] \right| \leq \epsilon(n)$$

for any $A \in \mathsf{SIZE}[\frac{\epsilon^2(n)(T(n)-\mathrm{poly}(n,1/\epsilon(n)))}{160n}]$ and any large enough n.

Proof of Lemma 2.4. Let $s(n) = \frac{\epsilon^2(n)(T(n)-\mathrm{poly}(n,1/\epsilon(n)))}{160n}$. We start by showing that there exists a distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ with $\mathcal{D}_n \in \{0,1\}^n$, such that

$$\Pr_{x\leftarrow\mathcal{D}_n}[A(x) = L(x)] \leq 1/2 + \epsilon(n)/4$$

for any $A \in \mathsf{SIZE}[s(n)]$ and any large enough n. Later we will show how to construct the ensembles $\mathcal{D}^Y, \mathcal{D}^N$ from $\mathcal{D}$.

We want to show that there is a hard distribution against any algorithm in $\mathsf{SIZE}[s(n)]$. Towards this, for $n \in \mathbb{N}$, consider a two-party game, where Alice chooses an algorithm A, and Bob chooses an input $x \in \{0,1\}^n$. Alice wins if $A(x) = L(x)$, and Bob wins otherwise. We want to show that for a large enough $n \in \mathbb{N}$, Bob have a mixed strategy, such that when he plays first Alice wins the game with probability at most $1/2 + \epsilon(n)/4$. By the min-max theorem it is enough to show that when Alice plays first, Bob has such a strategy. That is, we want to show that for every distribution $\mathcal{P}_n$ over circuits of size $s(n)$, there is an input $x \in \{0,1\}^n$, for which

$$\Pr_{A\leftarrow\mathcal{P}_n}[A(x) = L(x)] \leq 1/2 + \epsilon(n)/4.$$

Assume towards a contradiction this is not the case. That is, for every $x \in \{0,1\}^n$,

$$\Pr_{A\leftarrow\mathcal{P}_n}[A(x) = L(x)] > 1/2 + \epsilon(n)/4.$$

We will construct a circuit C of size $T(n)$ that decides L_n, which is a contradiction to the assumption. To construct C, let $k = 160n/\epsilon^2(n)$, and let $A_1, \ldots, A_k$ be k random samples from $\mathcal{P}_n$. Let $C_{A_1,\ldots,A_k}$ be the circuit that given an input x computes $A_1(x), \ldots, A_k(x)$, and outputs the majority of the answers. By assumption and Chernoff bound (Lemma 2.2), $x \in \{0,1\}^n$,

$$\Pr_{A_1,\ldots,A_k\leftarrow\mathcal{P}_n}[C_{A_1,\ldots,A_k}(x) \neq L(x)] \leq \mathbf{Pr}\left[\left|\sum_{i=1}^{k} A_i(x) - k\cdot\mathbb{E}[A_i(x)]\right| > \frac{k\epsilon(n)}{4}\right] < 2^{-n},$$

and thus by the union bound, there is a choice of $A_1, \ldots, A_k$ such that $C_{A_1,\ldots,A_k}$ will answer correctly for any $x \in \{0,1\}^n$. Since the size of $C_{A_1,\ldots,A_k}$ is at most $\sum_i |A_i| + k^2 \leq T$, we get a contradiction to the assumption that $L \notin \mathsf{SIZE}[T]$.

Observe that by the fact that $\mathcal{D}_n$ is hard for small circuits, it must hold that $\mathbf{Pr}_{x\leftarrow\mathcal{D}_n}[x \in L] \in [1/2 - \epsilon/4, 1/2 + \epsilon/4]$. It follows that we can choose a distribution $\mathcal{D}'$ such that $\mathrm{SD}(\mathcal{D}_n, \mathcal{D}'_n) \leq \epsilon/4$, and

$$\Pr_{x\leftarrow\mathcal{D}'_n}[x \in L] = 1/2.$$

Since $\mathrm{SD}(\mathcal{D}_n, \mathcal{D}'_n) \leq \epsilon/4$, we get that for every $A \in \mathsf{SIZE}[s(n)]$, it holds that

$$\Pr_{x\leftarrow\mathcal{D}'_n}[A(x) = L(x)] \leq 1/2 + \epsilon(n)/2.$$

We next define the two ensembles $\mathcal{D}^Y, \mathcal{D}^N$. For $n \in \mathbb{N}$, let $\mathcal{D}^Y_n$ be the distribution of $(x \leftarrow \mathcal{D}'_n)|_{x\in L}$, and let $\mathcal{D}^N_n$ be the distribution of $(x \leftarrow \mathcal{D}'_n)|_{x\notin L}$. Assume that there is an algorithm $A \in \mathsf{SIZE}[s(n)]$ such that

$$\left| \Pr_{x\leftarrow\mathcal{D}^Y}[A(x)=1] - \Pr_{x\leftarrow\mathcal{D}^N}[A(x)=1] \right| > \epsilon(n)$$

and assume without loss of generality that

$$\mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^Y}[A(x)=1] > \mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^N}[A(x)=1]+\epsilon(n).$$

We get that

$$\mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}'_n}[A(x)=L(x)] = 1/2\mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^Y_n}[A(x)=1]+1/2(1-\mathop{\mathbf{Pr}}_{x\leftarrow\mathcal{D}^N_n}[A(x)=1]) > 1/2+\epsilon(n)/2,$$

with contradiction to the choice of $\mathcal{D}'$. □

B.2 Proving Lemma 2.11

Lemma B.2 (Lemma 2.11, restated)**.** *There exists an oracle-aided* $\mathsf{PPT}\widehat{P}$ *such that the following holds. Let* $\ell, n \in N$ *be numbers, and* $\mathcal{D}$ *a distribution over* $\{0,1\}^n \times \{0,1\}^*$*. and let* A *be an algorithm such that*

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z,M,M(x)_{\le\ell})=1] - \mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z,M,\mathbf{U}_\ell)=1] \ge \alpha$$

for some α*, where* $GL(x,r) := \langle x,r\rangle$ *is the Goldreich-Levin predicate. Then*

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}\left[\widehat{P}^A(1^n,1^{\lceil 1/\alpha\rceil},\ell,z,M,M(x)_{\le\ell})=x\right] \ge \alpha^3/8n\ell^4.$$

Proof. Assume that A is such that

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z,M,M(x)_{\le\ell})=1] - \mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},r\leftarrow\{0,1\}^n}[A(z,r,\mathbf{U}_\ell)=1] \ge \alpha.$$

By a simple hybrid argument, there exists $i\in[\ell]$ such that

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}[A(z,M,M(x)_{<i},M(x)_i)=1] - \mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},r\leftarrow\{0,1\}^n}[A(z,r,M(x)_{<i},\mathbf{U}_1)=1] \ge \alpha/\ell.$$

By Lemma 2.10, it holds that

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D},M\leftarrow\{0,1\}^{n\times n}}\left[P^A(1^n,1^{\lceil 1/\alpha\rceil},z,M,M(x)_{<i})=x\right] \ge (\alpha/\ell)^3/8n.$$

Let $\widehat{P}$ be the algorithm that given $1^n, 1^{\lceil 1/\alpha\rceil}, M, M(x)_{<i}$ and z, guesses the value of i,and $P^A(1^n,1^{\lceil 1/\alpha\rceil},z,M,M(x)_{<i})$.

Then $\widehat{P}$ guesses correctly i with probability at least $1/\ell$, thus

$$\mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathcal{D}}\left[\widehat{P}^A(1^n,1^{\lceil 1/\alpha\rceil},z,\ell)=x\right] \ge 1/\ell\cdot(\alpha/\ell)^3/8n.$$

□

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