Electronic Colloquium on Computational Complexity, Report No. 190 (2025)

The Oracle Derandomization Hypothesis is False (And More) Assuming No Natural Proofs

November 18, 2025

Abstract

Razborov and Rudich's natural proofs barrier roughly says that it is computationally hard to certify that a uniformly random truth table has high circuit complexity. In this work, we show that the natural proofs barrier (specifically, Rudich's conjecture that there are no NP-constructive natural properties against P/poly) implies the following important consequences in derandomization, proof complexity, and cryptography.

1. *Derandomization:* **Fortnow and Santhanam's *Oracle Derandomization Hypothesis* is false.** In particular, this means that one cannot use the hardness-versus-randomness paradigm to derandomize data structures, at least in the straightforward way. Our result is the first direct evidence that the Oracle Derandomization Hypothesis is false. As a corollary, we also get the first average-case hardness result for the Circuit Range Avoidance Problem (Avoid).
2. *Proof Complexity:* **There is a single non-uniform proof complexity generator secure against all proof systems.** This is the first construction of such a proof complexity generator under any complexity assumption. Indeed, it was previously not clear whether such an object should exist.
3. *Cryptography:* **In the non-uniform setting, zero-knowledge does not require interaction.** We construct a non-uniform, truly non-interactive prover and verifier where the verifier is perfectly sound and the prover has every "falsifiable" consequence of being zero-knowledge. Thus, in the non-uniform setting, we bypass a classical impossibility result of Goldreich and Oren that says zero-knowledge proofs require interaction. This is the first construction of such an object.

ISSN 1433-8092

1 Introduction

A central question in complexity theory is to understand the power of *randomness*. For example, are there problems that have fast randomized algorithms but no fast deterministic algorithms (i.e., is $\mathsf{P} \neq \mathsf{BPP}$)? Historically, many researchers believed that the answer should be *yes*, i.e., $\mathsf{P} \neq \mathsf{BPP}$.[1]

Indeed, it is easy to think of settings where randomness gives exponential power over deterministic algorithms. For example, imagine you have query access to the bits of a string $x \in \{0,1\}^{2^n}$, and you want to distinguish whether x is all zero or is at least half ones. A randomized algorithm can query x on $\mathsf{poly}(n)$ random indices and distinguish the two cases with exponentially small failure probability. On the other hand, a deterministic algorithm must make $\Omega(2^n)$ queries to distinguish the two cases.

Today, researchers generally believe that the example above gives the *wrong* intuition and that BPP actually equals P. This reversal is in large part due to the discovery of the *hardness-versus-randomness paradigm* [NW94]. For instance, one can utilize the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ that has large circuit complexity to deterministically generate bits that — despite not being random — look random to computationally bounded adversaries. In a celebrated result, Impagliazzo and Wigderson [IW97] use this approach to show that $\mathsf{P} = \mathsf{BPP}$ if E requires circuits of size $2^{\Omega(n)}$.

Thus, it may appear that, assuming one believes that sufficiently strong lower bounds hold, then hardness-versus-randomness lets us more-or-less completely understand the power of randomness. But that is not quite true. For instance, consider the setting of *randomized data structures*. In this setting, there is a data structure $D \in \{0,1\}^\star$ and a randomized algorithm A that solves some task given query access to D. Ideally, one wants to replace A with a similarly efficient deterministic algorithm.

Can we do this using hardness-versus-randomness? It turns out that the natural way to do so requires one to come up with a function f such that

- f is hard to compute *even given oracle access to* D, and
- the truth table of f is significantly shorter than $|D|$ (this is needed for the derandomization overhead to be efficient).

A random function f of the appropriate length will have both of these properties with high probability. But if we want to derandomize the data structure, we want to generate such f deterministically. Can this be done? In other words, given $D \in \{0,1\}^\star$, can we quickly and efficiently generate such an f?

Fortnow and Santhanam's [FS11] *Oracle Derandomization Hypothesis* hypothesizes that the answer is *yes*. To aid readability, we state a slightly weaker informal version of the hypothesis below. Here $\mathsf{CC}(f \mid g)$ denotes the size of the smallest circuit computing f with g-oracle gates. Throughout this paper we identify a function $f : \{0,1\}^n \to \{0,1\}$ and its 2^n-bit truth table.

Hypothesis 1.1 (The Oracle Derandomization Hypothesis (Informal Version of Hypothesis 3.1))**.** *There is a deterministic polynomial time algorithm G that, given a 2^n-bit string x, outputs a $2^{.01n}$-bit string y such that $\mathsf{CC}(x|y) = 2^{\Omega(n)}$.*

Fortnow and Santhanam use the Oracle Derandomization Hypothesis to rule out the existence of certain types of PCPs (assuming the hypothesis is true). They also feel that the hypothesis is valuable for "test[ing] our intuitions of which kinds of derandomization are plausible and which are not." Indeed, its importance is further bolstered by the aforementioned connection to derandomizing data structures.[2] A frontier question in derandomization is whether or not this hypothesis is true.

Question 1.2. Is the Oracle Derandomization Hypothesis true? Is there a plausible assumption under which it is true or false?

[1] For one example, see the discussion of RP in [GMR85].

[2] This data structures perspective was suggested to us by Lijie Chen.

1.1 Our Results

We show that the natural proofs barrier [RR97] (specifically, Rudich's conjecture [Rud97] that there are no NP-constructive natural properties against P/poly, which we will explain soon) implies that the Oracle Derandomization Hypothesis is *false*. We interpret this as strong evidence that the Oracle Derandomization Hypothesis is indeed false and hence that using hardness-versus-randomness to derandomize data structures does not work, at least in the most straightforward way.

Before we state our result, we discuss the natural proofs barrier and the precise assumption we use. The core of Razborov and Rudich's natural proof barrier is the assumption that there is no "P-constructive natural property against P/poly." Intuitively, this assumption says that it is hard to certify in polynomial-time that a uniformly random 2^n-bit truth table has superpolynomial circuit complexity. In follow-up work, Rudich [Rud97] further conjectured that this hardness even holds for non-deterministic polynomial-time algorithms.[3] The formal definition is as follows.

Definition 1.3 (P and NP-constructive natural properties against P/poly [RR97; Rud97])**.** A polynomial-time (respectively, non-deterministic polynomial-time) algorithm A is a P-*constructive (respectively,* NP-*constructive) natural property against* P/poly if there exists an $s = n^{\omega(1)}$ such that for all sufficiently large n we have that

- $A(T)$ accepts at least half[4] of all 2^n-bit truth tables T
- $A(T)$ rejects if T has a circuit of size at most $s(n)$.

Using Rudich's conjecture, we show that the Oracle Derandomization Hypothesis is false.

Theorem 1.4 (Less Detailed Theorem 3.3)**.** *Assume there are no* NP*-constructive natural properties against* P/poly*. Then the Oracle Derandomization Hypothesis is false. In fact, even the Weak Oracle Derandomization Hypothesis (defined below) is false.*

Our proof refutes even a severe weakening of the Oracle Derandomization Hypothesis that (for lack of a better name) we call the *Weak Oracle Derandomization Hypothesis*, which simultaneously weakens the Oracle Derandomization Hypothesis in four different ways:

- *Non-deterministic algorithm*: G is now allowed to use non-determinism.
- *Deterministic lower bound*: We only require a lower bound against deterministic circuits instead of non-deterministic circuits.
- *Superpolynomial lower bound*: We require a superpolynomial lower bound instead of an exponential lower bound.
- *Average-case correctness*: We only require the algorithm works with zero-error on average on the uniform distribution with zero-error.

In more detail, we have the following.

Hypothesis 1.5 (The Weak Oracle Derandomization Hypothesis)**.** *There is a non-deterministic polynomial-time algorithm G that, given a 2^n-bit string x, either outputs ? or a $2^{.01n}$-bit string y such that the smallest x-oracle circuit for y has size $n^{\omega(1)}$. Additionally, $G(x) \neq ?$ for at least a $2^{-2^{.01n}}$ fraction of all 2^n-bit strings x.*

Besides being interesting in its own right, our refutation of the Weak Oracle Derandomization Hypothesis has important consequences. As immediate corollaries, it resolves another frontier question in derandomization (average-case hardness for the Circuit Range Avoidance Problem [KKMP21]) and also a frontier question in proof complexity (a single proof complexity generator secure against all proof systems). We now discuss these two corollaries and the necessary background.

[3] In fact, Rudich made the stronger conjecture that there are no $\mathsf{NTIME}[2^{\mathsf{poly}\log n}]/\mathsf{poly}$-constructive natural properties against P/poly.

[4] For readers familiar with natural properties, here we have set the largeness density to be half for simplicity.

The First Average-Case Hardness for Range Avoidance. The Circuit Range Avoidance Problem (Avoid) [KKMP21] asks one to find a string $y \in \{0,1\}^m$ that is outside the range of a length-increasing circuit $C : \{0,1\}^n \to \{0,1\}^m$. Since the circuit has more output bits than input bits, such a string must exist. Indeed, a uniformly random m-bit string works with probability $1 - 2^{n-m}$.

In recent years, there have been a flurry of works studying Avoid and its applications [Kor21; RSW22; ILW23; CL24; CHLR23; GGNS23; KP24; CHR24; Li24; GLW25]. One of the main questions in these works is understanding the complexity of solving Avoid deterministically. Recent work shows that no efficient deterministic *worst-case* algorithm for Avoid exists under plausible assumptions [ILW23; CL24].[5] It is easy to see that an efficient deterministic algorithm for Avoid lets one deterministically generate a string with large x-oracle circuit complexity (run Avoid on the "oracle truth table generator" circuit that maps descriptions of x-oracle circuits to their truth table). As a result, an immediate corollary of our refutation of the weak Oracle Derandomization Hypothesis is the first average-case hardness for Avoid.

Corollary 1.6 (Informal Corollary 3.7)**.** *Assume there are no* NP*-constructive natural properties against* P/poly*. Then there is an efficiently samplable distribution on which* Avoid *is zero-error average-case hard for non-deterministic polynomial-time algorithms.*

A Single Proof Complexity Generator Secure Against All Proof Systems. Another (almost immediate) corollary of our results is the first construction of a single non-uniform proof complexity generator simultaneously secure against all proof systems.

We now explain what this means. In proof complexity, one is generally interested in understanding how long proofs of coNP statements must be in various proof systems. In this setting, a proof system [CR79] for a language L (e.g. $L = \mathsf{UNSAT}$) is just a polynomial-time algorithm $V(x, \pi)$ that takes as input a "statement" string x and a "proof of membership in L" string π and either accepts or rejects. V is required to be *sound* in the sense that if $V(x, \pi)$ accepts then $x \in L$. One can also require completeness, but we will not need it.

An important and well-studied (e.g., [Kra01b; Kra04a; Kra04b; Kra05; Kra07; Kra11a; Kra11b; Pic11; Raz15; RSW22; Kha22; Kra22]) notion in proof complexity is that of a *proof complexity generator* [ABRW04; Kra01a]. Specifically, a *proof complexity generator against a proof system* V is a (possibly non-uniform) length-increasing circuit $G_n : \{0,1\}^n \to \{0,1\}^m$ where for *every* string $y \in \{0,1\}^m$ there are no $\mathsf{poly}(n)$-length V-proofs of the coNP statement that "y is not in the range of G_n." Krajíček [Kra04b] first raised the possibility that there could be a single proof complexity generator that is simultaneously secure against *all* proof systems.

Conjecture 1.7 (Krajíček [Kra04b] (see also [Kra22, Conjecture 1.1]))**.** *There is a single proof complexity generator secure against all proof systems.*

We remark that it would be quite surprising if such a generator G could be computable in *uniform* polynomial-time, at least when $m \geq n + \omega(1)$. This is because any string y with t-time-bounded Kolmogorov complexity[6] greater than $n + O(1)$ for a sufficiently large polynomial t must be outside the range of G (since otherwise one could get a short description for y by giving a pre-image of it and running G on it). It seems plausible that one can generate a sequence of such y in uniform polynomial-time [KS25] and thus get a P-uniform sequence of strings that are outside the range of G. Then there is a trivial proof system V which knows the description of these strings and has short proofs that they are not in the range of G.

As a result, to get a single proof complexity generator secure against all proof systems it seems like one must necessarily consider *non-uniform* proof complexity generators. Indeed this is what we get. An immediate consequence of Theorem 1.4 is a non-uniform proof complexity generator secure against all proof systems. To the best of our knowledge, this is the first such construction under *any* plausible complexity assumption. Moreover, this construction has nearly exponential stretch.

[5] The assumptions in [ILW23] are NP $\neq$ coNP and the existence of a subexponentially-secure indistinguishability obfuscator [BGI+12; JLS21].

[6] The t-time-bounded Kolmogorov complexity of a string y is length of the shortest program that outputs y in time at most t. We say it is near maximum if its complexity is at least, say, $|y| - 10$.

Corollary 1.8 (Informal Theorem 3.6)**.** *Assume there are no infinitely often* NP*-constructive natural properties against* P/poly*. Then for every $s(n) = n^{\omega(1)}$ there is a single non-uniform proof complexity generator $G : \{0,1\}^{\mathsf{poly}(s(n))} \to \{0,1\}^{2^n}$ that is simultaneously secure against all proof systems.*

Indeed, the construction is very simple and efficiently samplable. Pick a sequence of functions $g_n : \{0,1\}^{3n} \to \{0,1\}$ uniformly at random. The generator G_n is the circuit that takes as input the description of a $s(n)$-size oracle circuit C and outputs the truth table of C using oracle g_n. Proving a string f is outside the range of G_n is equivalent to proving a $s(n)$-size lower bound on the g_n-oracle circuit complexity of f. Applying Theorem 1.4 and union bounding[7] over every proof system, we get that G_n is a proof complexity generator against all proof systems with probability one (in the measure theoretic sense).

1.1.1 Bypassing Cryptographic Impossibilities: Zero-Knowledge Without Interaction

We believe that our results should be useful for overcoming barriers in cryptography. In particular, several impossibility results in cryptography (e.g., [HW15]) are proved via "incompressibility arguments." In such arguments, one shows that a certain object cannot exist because if it did exist, then one could describe an arbitrary n-bit string with a description of length less than n, which contradicts the pigeonhole principle (since two different strings would have the same description).

On the other hand, one interpretation of the proof complexity generator in Corollary 1.8 is as follows: it is hard to rule out that any particular function $f : \{0,1\}^n \to \{0,1\}$ has a small g_n-oracle circuit (and hence a short description given g_n). This is despite the fact that the pigeonhole principle shows that not every function can have a short description, even given g_n. As a result, we suspect that one might be able to use Corollary 1.8 to bypass an incompressibility-based impossibility argument. It's worth noting that in many cryptographic settings it is okay to sample a g_n uniformly at random (perhaps as part of a "common reference string").

Despite this initial intuition, our results did not end up going along these lines, although we still think that it is a promising approach. Instead, we use our results to overcome a cryptographic impossibility result for zero-knowledge proofs [GO94] that is (to our knowledge) unrelated to incompressibility arguments. Besides being remarkable in its own right, this result is strong evidence that our results (and similar ones) are useful in cryptography, which we view as a major conceptual contribution of this paper. We now describe the zero-knowledge result.

Zero-knowledge proofs [GMR89] allow a prover to convince a verifier that a statement is true (e.g., that x is in a language L) without revealing *anything* besides the validity of the statement. The crux of the definition is the notion of a simulator. Intuitively, if the verifier can simulate how the interaction with the prover will go, then it cannot learn anything it did not already know.

Amazingly, zero-knowledge proofs exist for all languages in NP [GMW86] (and even all of PSPACE [IY87; Sha92]). However, these proofs differ significantly from the traditional notion of a mathematical proof:

- **Interaction:** All known zero-knowledge proof systems for NP require the prover and verifier to engage in a multi-round interactive protocol or require trusted setup [BFM88; BSMP91]. In contrast, a traditional mathematical proof is a single string a prover can send to a verifier.
- **Soundness:** In all known zero-knowledge proof systems, the verifier can be convinced of false statements, just with low probability. In contrast, traditional mathematical proofs offer perfect soundness (if there is a proof, then the statement is true).

Indeed, Goldreich and Oren [GO94] show it is impossible to remove either interaction or imperfect soundness from zero-knowledge proofs for languages outside of BPP. This impossibility even holds if the prover and verifier are allowed to be non-uniform.

Perhaps surprisingly, we overcome this barrier. Building on several ideas in prior work [KP89; Pud86; FS90; FLS90; BOV07; KZ20], we show that there are traditional mathematical proofs for NP statements

[7] This really means applying the Borel-Cantelli lemma, which one can think of a generalization of the union bound to handle infinitely many events.

that achieve almost all security properties guaranteed by zero-knowledge. Our key idea is this: rather than requiring a simulator actually exist, we require that it is *hard to refute* the existence of a simulator in a proof system. This led us to the following definition (which uses terms we have not introduced yet, but should still give the reader the flavor of its meaning).

Definition 1.9 (Effectively Zero-Knowledge to a Proof System)**.** Say a prover P is *effectively zero-knowledge to a proof system* V if for some $\epsilon = \lambda^{-\omega(1)}, \ell = \lambda^{\omega(1)}$ and some $s = \mathsf{poly}(\lambda)$ we have that for all λ there is no $\ell(\lambda)$-length V-proof that

$$\text{“}P_\lambda \text{ lacks an } \epsilon(\lambda) \text{ secure simulator of size } s(\lambda)\text{”}$$

The usefulness of this definition is as follows: Suppose ZFC proves that ρ is implied by zero-knowledge. If P is zero-knowledge to ZFC, then there can be no conclusive evidence that ρ fails, since such evidence would refute the existence of a simulator, contradicting that P is zero-knowledge to ZFC. In a certain sense, one can view this as saying P is indistinguishable from being zero-knowledge to ZFC.

Remarkably, we show (under a standard assumption in cryptography and the non-existence of NP-constructive natural properties) that there is a single non-uniform, truly non-interactive prover and verifier that are effectively zero-knowledge to *every* proof system! This means this prover and verifier enjoy almost all natural security properties (classical) zero-knowledge entails.

Our construction relies on the existence of a non-interactive witness indistinguishable proof system for SAT (NIWI) [FS90; BOV07], which follows from several standard assumptions in cryptography [BOV07; GOS12; BP15].

Theorem 1.10 (Less Detailed Theorem 4.10)**.** *Assume a NIWI exists and there are no infinitely often* NP*-constructive natural properties against* $\mathsf{P/poly}$*. Then there exists a non-uniform, non-interactive, perfectly sound prover and verifier that are effectively zero-knowledge to every proof system.*

After proving Theorem 1.10, we realized how to prove a version where the prover and verifier are *uniform* instead of non-uniform. The latter result (a) does not rely on the natural proofs barrier, (b) is of significant cryptographic interest, and (c) requires a large amount of background and exposition to fully explain and use. Thus, to aid readability, we split off the uniform version of this result to a separate paper [Aut25]. Consequently, in this paper we keep our discussion of effectively zero-knowledge proofs relatively short and defer altogether on how to use them, pointing the interested reader to the companion paper [Aut25].

For example, in the companion paper, we show that the construction in Theorem 1.10 is *distributionally witness hiding* [FS90; KZ20] meaning roughly that a proof that a circuit C is satisfiable generated using a witness w does not reveal a satisfying assignment to C whenever C and w are sampled from a P-samplable[8] distribution for which solving $\mathsf{Search\text{-}SAT}$ on C is hard. Our result is the first construction of such a non-interactive non-uniform prover on which this property holds for all P-sampleable distributions simultaneously. This improves on prior work by Kuykendall and Zhandry [KZ20], who gave a construction where the prover and verifier depend in a non-uniform way on the precise distribution one wants to be witness hiding against.

1.2 Related Work

We now discuss some related work that we have not already discussed. Ilango, Li, and Williams [ILW23] made partial progress on refuting the Oracle Derandomization Hypothesis. Specifically, they rule out a variant of the hypothesis for time-bounded Kolmogorov complexity, assuming $\mathsf{NP} \neq \mathsf{coNP}$ and subexponentially-secure indistinguishability obfuscators exist [BGI+12; JLS22]. In more detail, their variant roughly asks one to, given x, generate a short y with high conditional time-bounded Kolmogorov complexity conditioned on x. It seems difficult to extend their argument to refute the original Oracle Derandomization Hypothesis because of difficulties related to how they use indistinguishability obfuscation.

Krajíček has put forth two candidates for a proof complexity generator against all proof systems, one utilizing the Nisan-Wigderson generator [Kra11a, Chapter 30.3] and one (more general construction) based

[8] This means there is a polynomial-time Turing machine that given 1^n samples from the distribution indexed by n.

on the "gadget" generator [Kra22]. However, to our knowledge, neither construction has a security reduction to a previously-studied complexity assumption.

There are several prior works [BP04; BOV07; BL18; KZ20] studying relaxations of zero-knowledge achievable with truly no interaction. One line of work [BP04; BL18], which we have not yet discussed, achieves (weak [Pas03]) zero-knowledge with zero-interaction by relaxing statistical soundness to *computational* soundness (for comparison, we consider perfect soundness). Specifically, Barak and Pass [BP04] give a construction with uniform soundness (i.e., no uniform algorithm can prove a false statement). Bitansky and Lin [BL18] extend this to a weak form of soundness against non-uniform adversaries (the number of false statements an adversary can generate is bounded by its amount of non-uniformity).

Independent and Concurrent Work. In an independent and concurrent work, Ren, Wang, and Zhong [RWZ25] prove related versions of our results for proof complexity generators (Corollary 1.8) and Avoid (Corollary 1.6). Specifically, under a similar assumption[9] they show:

- worst-case non-deterministic hardness of Avoid (in comparison, we give average-case non-deterministic hardness in Corollary 1.6) and
- for every proof system V, there exists a non-uniform proof complexity generator G_V against V (as opposed to our single non-uniform proof complexity generator G that is *simultaneously* secure against all V in Corollary 1.8). However, [RWZ25] proves stronger properties about their generator G_V than we do, such as pseudosurjectivity.

Interestingly, our techniques are somewhat different from [RWZ25]. Whereas we rely on the parity shift lemma, they rely on randomness extractors. Indeed, our use of the parity shift lemma is useful for ruling out the Oracle Derandomization Hypothesis (Theorem 1.4). [RWZ25] does not prove results about zero-knowledge.

1.3 Main Technical Ideas

We now discuss some of our main technical ideas. One strength of our work is that our proofs are quite crisp.

Refuting the Oracle Derandomization Hypothesis. For contradiction, suppose that the Oracle Derandomization Hypothesis is true. In particular, suppose there is an efficient algorithm G that maps truth tables of oracles $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$ to truth tables of functions $f : \{0,1\}^n \to \{0,1\}$ such that f is hard even with oracle access to $\mathcal{O}$.

At first glance, it seems difficult to actually make use of the power of G. The reason is as follows. Imagine we instead got oracle access to a G' that is constructed by setting $G(\mathcal{O}) = f$ for a uniformly random f. It turns out that, if one believes that $\mathsf{P} = \mathsf{BPP}$, then oracle access to G' is essentially useless. Hence, if our argument is black-box in G, it must somehow distinguish between oracle access to G and G'. This seems quite difficult since G' does satisfy the correctness guarantee that $G'(x)$ requires large x-oracle circuits on all but an exponentially small fraction of all x (since it outputs a uniformly random string).

We overcome this by using *non-determinism* to effectively make exponentially many queries to G. In particular, we will use G to get a certificate for the hardness of a uniformly random function $f : \{0,1\}^n \to \{0,1\}$ as follows: we will non-deterministically guess an $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$ and $w \in \{0,1\}^{2n}$ such that $G(\mathcal{O}) = f \oplus \mathcal{O}|_w$, where $\mathcal{O}|_w$ denotes the restriction of $\mathcal{O}$ on its first $2n$ inputs to w. Observe that if $f \oplus \mathcal{O}|_w$ requires large $\mathcal{O}$-oracle circuits, then it must be the case that f requires large circuits (without an $\mathcal{O}$ oracle). Hence, if the algorithm finds such $\mathcal{O}$ and w, it can be sure that f is hard.

[9] Their assumption is the existence of a demi-bit generator [Rud97]. The non-existence of NP-constructive natural properties is a special case of the existence of a demi-bit generator, which makes their assumption weaker than ours. However, we note that our proofs of Corollary 1.8 and Corollary 1.6 easily generalizes to work with the demi-bit generator assumption, as discussed in [RWZ25].

It remains to show that such $\mathcal{O}$ and w exist for random f. The key idea comes from Lautemann's proof [Lau83] that BPP is in the polynomial hierarchy! Lautemann uses a "parity shift lemma" that says if $A \subseteq \{0,1\}^N$ has density p, then a random set $B \subseteq \{0,1\}^N$ of size $\mathsf{poly}(\frac{N}{p})$ satisfies

$$\{0,1\}^N = A \oplus B = \{x \oplus y : x \in A, y \in B\}$$

with high probability. In our setting:

- A corresponds to truth tables f for which no certificate pair $(\mathcal{O}, w)$ exists
- B corresponds to the truth tables of subfunctions of a uniformly random $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$.
- The fact that $A \oplus B = \{0,1\}^N$ corresponds to the fact that every possible output of $G(\mathcal{O})$ leads to a certificate for some $f \in A$, which contradicts that the functions in A lack a certificate.

We note our use of the parity shift lemma is reminiscent of the "drag-along principle" in [RR97].

Constructing Effectively Zero-Knowledge Proofs. An immediate consequence of our result on proof complexity generators (Corollary 1.8) is that there is a non-uniform sequence ψ_λ of unsatisfiable formulas such that "ψ_λ is unsatisfiable" lacks a short proof in every proof system.

We combine these ψ_λ with a construction of Feige-Lapidot-Shamir [FLS90], which works as follows. The prover, given a formula φ and a witness w, will give a truly non-interactive, perfectly sound, "witness indistinguishable" [FS90] proof π that "either φ is satisfiable or ψ_λ is satisfiable." Roughly, the witness indistinguishability guarantee says that one cannot tell if π is generated using a satisfying assignment for φ or a satisfying assignment for ψ_λ.

Thus, in particular, the "witness indistinguishability" guarantee implies that the prover is zero-knowledge if ψ_λ has a satisfying assignment. Specifically, one can simulate generating π without knowing a witness to φ as follows: generate π using the satisfying assignment for ψ_λ. Witness indistinguishability says that this simulated π is indistinguishable from honestly generating π using a satisfying assignment for φ.

Hence, if ψ_λ is satisfiable, then the prover is zero-knowledge. Now, in truth, ψ_λ is not satisfiable. But, because "ψ_λ is unsatisfiable" lacks a short proof in any proof system, we get the desired guarantee that it is hard to rule out that our prover is zero-knowledge in any proof system.

On the other hand, the verifier is perfectly sound because ψ_λ is, in fact, unsatisfiable.

2 Preliminaries

Unless otherwise specified, we let $N = 2^n$. The *density* of a set $A \subseteq \{0,1\}^n$ is $\frac{|A|}{2^n}$. Let $\{0,1\}^{\leq n}$ denote the set of binary strings of length at most n. For a language L, let $L|_n \in \{0,1\}^{2^n}$ denote the truth table of L restricted to n-bit inputs.

2.1 Proof Systems

In this paper, we use the following notion for a proof system following Cook and Reckhow [CR79].

Definition 2.1 (Proof System [CR79])**.** A *proof system for a language* L is a uniform deterministic polynomial time algorithm $V : \{0,1\}^\star \times \{0,1\}^\star \to \{0,1\}$ with the property that if $V(x,\pi) = 1$, then $x \in L$. We say that there is an *ℓ-length V-proof of x* if $V(x,\pi) = 1$ for some $\pi \in \{0,1\}^{\leq \ell}$.

The definition of a *non-uniform proof system* is the same except V can be a non-uniform deterministic polynomial-time algorithm. Unless otherwise specified, a proof system refers to the uniform definition.

For our purposes, we do not require completeness of a proof system V.

Throughout the paper, we assume we have fixed a particular language L sufficiently powerful enough to decide the truth of the statements we need to prove. For instance, $L = \mathsf{HALT}$ suffices. Unless otherwise specified, all proof systems we consider are proof systems for this fixed sufficiently powerful L.

We usually put statements being proved in double quotes and write an English language or mathematical description of the statement (we implicitly assume an unspecified but sufficiently "nice" encoding of these statements as instances of L).

We note that even powerful logical systems like ZFC (Zermelo Fraenkel with Choice, the standard axioms in mathematics) can be viewed as proof systems for a language L. For example, if $L = \mathsf{HALT}$, then one can consider the V with $V(x, \pi) = 1$ if and only if π is a ZFC-proof that $x \in \mathsf{HALT}$.

2.2 Circuit Complexity

In this paper, we consider circuits with NOT gates and fan-in-two AND/OR gates. The *size* of a circuit is the number of AND/OR gates in the circuit. For a function $f : \{0,1\}^n \to \{0,1\}$, let $\mathsf{CC}(f)$ denote the size of the smallest circuit for f. We say that f *has no size s circuit* if $\mathsf{CC}(f) > s$.

For a function $\mathcal{O} : \{0,1\}^m \to \{0,1\}$, we will also consider $\mathcal{O}$-oracle circuits where one is additionally allowed gates that take m-inputs and compute the $\mathcal{O}$ function. The size of an oracle circuit is the number of AND/OR gates plus m times the number of oracle gates. Let $\mathsf{CC}(f \mid \mathcal{O})$ denote the size of the smallest $\mathcal{O}$-oracle circuit for f.

2.3 Natural Proofs

Now we give the more general definition of NP-constructive natural properties.

Definition 2.2 ((infinitely often) NP-constructive natural properties against $\mathsf{SIZE}[s]$ [RR97; Rud97])**.** Let $s = s(n)$. A non-deterministic polynomial-time algorithm A is a (respectively, infinitely often) NP-*constructive natural property against* $\mathsf{SIZE}[s]$ if for all sufficiently large n (respectively, infinitely many n) we have that

- $A(T)$ accepts at least half[10] of all 2^n-bit truth tables T
- $A(T)$ rejects if T has a circuit of size at most $s(n)$.

Rudich conjectured [Rud97] there are no NP-constructive natural properties against $\mathsf{SIZE}[s]$ for any $s = n^{\omega(1)}$, and supported this conjecture using a new but plausible cryptographic assumption. We note that this assumption becomes weaker as one chooses larger s.

2.4 Cryptography

We recall the definition of computational indistinguishability in both the non-asymptotic and asymptotic context.

Definition 2.3 (Computational Indistinguishability)**.** Let $\epsilon \in \mathbb{R}$, and let $\mathcal{D}$ and $\mathcal{D}'$ be distributions on binary strings. We say $\mathcal{D}$ and $\mathcal{D}'$ are ϵ-*computationally indistinguishable* (written $\mathcal{D} \approx_\epsilon \mathcal{D}'$) if for every circuit A of size at most $\frac{1}{\epsilon}$ we have that

$$\left| \Pr_{x \leftarrow \mathcal{D}}[A(x) = 1] - \Pr_{x \leftarrow \mathcal{D}'}[A(x) = 1] \right| \leq \epsilon.$$

The asymptotic definition is analogous: Let $\epsilon : \mathbb{N} \to \mathbb{R}$ and let $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ and $\mathcal{D}' = \{D'_n\}_{n \in \mathbb{N}}$ denote sequences of distributions. We say $\mathcal{D}$ and $\mathcal{D}'$ are ϵ-computationally indistinguishable (written $\mathcal{D} \approx_\epsilon \mathcal{D}'$) if for all $n \in \mathbb{N}$ and every circuit A of size at most $\frac{1}{\epsilon(n)}$ we have that

$$\left| \Pr_{x \leftarrow D_n}[A(x) = 1] - \Pr_{x \leftarrow D'_n}[A(x) = 1] \right| \leq \epsilon(n).$$

Next, we recall the definition of a non-interactive witness indistinguishable proof system (NIWI) [FS90; DN07; BOV07] for SAT. (In this paper, whenever we refer to NIWIs, we refer to NIWIs for SAT.)

We note that the definition below differs mildly from the usual definition in that we require that ϵ be efficiently computable.

[10] For readers familiar with natural properties, here we have set the largeness density to be half for simplicity.

Definition 2.4 (Non-Interactive Witness Indistinguishable Proof (NIWI))**.** A *non-interactive witness indistinguishable proof system* is a tuple of algorithms (**NIWI.Prove**, **NIWI.Verify**) satisfying all of the following properties:

- **NIWI.Prove**(formula φ, satisfying assignment w, security parameter 1^λ) is a uniform randomized polynomial time algorithm that outputs a binary string (usually denoted π).
- **NIWI.Verify**(formula φ, purported proof π) is a uniform deterministic[11] polynomial-time algorithm that outputs either accept or reject.
- **Functionality:** For all formulas φ with $\varphi(w) = 1$ and all λ

$$\Pr[\textbf{NIWI.Verify}(\varphi, \textbf{NIWI.Prove}(\varphi, x, 1^\lambda)) = 1] = 1.$$

- **Perfect Soundness:** **NIWI.Verify**(φ, π) rejects on all unsatisfiable φ and all π.
- **Security (Witness Indistinguishability):** There is a polynomial-time computable negligible function ϵ such that for all formulas φ with $\varphi(w) = \varphi(w') = 1$, we have that

$$\textbf{NIWI.Prove}(\varphi, w, 1^\lambda) \approx_{\epsilon(\lambda)} \textbf{NIWI.Prove}(\varphi, w', 1^\lambda).$$

NIWIs exist under standard assumptions [BOV07; GOS12; BP15].

2.5 Probability Theory

We use the following lemma from measure theory.

Lemma 2.5 (Borel-Cantelli Lemma)**.** *Let $\{E_n\}_{n\in\mathbb{N}}$ be a collection of events. Assume $\sum_{n\in\mathbb{N}} \Pr[E_n]$ is finite. Then with probability one, only a finite number of events E_n occur.*

A simple consequence of the Borel-Cantelli lemma is the following.

Proposition 2.6. *Let $\{E_{m,n}\}_{m,n\in\mathbb{N}}$ be a collection of events. Assume that $\sum_{n\in\mathbb{N}} \sum_{m<g(n)} \Pr[E_{m,n}]$ is finite for some function $g(n) = \omega(1)$. Then*

$$\Pr[\textit{for all } m, \textit{ only finitely many } E_{m,n} \textit{ occur}] = 1.$$

Proof. Let G_n be the event that $E_{m,n}$ occurs for some $m \le g(n)$. By the Borel-Cantelli lemma, with probability one, only finitely many G_n occur.

On the other hand, if for some m, infinitely many $E_{m,n}$ occur, then infinitely many G_n occur (using that $g = \omega(1)$). The proposition follows. □

In particular, one setting of parameters gives the following lemma.

Lemma 2.7. *Let $\{E_{m,n}\}_{m,n\in\mathbb{N}}$ be a collection of events. Assume that for all m, there exists an integer n_m such that*

$$\Pr[E_{m,n}] \le \frac{1}{n^2}$$

for all $n \ge n_m$. Then

$$\Pr[\textit{for all } m, \textit{ only finitely many } E_{m,n} \textit{ occur}] = 1$$

Proof. Define $g : \mathbb{N} \to \mathbb{N}$ by $g(n) = \min\{m \le \sqrt{n} : n_m > n\} \cup \{\sqrt{n}\}$. Observe that $g = \omega(1)$, as desired. Then we have that

$$\sum_{n\in\mathbb{N}} \sum_{m<g(n)} \Pr[E_{m,n}] \le \sum_{n\in\mathbb{N}} \frac{\sqrt{n}}{n^2} = \sum_{n\in\mathbb{N}} \frac{1}{n^{1.5}}.$$

is finite. Then the result follows from Proposition 2.6. □

[11] We assume the verifier is deterministic. A randomized verifier can always be made deterministic if Promise-P = Promise-BPP.

3 Refuting the Oracle Derandomization Hypothesis

We begin by stating the Oracle Derandomization Hypothesis of Fortnow and Santhanam [FS11].

Hypothesis 3.1 (Oracle Derandomization Hypothesis [FS11])**.** *There is a constant $\epsilon > 0$ such that for all constants $\delta > 0$, there is a polynomial-time computable function G_δ with the following two properties:*

- *N^δ **shrinking:** G_δ maps N-bit strings to $\lfloor N^\delta \rfloor$-bit strings*
- *$G(x)$ **is exponentially non-deterministically hard relative to** x**:** for all x, the non-deterministic x-oracle circuit complexity of $G_\delta(x)$ is always at least $N^{\delta\epsilon}$ (where we interpret x and $G_\delta(x)$ as the truth tables of Boolean functions in the natural way by potentially padding with zeros).*

We will refute this hypothesis using the non-existence of NP-constructive natural properties. Our proof uses the "parity shift lemma" from Lautemann's proof [Lau83] that BPP is in the polynomial hierarchy. For two sets A and B, let $A \oplus B$ denote the *parity set* given by $\{a \oplus b : a \in A, b \in B\}$.

Lemma 3.2 ("Parity Shift Lemma" [Lau83])**.** *For any set $A \subseteq \{0,1\}^n$,*

$$\Pr_{x_1,\ldots,x_k \leftarrow \{0,1\}^n}[\{0,1\}^n = A \oplus \{x_1,\ldots,x_k\}] \geq 1 - e^{-k\frac{|A|}{2^n}+n}.$$

Proof. The proof is a union bound argument. Fix an arbitrary $y \in \{0,1\}^n$. Observe

$$\Pr_{x \leftarrow \{0,1\}^n}[y \in A \oplus \{x\}] = \frac{|A|}{2^n}.$$

Therefore, the probability that $y \notin A \oplus \{x_1,\ldots,x_k\}$ is at most

$$(1 - \frac{|A|}{2^n})^k \leq e^{-k\frac{|A|}{2^n}}.$$

Union bounding over all $N = 2^n \leq e^n$ possibilities for $y \in \{0,1\}^n$ proves the lemma. □

We are now ready to prove our theorem.

Theorem 3.3. *Assume there is non-deterministic polynomial-time algorithm G with both the following properties for all (respectively, infinitely many) $n \in \mathbb{N}$*

- Correctness: *Let $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$. If $G(\mathcal{O}) \neq ?$, then $f = G(\mathcal{O})$ is the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{CC}(f|\mathcal{O}) > s(n)$*
- Utility: *$G(\mathcal{O}) \neq ?$ for at least a 2^{-N}-fraction of all $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$.*

Then there is a (respectively, infinitely often) NP-constructive natural property against $\mathsf{SIZE}[s(n) - 3n - 3]$.

Proof. We give the proof for the almost everywhere version. The infinitely often proof is similar.

Let A be the following non-deterministic polynomial-time algorithm:

Non-deterministic algorithm A

Given the truth table of $f : \{0,1\}^n \to \{0,1\}$

1. Non-deterministically guess $\mathcal{O} : \{0,1\}^{3n} \to \{0,1\}$ and $w \in \{0,1\}^{2n}$
2. Accept if $G(\mathcal{O}) = f \oplus \mathcal{O}|_w$, where $\mathcal{O}|_w : \{0,1\}^n \to \{0,1\}$ is the restriction of the first $2n$ inputs of $\mathcal{O}$ to w

We will show that A is a NP-constructive natural property against $\mathsf{SIZE}[s(n) - 3n - 3]$. There are two things to show. First, that it rejects all low complexity truth tables, and second that it accepts random truth tables.

We do these in order. Suppose A accepts f. Then $G(\mathcal{O}) = f \oplus \mathcal{O}|_w$ for some $\mathcal{O}$ and w. By the guarantee on G, we must have that $\mathsf{CC}(G(\mathcal{O}) \mid \mathcal{O}) > s(n)$. Since $\mathcal{O}|_w$ is computable by an $\mathcal{O}$-oracle circuit of size $3n$, we must have that $\mathsf{CC}(f) > s(n) - 3n - 3$ (the extra 3 comes from computing the parity between f and $\mathcal{O}|_w$).

It remains to show that A accepts at least half of all f. Let Bad be the set of $f : \{0,1\}^n \to \{0,1\}$ that A does not accept. For contradiction, suppose that $|Bad| > 2^N/2$. Then we get the following contradiction for sufficiently large n

$$\begin{aligned}
0 &= \Pr_{\mathcal{O}:\{0,1\}^{3n}\to\{0,1\}}[G(\mathcal{O}) \in Bad \oplus \{\mathcal{O}|_w : w \in \{0,1\}^{2n}\}] \\
&\geq \Pr_{\mathcal{O}:\{0,1\}^{3n}\to\{0,1\}}[\{0,1\}^N = \{\mathcal{O}|_w : w \in \{0,1\}^{2n}\} \text{ and } G(\mathcal{O}) \neq ?] - (1 - 2^{-n}) \\
&\geq \Pr_{\mathcal{O}:\{0,1\}^{3n}\to\{0,1\}}[\{0,1\}^N = \{\mathcal{O}|_w : w \in \{0,1\}^{2n}\}] - (1 - 2^{-n}) \\
&\geq 1 - \exp(-N^2|Bad|/2^N + N) - (1 - 2^{-N}) \\
&\geq 2^{-N} - \exp(-N^2/2 + N) > 0
\end{aligned}$$

where the first line is by the definition of Bad, the second line is by the correctness of G, the third line is by the union bound, and the fourth line is by the parity shift lemma. □

3.1 Proof Complexity Generators

The notion of proof complexity generators was first considered (independently) in the works of Alekhnovich, Ben-Sasson, Razborov, and Wigderson [ABRW04] and Krajíček [Kra01a].

Definition 3.4 (Proof Complexity Generator [ABRW04; Kra01a])**.** Let V be a proof system. Let $G = \{G_N : \{0,1\}^{M(N)} \to \{0,1\}^N\}$ be a family of length-increasing functions computable by polynomial-size circuits. We say that G is a *proof complexity generator against* V if for every polynomial ℓ, we have that

$$\Pr_{y\in\{0,1\}^N}[\exists\ \ell(N)\text{-length } V\text{-proof that ``}y \text{ is not in the range of } G_N\text{''}] = 0$$

for all sufficiently large N.

Krajíček first raised the possibility that there could be a single proof complexity generator that is simultaneously secure against *all* proof systems.

Conjecture 3.5 (Krajíček [Kra04b] (see also [Kra22, Conjecture 1.1]))**.** *There is a proof complexity generator against all proof systems.*

We give, to the best of our knowledge, the first candidate of such a proof complexity generator with a security reduction to a plausible complexity assumption. Our construction is the natural oracle generalization of the truth table generator (which maps circuits to their truth tables).

Oracle Truth Table Generator

Parameters: an output length $N = 2^n$, a size parameter s, and an oracle $\mathcal{O} : \{0,1\}^m \to \{0,1\}$

The generator $G_N^{\mathsf{CC}}[s, \mathcal{O}] : \{0,1\}^{\tilde{O}(s(N))} \to \{0,1\}^N$ is given by

1. Interpret the input as the description of an oracle circuit C of size at most s mapping n-bits to one bit

2. Output the N-bit truth table of the circuit $C^{\mathcal{O}}$.

Assuming the non-existence of NP-constructive natural properties, one can show that the oracle truth table generator on uniformly random $\mathcal{O}$ is a proof complexity generator secure against all (uniform) proof systems.

Theorem 3.6. *Assume there are no infinitely often* NP*-constructive natural properties against* $\mathsf{SIZE}[s(n)+3n+3]$*. For all* n*, let* $\mathcal{O}_n : \{0,1\}^{3n} \to \{0,1\}$ *be a uniformly random function. Then, with probability one,* $G = \{G_N^{\mathsf{CC}}[s(N), \mathcal{O}_n]\}$ *is a proof complexity generator against all proof systems (as long as* s *is small enough that it is length increasing).*

Proof. For succinctness, we write $G_N = G_N^{\mathsf{CC}}[s(N), \mathcal{O}_n]$. We let $\mathcal{O}$ denote the collection of all $\mathcal{O}_n$.

For a (uniform) proof system V and a polynomial ℓ, let $E_{V,\ell,N}$ be the event that there exists an $\ell(N)$-length V-proof that "y is outside the range of G_N" for some $y \in \{0,1\}^N$.

Because proof systems correspond to zero-error non-deterministic algorithms, by Theorem 3.3, for any proof system V and any polynomial ℓ, we have that

$$\Pr_{\mathcal{O}}[E_{V,\ell,N}] < 2^{-N}.$$

Then, by Lemma 2.7 (essentially the Borel-Cantelli lemma), we have that

$$1 = \Pr_{\mathcal{O}}[\text{for all } V \text{ and } \ell,\ E_{V,\ell,N} \text{ occurs for only finitely many } N]$$

This proves the theorem. □

We remark that our proof generalizes straightforwardly to the setting of conditional time-bounded Kolmogorov complexity, if one replaces the natural properties assumption with the analogous assumption for time-bounded Kolmogorov complexity.

3.2 Average-Case Hardness of Range Avoidance

Now we show the average-case hardness of Avoid. Indeed, the hard average-case instances are just the oracle truth table generator on random oracles.

Corollary 3.7. *Assume there are no infinitely often* NP*-constructive natural properties against* $\mathsf{SIZE}[s(n)+3n+3]$*. Then every non-deterministic polynomial-time zero-error average-case*[12] *algorithm* A *for* Avoid *has*

$$\Pr_{\mathcal{O}_n:\{0,1\}^{3n}\to\{0,1\}}[A(G_N^{\mathsf{CC}}[s(n), \mathcal{O}_n]) \neq ?] < 2^{-N}.$$

Proof. By the assumption on A and the construction of G_N^{CC}, when $A(G_N^{\mathsf{CC}}[s(n), \mathcal{O}_n]) = y \neq ?$, it must be that $\mathsf{CC}(y \mid \mathcal{O}_n) > s(n)$. Thus, the corollary follows immediately from Theorem 3.3. □

4 On Truly Non-Interactive Zero-Knowledge

In this section, we define a new relaxation of zero-knowledge and construct it under plausible assumptions.

4.1 Definitions

First, we define an abstract notion of a prover.

Definition 4.1 (Prover)**.** A *prover* is a (potentially non-uniform) probabilistic polynomial-time algorithm $P(\cdot,\cdot,\cdot)$ that works as follows

[12] To be clear, this means that $A(x)$ can output ?, but whenever it outputs an answer it needs to be correct.

- **Inputs:** a formula φ, a satisfying assignment w to φ, and a security parameter 1^λ
- **Output:** a string π.

To make some technical aspects easier, we require that $|\pi| \geq \lambda \geq |\varphi|$. If P is a uniform algorithm, we say it is a *uniform prover*. If it is non-uniform, we say it is a *non-uniform prover*.

Note that this definition, by itself, does not require the prover to actually produce interesting output (for example, a prover may just output a sufficiently long string of zeros).

Definition 4.2 (Useful Prover)**.** We say a uniform prover (respectively, non-uniform prover) is *useful* if there exists a uniform (respectively, non-uniform) proof system V such that

$$\Pr_P[P(\varphi, w, 1^\lambda) \text{ outputs a } V\text{-proof that ``}\varphi \text{ is satisfiable''}] = 1$$

whenever $\varphi(w) = 1$ and $|\varphi| \leq \lambda$.[13]

Next, we recall the notion of a simulator [GMR89].

Definition 4.3 (Simulator)**.** Let $C(\varphi, w)$ be a probabilistic circuit that maps formula-witness-pairs to a string. We say C has an *s-size ϵ-secure simulator* if there is a probabilistic circuit Sim of size s such that

$$Sim_\lambda(\varphi) \approx_\epsilon C(\varphi, w).$$

for all formulas φ with $\varphi(w) = 1$.

Moreover, if P is a prover and $\lambda \in \mathbb{N}$, we say P_λ *has an s-size ϵ-secure simulator* if the probabilistic circuit $P_\lambda(\varphi, w) = P(\varphi, w, 1^\lambda)$ has an s-size ϵ-secure simulator.

We now restate our main definition with slightly different notation.

Definition 4.4 (Effectively Zero-Knowledge to V)**.** Let P be a prover, and let V be a proof system. We say P is *effectively zero-knowledge to V* if for some $t = \lambda^{\omega(1)}$ and $s = \mathsf{poly}(\lambda)$ the following holds for all λ:

$$\text{``}P_\lambda \text{ has no } s(\lambda)\text{-size } \frac{1}{t(\lambda)}\text{-secure simulator''}^{14}$$

has no $t(\lambda)$-length V-proof.

4.2 Our Construction

Our construction follows the approach suggested by Kuykendall and Zhandry [KZ20] building on Feige, Lapidot, and Shamir [FLS90]. The construction will make use of a non-interactive witness indistinguishable proof system (NIWI) [FS90; DN07; BOV07]. We assume a NIWI exists and fix it for the remainder of this section.

Our construction will be parameterized by a (potentially non-uniform) sequence $\{\varphi_\lambda\}$ of $\mathsf{poly}(\lambda)$-sized formulas indexed by $\lambda \in \mathbb{N}$.

Prover $P_{\{\varphi_\lambda\}}$

Given $\psi, w, 1^\lambda$:

1. Reject if $\psi(w) \neq 1$ or $|\psi| > \lambda$.
2. Output NIWI proof of "$\psi \vee \varphi_\lambda$ is satisfiable" with witness w and security parameter λ.[a]

[13] One might also consider cases where the prover is non-uniform but it outputs proofs in a uniform proof system, or vice-versa. We do not in this paper.

[14] We clarify a potential ambiguity in the notation. In the statement being proved $t(\lambda)$ and $s(\lambda)$ denote the specific numbers the corresponding functions evaluate to. We do not need to include in the statement any information about how to compute t and s on values other than λ. Similarly, P_λ denotes the specific circuit corresponding to $P(\varphi, w, 1^\lambda)$.

[a] If needed, pad this proof to length at least λ in any reasonable way.

Verifier $V_{\{\varphi_\lambda\}}$

Given a "statement" x and a "proof" π:

1. Accept if x encodes (according to our sufficiently nice encoding) "ψ is satisfiable" and π is a NIWI proof of "$\psi \vee \varphi_\lambda$ is satisfiable" for some $\lambda \leq |\pi|$

We note that we refer to $V_{\{\varphi_\lambda\}}$ as a verifier instead of a proof system. This is because it is a proof system if and only if all the φ_λ are unsatisfiable.

We also define the following related probabilistic circuit, which is parameterized by a $\lambda \in \mathbb{N}$ and a single formula φ of size at most λ.

Probabilistic Circuit $P[\varphi, \lambda]$

Given a formula ψ of size at most λ and w:

1. Reject if $\psi(w) \neq 1$.
2. Output NIWI proof of "$\varphi \vee \psi$ is satisfiable" with witness w and security parameter λ.[a]

[a] If needed, pad this proof to length at least λ in any reasonable way.

We now prove the key lemma regarding this construction.

Lemma 4.5. *There exist functions $\epsilon(\lambda) = \lambda^{-\omega(1)}$ and $s(\lambda) = \mathsf{poly}(\lambda)$ such that the following holds for every proof system V, every $\lambda \in \mathbb{N}$, and every formula φ of size at most λ. If there is an ℓ-length V-proof of the statement*

"$P[\varphi, \lambda]$ lacks an $s(\lambda)$-size $\epsilon(\lambda)$-secure simulator"

then there is a $\mathsf{poly}(\ell, \lambda)$-length proof that "$\varphi$ is unsatisfiable" in the proof system $V_{extended}$ (which is defined below and depends only on V and the choice of NIWI).

Proof. Let $\epsilon(\lambda) = \lambda^{-\omega(1)}$ be the polynomial-time computable function corresponding to the security of the NIWI. Let $s(\lambda) = \mathsf{poly}(\lambda)$ be a polynomial such that $s(\lambda) \geq |P[\varphi', \lambda]|$ for any φ' of size at most λ (note that $s(\lambda)$ is polynomial because the NIWI runs in polynomial time).

Fix any proof system V. Let $V_{extended}$ be the proof system corresponding to the following algorithm.

Proof System $V_{extended}$

On input x and $\pi = (1^\ell, \pi_0, \varphi')$:

1. Accept if π_0 is a V-proof of x
2. Accept if there is a $\lambda \in [\ell]$ such that all of the following hold:
   - φ' is of size at most λ,
   - x encodes (according to our choice of encoding of statements) the statement "φ' is unsatisfiable."
   - π is a V-proof of "$P[\varphi', \lambda]$ has no $s(\lambda)$-size $\epsilon(\lambda)$-secure simulator."[a]
3. Otherwise, reject

[a] This is where we need that ϵ is efficiently computable.

If $V_{extended}$ is indeed a proof system, the lemma follows immediately from the definition of $V_{extended}$. It remains to show the following claim.

Claim 4.6. *$V_{extended}$ is a proof system.*

Proof. Let L be the language that V is a proof system for. Let x be an arbitrary statement that $V_{extended}$ accepts. If x is accepted in step (1), then $x \in L$ by the soundness of V. Thus, we assume that x is accepted in step (2) with the values λ, π_0, and φ'.

By the soundness of V, we know $P[\varphi', \lambda]$ does not have an $s(\lambda)$-size $\epsilon(\lambda)$-secure simulator. On the other hand, if it was the case that $\varphi'(w) = 1$ for some w, then NIWI security guarantees that

$$Sim_\lambda(\psi') = P[\varphi', \lambda](\psi', w)$$

is an $\epsilon(\lambda)$-secure simulator for $P[\varphi', \lambda]$ of size $s(\lambda)$. Hence, φ' is unsatisfiable. □

□

Next, we make a definition for when a sequence of formulas is hard to prove unsatisfiable using a proof system.

Definition 4.7 (Formulas Hard to Prove Unsatisfiable)**.** Let $\{\varphi_\lambda\}$ be a sequence of $\mathsf{poly}(\lambda)$-sized formulas. Let V be a (potentially non-uniform) proof system. We say $\{\varphi_\lambda\}$ is *hard to prove unsatisfiable using V* if there is an $\ell = \lambda^{\omega(1)}$ such that for all λ there is no $\ell(\lambda)$-length $V_{extended}$-proof that φ_λ is unsatisfiable. *(We stress that the previous sentence talks about about $V_{extended}$-proofs.)*

We say that $\{\varphi_\lambda\}$ is *universally hard to prove unsatisfiable* if it is hard to prove unsatisfiable using every uniform proof system.

Note that this definition depends on the definition of $V_{extended}$, which depends on our choice of NIWI. Also note that this definition does not require that the formulas φ_λ actually be unsatisfiable. For example, any sequence of satisfiable formulas is universally hard to prove unsatisfiable.

Combining this definition with Lemma 4.5, we get the following lemma.

Lemma 4.8. *There exists a polynomial s such that the following holds. If a sequence of polynomial-sized formulas $\{\varphi_\lambda\}$ is hard to prove unsatisfiable using a proof system V, then $P_{\{\varphi_\lambda\}}$ is effectively zero-knowledge to V.*

Proof. Let $\epsilon = \lambda^{-\omega(1)}$, and $s = \lambda^{O(1)}$ be the parameters given by Lemma 4.5. Let $t(\lambda) = \lambda^{\omega(1)}$ be a sufficiently small superpolynomial function satisfying $t(\lambda) \leq \frac{1}{\epsilon(\lambda)}$.

Now fix some λ and suppose there is a $t(\lambda)$-length V-proof that

$$\text{“}P_\lambda \text{ has no } s(\lambda)\text{-size } t(\lambda)\text{- secure simulator.”}$$

Then by Lemma 4.5, there is a $\mathsf{poly}(t(\lambda))$-length $V_{extended}$-proof that “φ_λ is unsatisfiable.” But by assumption, any $V_{extended}$-proof that “φ_λ is unsatisfiable” has length at least $\lambda^{\omega(1)} > \mathsf{poly}(t(\lambda))$ by setting t to be a sufficiently small superpolynomial function. □

Finally, we get the following lemma if $\{\varphi_\lambda\}$ is universally hard to prove unsatisfiable.

Lemma 4.9. *If a sequence of polynomial-sized formulas $\{\varphi_\lambda\}$ is universally hard to prove unsatisfiable, then $P_{\{\varphi_\lambda\}}$ is effectively zero-knowledge to every (uniform) proof system V.*

Proof. This follows from Lemma 4.8 and universal hardness. □

4.3 Result

In this section, we construct a useful non-uniform prover and verifier that is effectively zero-knowledge to every (uniform) proof system.

Theorem 4.10. *Assume a NIWI exists and there is a non-uniform sequence $\{\varphi_\lambda\}$ of polynomial-size unsatisfiable formulas that are universally hard to prove unsatisfiable. Then there is a useful non-uniform prover that is effectively zero-knowledge to every (uniform) proof system.*

Proof. Let $P = P_{\{\varphi_\lambda\}}$ and $V = V_{\{\varphi_\lambda\}}$. V is sound because all the φ_λ are unsatisfiable and the NIWI is perfectly sound. P is effectively zero-knowledge to every uniform proof system by Lemma 4.9. □

We describe two (related) ways to construct such non-uniform sequences. The first is via a universally hard UNSAT distribution.

Definition 4.11 (Universally Hard UNSAT Distribution)**.** Let $\mathcal{D}_\lambda$ be a polynomial-time samplable distribution on formulas. We say that $\mathcal{D}_\lambda$ is a *universally hard distribution for* UNSAT if for all uniform proof systems V there is an $\ell = \lambda^{\omega(1)}$ such that

$$\Pr_{\varphi \leftarrow \mathcal{D}_\lambda}[\text{either } \varphi \text{ is satisfiable or there is an } \ell(\lambda)\text{-length } V\text{-proof that ``}\varphi \text{ is unsatisfiable''}] = o(\frac{1}{\lambda^2}).$$

A universally hard UNSAT distribution follows, for example, from the assumption that s-lower bound proofs are, say, $\frac{1}{\lambda^3}$-rare when $s \ll \frac{2^n}{n}$.

Sampling from a universally hard UNSAT distribution leads (with high probability) to a sequence of unsatisfiable formulas universally hard to prove unsatisfiable.

Proposition 4.12. *Assume there exists a universally hard distribution $\mathcal{D}_\lambda$ for* UNSAT*. Then there is a non-uniform sequence of polynomial-sized unsatisfiable formulas $\{\varphi_\lambda\}$ that is universally hard to prove unsatisfiable.*

Proof. We will do this by a probabilistic argument. In particular, we will set $\varphi_\lambda \leftarrow D_\lambda$ and show that, with positive probability, it has the desired properties.

For a polynomial ℓ and proof system V, let $E_{V,\ell,\lambda}$ be the event that either φ_λ is satisfiable or there is an $\ell(\lambda)$-length V-proof that "φ_λ is unsatisfiable."

By assumption, we have that for every proof system V and every polynomial ℓ

$$\Pr[E_{V,\ell,\lambda}] = o(\frac{1}{\lambda^2}).$$

Hence, by Lemma 2.7 (essentially the Borel-Cantelli lemma), we get that

$$\Pr[\text{For all } V \text{ and } \ell\text{, only finitely many } E_{V,\ell,\lambda} \text{ occur}] = 1,$$

proving the proposition (replace the at most finitely many satisfiable formulas with the trivial unsatisfiable formula). □

One can also obtain such a sequence from a proof complexity generator against all proof systems.

Proposition 4.13. *If there is a proof complexity generator against all proof systems, then there is a non-uniform sequence of unsatisfiable formulas universally hard to prove unsatisfiable.*

Proof. For each input length λ, let y_λ be a string outside the range of the generator. Let φ_λ be the formula that is satisfiable if and only if y_λ is in the range of the generator. The proposition follows from the security of the proof complexity generator. □

Consequently, we get the following result.

Theorem 4.14. *Assume a NIWI exists. Assume there is no* NP*-constructive natural property against* SIZE$[2^n/n^2]$*. Then there is a useful non-uniform prover that is effectively zero-knowledge to every proof system.*

Proof. This follows from Proposition 4.13, Theorem 3.6, and Theorem 4.10. □

References

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. "Pseudorandom Generators in Propositional Proof Complexity". In: *SIAM J. Comput.* 34.1 (2004), pp. 67–88. DOI: 10.1137/S0097539701389944. URL: https://doi.org/10.1137/S0097539701389944 (cit. on pp. 4, 12).

[Aut25] Anonymous Author. "Godel in Cryptography: Effectively Zero-Knowledge Proofs for NP with No Interaction, No Setup, and Perfect Soundness". In: *Electron. Colloquium Comput. Complex.* TR25-095 (2025). ECCC: TR25-095. URL: https://eccc.weizmann.ac.il/report/2025/095 (cit. on p. 6).

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. Ed. by Janos Simon. ACM, 1988, pp. 103–112. DOI: 10.1145/62212.62222. URL: https://doi.org/10.1145/62212.62222 (cit. on p. 5).

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (im)possibility of obfuscating programs". In: *J. ACM* 59.2 (2012), 6:1–6:48. DOI: 10.1145/2160158.2160159. URL: https://doi.org/10.1145/2160158.2160159 (cit. on pp. 4, 6).

[BL18] Nir Bitansky and Huijia Lin. "One-Message Zero Knowledge and Non-malleable Commitments". In: *TCC 2018*. Ed. by Amos Beimel and Stefan Dziembowski. Vol. 11239. Lecture Notes in Computer Science. Springer, 2018, pp. 209–234. DOI: 10.1007/978-3-030-03807-6_8. URL: https://doi.org/10.1007/978-3-030-03807-6%5C_8 (cit. on p. 7).

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. "Derandomization in Cryptography". In: *SIAM J. Comput.* 37.2 (2007), pp. 380–400. DOI: 10.1137/050641958. URL: https://doi.org/10.1137/050641958 (cit. on pp. 5–7, 9, 10, 14).

[BP04] Boaz Barak and Rafael Pass. "On the Possibility of One-Message Weak Zero-Knowledge". In: *TCC 2004*. Ed. by Moni Naor. Vol. 2951. Lecture Notes in Computer Science. Springer, 2004, pp. 121–132. DOI: 10.1007/978-3-540-24638-1_7. URL: https://doi.org/10.1007/978-3-540-24638-1%5C_7 (cit. on p. 7).

[BP15] Nir Bitansky and Omer Paneth. "ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation". In: *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Springer, 2015, pp. 401–427. DOI: 10.1007/978-3-662-46497-7_16. URL: https://doi.org/10.1007/978-3-662-46497-7%5C_16 (cit. on pp. 6, 10).

[BSMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. "Noninteractive Zero-Knowledge". In: *SIAM J. Comput.* 20.6 (1991), pp. 1084–1118. DOI: 10.1137/0220068. URL: https://doi.org/10.1137/0220068 (cit. on p. 5).

[CHLR23] Yeyuan Chen, Yizhi Huang, Jiatu Li, and Hanlin Ren. "Range Avoidance, Remote Point, and Hard Partial Truth Table via Satisfying-Pairs Algorithms". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. STOC 2023. Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 1058–1066. ISBN: 9781450399135. DOI: 10.1145/3564246.3585147. URL: https://doi.org/10.1145/3564246.3585147 (cit. on p. 4).

[CHR24] Lijie Chen, Shuichi Hirahara, and Hanlin Ren. "Symmetric Exponential Time Requires Near-Maximum Circuit Size". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. STOC 2024. Vancouver, BC, Canada: Association for Computing Machinery, 2024, pp. 1990–1999. ISBN: 9798400703836. DOI: 10.1145/3618260.3649624. URL: https://doi.org/10.1145/3618260.3649624 (cit. on p. 4).

[CL24] Yilei Chen and Jiatu Li. "Hardness of Range Avoidance and Remote Point for Restricted Circuits via Cryptography". 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. 620–629. DOI: 10.1145/3618260.3649602. URL: https://doi.org/10.1145/3618260.3649602 (cit. on p. 4).

[CR79] Stephen A. Cook and Robert A. Reckhow. "The Relative Efficiency of Propositional Proof Systems". In: *J. Symb. Log.* 44.1 (1979), pp. 36–50. DOI: 10.2307/2273702. URL: https://doi.org/10.2307/2273702 (cit. on pp. 4, 8).

[DN07] Cynthia Dwork and Moni Naor. "Zaps and Their Applications". In: *SIAM J. Comput.* 36.6 (2007), pp. 1513–1543. DOI: 10.1137/S0097539703426817. URL: https://doi.org/10.1137/S0097539703426817 (cit. on pp. 9, 14).

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)". In: *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*. IEEE Computer Society, 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549. URL: https://doi.org/10.1109/FSCS.1990.89549 (cit. on pp. 5, 8, 14).

[FS11] Lance Fortnow and Rahul Santhanam. "Infeasibility of instance compression and succinct PCPs for NP". In: *J. Comput. Syst. Sci.* 77.1 (2011), pp. 91–106. DOI: 10.1016/J.JCSS.2010.06.007. URL: https://doi.org/10.1016/j.jcss.2010.06.007 (cit. on pp. 2, 11).

[FS90] Uriel Feige and Adi Shamir. "Witness Indistinguishable and Witness Hiding Protocols". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. Ed. by Harriet Ortiz. ACM, 1990, pp. 416–426. DOI: 10.1145/100216.100272. URL: https://doi.org/10.1145/100216.100272 (cit. on pp. 5, 6, 8, 9, 14).

[GGNS23] Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. "Range Avoidance for Constant Depth Circuits: Hardness and Algorithms". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA*. Ed. by Nicole Megow and Adam D. Smith. Vol. 275. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 65:1–65:18. DOI: 10.4230/LIPICS.APPROX/RANDOM.2023.65. URL: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2023.65 (cit. on p. 4).

[GLW25] Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. "Range Avoidance for Low-Depth Circuits and Connections to Pseudorandomness". In: *ACM Trans. Comput. Theory* (Mar. 2025). Just Accepted. ISSN: 1942-3454. DOI: 10.1145/3718745. URL: https://doi.org/10.1145/3718745 (cit. on p. 4).

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)". In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*. Ed. by Robert Sedgewick. ACM, 1985, pp. 291–304. DOI: 10.1145/22145.22178. URL: https://doi.org/10.1145/22145.22178 (cit. on p. 2).

[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. DOI: 10.1137/0218012. URL: https://doi.org/10.1137/0218012 (cit. on pp. 5, 14).

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Prove all NP-Statements in Zero-Knowledge, and a Methodology of Cryptographic Protocol Design". In: *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*. Ed. by Andrew M. Odlyzko. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 171–185. DOI: 10.1007/3-540-47721-7_11. URL: https://doi.org/10.1007/3-540-47721-7%5C_11 (cit. on p. 5).

[GO94] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *J. Cryptol.* 7.1 (1994), pp. 1–32. DOI: 10.1007/BF00195207. URL: https://doi.org/10.1007/BF00195207 (cit. on p. 5).

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "New Techniques for Noninteractive Zero-Knowledge". In: *J. ACM* 59.3 (2012), 11:1–11:35. DOI: 10.1145/2220357.2220358. URL: https://doi.org/10.1145/2220357.2220358 (cit. on pp. 6, 10).

[HW15] Pavel Hubácek and Daniel Wichs. "On the Communication Complexity of Secure Function Evaluation with Long Output". In: *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*. Ed. by Tim Roughgarden. ACM, 2015, pp. 163–172. DOI: 10.1145/2688073.2688105. URL: https://doi.org/10.1145/2688073.2688105 (cit. on p. 5).

[ILW23] Rahul Ilango, Jiatu Li, and R. Ryan Williams. "Indistinguishability Obfuscation, Range Avoidance, and Bounded Arithmetic". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*. Ed. by Barna Saha and Rocco A. Servedio. ACM, 2023, pp. 1076–1089. DOI: 10.1145/3564246.3585187. URL: https://doi.org/10.1145/3564246.3585187 (cit. on pp. 4, 6).

[IW97] Russell Impagliazzo and Avi Wigderson. "$P = BPP$ if E Requires Exponential Circuits: Derandomizing the XOR Lemma". In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*. Ed. by Frank Thomson Leighton and Peter W. Shor. ACM, 1997, pp. 220–229. DOI: 10.1145/258533.258590. URL: https://doi.org/10.1145/258533.258590 (cit. on p. 2).

[IY87] Russell Impagliazzo and Moti Yung. "Direct Minimum-Knowledge Computations". In: *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*. Ed. by Carl Pomerance. Vol. 293. Lecture Notes in Computer Science. Springer, 1987, pp. 40–51. DOI: 10.1007/3-540-48184-2_4. URL: https://doi.org/10.1007/3-540-48184-2%5C_4 (cit. on p. 5).

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability obfuscation from well-founded assumptions". In: *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM, 2021, pp. 60–73. DOI: 10.1145/3406325.3451093. URL: https://doi.org/10.1145/3406325.3451093 (cit. on p. 4).

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability Obfuscation from LPN over *nmathbb{F}_p*, DLIN, and PRGs in NC^0". In: *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13275. Lecture Notes in Computer Science. Springer, 2022, pp. 670–699. DOI: 10.1007/978-3-031-06944-4_23. URL: https://doi.org/10.1007/978-3-031-06944-4%5C_23 (cit. on p. 6).

[Kha22] Erfan Khaniki. "Nisan-Wigderson Generators in Proof Complexity: New Lower Bounds". In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 17:1–17:15. DOI: 10.4230/LIPICS.CCC.2022.17. URL: https://doi.org/10.4230/LIPIcs.CCC.2022.17 (cit. on p. 4).

[KKMP21] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos H. Papadimitriou. "Total Functions in the Polynomial Hierarchy". In: *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*. Ed. by James R. Lee. Vol. 185. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 44:1–44:18. DOI: 10.4230/

LIPICS.ITCS.2021.44. URL: https://doi.org/10.4230/LIPIcs.ITCS.2021.44 (cit. on pp. 3, 4).

[Kor21] Oliver Korten. "The Hardest Explicit Construction". In: *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021, pp. 433–444. DOI: 10.1109/FOCS52979.2021.00051. URL: https://doi.org/10.1109/FOCS52979.2021.00051 (cit. on p. 4).

[KP24] Oliver Korten and Toniann Pitassi. "Strong vs. Weak Range Avoidance and the Linear Ordering Principle". In: *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*. 2024, pp. 1388–1407. DOI: 10.1109/FOCS61266.2024.00089 (cit. on p. 4).

[KP89] Jan Krajíček and Pavel Pudlák. "Propositional Proof Systems, the Consistency of First Order Theories and the Complexity of Computations". In: *J. Symb. Log.* 54.3 (1989), pp. 1063–1079. DOI: 10.2307/2274765. URL: https://doi.org/10.2307/2274765 (cit. on p. 5).

[Kra01a] Jan Krajíček. "On the weak pigeonhole principle". eng. In: *Fundamenta Mathematicae* 170.1-2 (2001), pp. 123–140. URL: http://eudml.org/doc/282141 (cit. on pp. 4, 12).

[Kra01b] Jan Krajíček. "Tautologies from pseudo-random generators". In: *Bull. Symb. Log.* 7.2 (2001), pp. 197–212. DOI: 10.2307/2687774. URL: https://doi.org/10.2307/2687774 (cit. on p. 4).

[Kra04a] Jan Krajíček. "Diagonalization in proof complexity". In: *Fundamenta Mathematicae* 182 (2004), pp. 181–192 (cit. on p. 4).

[Kra04b] Jan Krajíček. "Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds". In: *J. Symb. Log.* 69.1 (2004), pp. 265–286. DOI: 10.2178/JSL/1080938841. URL: https://doi.org/10.2178/jsl/1080938841 (cit. on pp. 4, 12).

[Kra05] Jan Krajíček. "Structured pigeonhole principle, search problems and hard tautologies". In: *J. Symb. Log.* 70.2 (2005), pp. 619–630. DOI: 10.2178/JSL/1120224731. URL: https://doi.org/10.2178/jsl/1120224731 (cit. on p. 4).

[Kra07] Jan Krajíček. "A proof complexity generator". In: *Proc. from the 13th International Congress of Logic, Methodology and Philosophy of Science (Beijing*. 2007 (cit. on p. 4).

[Kra11a] Jan Krajíček. *Forcing with Random Variables and Proof Complexity*. Vol. 382. London Mathematical Society lecture note series. Cambridge University Press, 2011. ISBN: 978-0-521-15433-8. URL: http://www.cambridge.org/de/academic/subjects/mathematics/logic-categories-and-sets/forcing-random-variables-and-proof-complexity?format=PB (cit. on pp. 4, 6).

[Kra11b] Jan Krajíček. "On the Proof Complexity of the Nisan-Wigderson Generator based on a Hard NP ∩ coNP function". In: *J. Math. Log.* 11.1 (2011). DOI: 10.1142/S0219061311000979. URL: https://doi.org/10.1142/S0219061311000979 (cit. on p. 4).

[Kra22] Jan Krajíček. "On the existence of strong proof complexity generators". In: *Electron. Colloquium Comput. Complex.* TR22-120 (2022). ECCC: TR22-120. URL: https://eccc.weizmann.ac.il/report/2022/120 (cit. on pp. 4, 7, 12).

[KS25] Oliver Korten and Rahul Santhanam. "How to Construct Random Strings". In: *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*. Ed. by Srikanth Srinivasan. Vol. 339. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 35:1–35:32. DOI: 10.4230/LIPICS.CCC.2025.35. URL: https://doi.org/10.4230/LIPIcs.CCC.2025.35 (cit. on p. 4).

[KZ20] Benjamin Kuykendall and Mark Zhandry. "Towards Non-interactive Witness Hiding". In: *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I*. Ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12550. Lecture Notes in Computer Science. Springer, 2020, pp. 627–656. DOI: 10.1007/978-3-030-64375-1_22. URL: https://doi.org/10.1007/978-3-030-64375-1%5C_22 (cit. on pp. 5–7, 14).

[Lau83] Clemens Lautemann. "BPP and the Polynomial Hierarchy". In: *Inf. Process. Lett.* 17.4 (1983), pp. 215–217. DOI: 10.1016/0020-0190(83)90044-3. URL: https://doi.org/10.1016/0020-0190(83)90044-3 (cit. on pp. 8, 11).

[Li24] Zeyong Li. "Symmetric Exponential Time Requires Near-Maximum Circuit Size: Simplified, Truly Uniform". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. STOC 2024. Vancouver, BC, Canada: Association for Computing Machinery, 2024, pp. 2000–2007. ISBN: 9798400703836. DOI: 10.1145/3618260.3649615. URL: https://doi.org/10.1145/3618260.3649615 (cit. on p. 4).

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs Randomness". In: *J. Comput. Syst. Sci.* 49.2 (1994), pp. 149–167. DOI: 10.1016/S0022-0000(05)80043-1. URL: https://doi.org/10.1016/S0022-0000(05)80043-1 (cit. on p. 2).

[Pas03] Rafael Pass. "Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition". In: *EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. Lecture Notes in Computer Science. Springer, 2003, pp. 160–176. DOI: 10.1007/3-540-39200-9_10. URL: https://doi.org/10.1007/3-540-39200-9%5C_10 (cit. on p. 7).

[Pic11] Ján Pich. "Nisan-Wigderson generators in proof systems with forms of interpolation". In: *Math. Log. Q.* 57.4 (2011), pp. 379–383. DOI: 10.1002/MALQ.201010012. URL: https://doi.org/10.1002/malq.201010012 (cit. on p. 4).

[Pud86] Pavel Pudlák. "On the length of proofs of finitistic consistency statements in first order theories". In: *Logic Colloquium '84*. Ed. by J.B. Paris, A.J. Wilkie, and G.M. Wilmers. Vol. 120. Studies in Logic and the Foundations of Mathematics. Elsevier, 1986, pp. 165–196. DOI: https://doi.org/10.1016/S0049-237X(08)70462-2. URL: https://www.sciencedirect.com/science/article/pii/S0049237X08704622 (cit. on p. 5).

[Raz15] Alexander A. Razborov. "Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution". English. In: *Ann. Math. (2)* 181.2 (2015), pp. 415–472. ISSN: 0003-486X. DOI: 10.4007/annals.2015.181.2.1 (cit. on p. 4).

[RR97] Alexander A. Razborov and Steven Rudich. "Natural Proofs". In: *J. Comput. Syst. Sci.* 55.1 (1997), pp. 24–35. DOI: 10.1006/jcss.1997.1494. URL: https://doi.org/10.1006/jcss.1997.1494 (cit. on pp. 3, 8, 9).

[RSW22] Hanlin Ren, Rahul Santhanam, and Zhikun Wang. "On the Range Avoidance Problem for Circuits". In: *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 2022, pp. 640–650. DOI: 10.1109/FOCS54457.2022.00067. URL: https://doi.org/10.1109/FOCS54457.2022.00067 (cit. on p. 4).

[Rud97] Steven Rudich. "Super-bits, Demi-bits, and NP/qpoly-natural Proofs". In: *Randomization and Approximation Techniques in Computer Science, International Workshop, RANDOM'97, Bolognna, Italy, July 11-12. 1997, Proceedings*. Ed. by José D. P. Rolim. Vol. 1269. Lecture Notes in Computer Science. Springer, 1997, pp. 85–93. DOI: 10.1007/3-540-63248-4_8. URL: https://doi.org/10.1007/3-540-63248-4%5C_8 (cit. on pp. 3, 7, 9).

[RWZ25] Hanlin Ren, Yichuan Wang, and Yan Zhong. *Hardness of Range Avoidance and Proof Complexity Generators from Demi-Bits*. Manuscript. 2025 (cit. on p. 7).

[Sha92] Adi Shamir. "IP = PSPACE". In: *J. ACM* 39.4 (1992), pp. 869–877. DOI: 10.1145/146585.146609. URL: https://doi.org/10.1145/146585.146609 (cit. on p. 5).

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il