Non-Trivial Zero-Knowledge Implies One-Way Functions

Suvradip Chakraborty* James Hulett† Dakshita Khurana‡
Kabir Tomer§

Abstract

A recent breakthrough [Hirahara and Nanashima, STOC'2024] established that if $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$, the existence of zero-knowledge (ZK) with negligible errors for NP implies the existence of one-way functions (OWFs). This work obtains a characterization of one-way functions from the worst-case complexity of zero-knowledge in the high-error regime.

Assuming $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$, we show that any *non-trivial*, *constant-round* public-coin ZK argument for NP implies the existence of OWFs, and therefore also (standard) four-message zero-knowledge arguments for NP. Here, we call a ZK argument *non-trivial* if the sum of its completeness, soundness and zero-knowledge errors is bounded away from 1.

As a special case, we also prove that non-trivial non-interactive ZK (NIZK) arguments for NP imply the existence of OWFs. Using known amplification techniques, this also provides an unconditional transformation from weak to standard NIZK proofs for all meaningful error parameters. Prior work [Chakraborty, Hulett and Khurana, CRYPTO'2025] was limited to NIZKs with constant zero-knowledge error $\varepsilon_{\mathsf{zk}}$ and soundness error ε_{s} satisfying $\varepsilon_{\mathsf{zk}} + \sqrt{\varepsilon_{\mathsf{s}}} < 1$.

*Visa Research. suvchakr@visa.com
†UIUC. jhulett2@illinois.edu
‡UIUC and NTT Research. dakshita@illinois.edu
§UIUC. ktomer2@illinois.edu

Contents

1 Introduction **1**
- 1.1 Concurrent Work . . . 3

2 Technical Overview **3**
- 2.1 Previous Techniques . . . 3
- 2.2 Barriers to Improvement . . . 5
- 2.3 Repetition to the Rescue . . . 7
- 2.4 Generalizing to Interactive Zero Knowledge . . . 8
  - 2.4.1 Working with Computational Zero-Knowledge . . . 10
  - 2.4.2 Working with Private Coins . . . 11

3 Preliminaries **12**
- 3.1 One-Way Functions . . . 12
- 3.2 Universal Extrapolation . . . 13
- 3.3 Zero-Knowledge . . . 14
- 3.4 Complexity Classes . . . 15
- 3.5 Chernoff Bounds . . . 16
- 3.6 Lemmas From [LMP24] . . . 16

4 OWF From Non-Trivial NIZK **17**

5 OWF From Non-Trivial Public-Coin ZK **20**
- 5.1 Prover $\widetilde{\mathsf{P}}$ . . . 21
- 5.2 Distinguisher $\mathcal{D}$ . . . 23
- 5.3 Algorithm $\mathcal{A}$ . . . 25
- 5.4 Putting It All Together . . . 29

6 Applications to Amplification **29**
- 6.1 Non-Interactive Zero-Knowledge . . . 30
- 6.2 Interactive Zero-Knowledge . . . 30

References **31**

A Private to Public Coin in the Absence of ai-OWFs **33**

B Proof of Lemma 2 **35**

1 Introduction

Zero-knowledge proofs are a cornerstone of modern cryptography, allowing a prover to convince a verifier about the validity of an NP statement without leaking any information about the witness. A major research agenda, initiated by Ostrovsky and Wigderson [Ost91, OW93], asks: what are the minimal cryptographic assumptions needed for zero-knowledge (ZK) proof systems? Their celebrated results established that, assuming NP is hard on average, the existence of ZK proofs for NP implies the existence of one-way functions (OWFs), a fundamental cryptographic primitive.

More recently, Hirahara and Nanashima [HN24] made a notable advance by proving that even the *worst-case* hardness of NP (specifically, $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$) suffices to construct one-way functions from zero-knowledge proofs for NP. In fact, they only require the existence of zero-knowledge proofs for a *specific* type of language outside $\mathsf{ioP/poly}$, thereby opening up a new route for constructing one-way functions based solely on worst-case hardness.

Non-trivial ZK and One-way Functions. What happens if we allow ZK proof systems with non-negligible and possibly large security errors? Unfortunately, techniques of [Ost91, OW93, HN24] that link zero-knowledge to one-way functions are critically dependent on the assumption of small errors. As we will show, this is not merely an artifact of their analysis, but a fundamental limitation of their reductions. This state of affairs leaves open a basic gap in our understanding of the complexity of ZK.

Note that ZK proof systems with $\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{s}}+\varepsilon_{\mathsf{zk}} \geq 1$ can be realized trivially[1], without any assumptions. Here $\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}$ and $\varepsilon_{\mathsf{zk}}$ denote the correctness, soundness and zero-knowledge errors of the proof system respectively. In this work, we will focus on *non-trivial* ZK proof systems with $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} < 1 - 1/p(|x|)$ for some polynomial $p(\cdot)$ in the input length $|x|$. We ask:

Do non-trivial ZK proof systems imply the existence of OWFs?

Non-trivial ZK proof systems may be a lot easier to construct than standard ZK: relaxing the negligible soundness or zero-knowledge error requirement can significantly lower the barriers to designing these protocols, potentially making them feasible even in settings where negligible-error notions are hard to build. But if we did manage to construct them, we would not know how to leverage them to build OWFs—leaving unresolved questions about whether such protocols could function as a stepping stone towards one-way functions, or if they are fundamentally weaker and cannot be used to construct basic cryptography.

This Work. Our main result is a complete characterization of the worst-case complexity of non-trivial constant round ZK.

[1]Observe that when $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} = 1$, one can have the CRS sample one of three protocols: have the verifier always reject, have the verifier always accept, or have the prover send the full witness in the clear and the verifier check it. If the CRS chooses the first with probability ε_{c}, the second with probability ε_{s}, and the third with probability $\varepsilon_{\mathsf{zk}}$, this trivial protocol will satisfy our requirements.

Theorem 1 (Informal)**.** *If* NP $\not\subseteq$ ioP/poly *and non-trivial public-coin, constant round (computational)* ZK *arguments for* NP *exist, then one-way functions exist.*

The above result requires the assumption that the underlying ZK argument is public-coin. We also obtain a result for private-coin protocols by showing that unless auxiliary-input one-way functions exist, we can convert any constant-round private-coin protocol into a public-coin one. Thus, non-trivial constant round ZK protocols yield auxiliary-input OWFs.

Theorem 2 (Informal)**.** *If* NP $\not\subseteq$ ioP/poly *and non-trivial constant round (computational)* ZK *arguments for* NP *exist, then auxiliary-input one-way functions exist.*

We also consider the special case of non-interactive zero-knowledge (NIZK) arguments for NP, where the prover sends a single proof string and both parties share a common reference string (CRS). We show that nontrivial NIZKs for NP, together with the assumption that NP $\not\subseteq$ ioP/poly, imply the existence of one-way functions.

Theorem 3 (Informal)**.** *If* NP $\not\subseteq$ ioP/poly *and non-trivial (computational)* NIZK *arguments for* NP *exist, then one-way functions exist.*

This improves recent work [CHK25] which obtained one-way functions from any NIZK with zero-knowledge error $\varepsilon_{\mathsf{zk}}$ and soundness error ε_{s} satisfying $\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathsf{c}} + 2\sqrt{\varepsilon_{\mathsf{s}}} < 1 - \frac{1}{p(|x|)}$ for some polynomial $p(\cdot)$, or satisfying $\varepsilon_{\mathsf{zk}} + \sqrt{\varepsilon_{\mathsf{s}}} < 1$ in the case where $\varepsilon_{\mathsf{zk}}$ and ε_{s} are constants and $\varepsilon_{\mathsf{c}} = o(1)$. However, the regime where $\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathsf{s}} < 1$ but $\varepsilon_{\mathsf{zk}} + \sqrt{\varepsilon_{\mathsf{s}}} \geq 1$ remained open until this work.

Application: Amplifying ZK. Arguably one of the most natural questions about non-trivial ZK proofs is whether they can be *amplified* to drive down their errors to negligible, and obtain "standard" ZK. For NIZKs, this question has been studied extensively recently, and such amplification has been shown to be possible under additional assumptions such as the existence of public-key encryption [GJS19], and even just one-way functions [BG24, AK25].

Our theorem above can be viewed as eliminating the need to make any cryptographic assumptions when amplifying NIZK. In more detail, if NP is easy in the worst case then NIZKs for NP exist unconditionally, whereas if NP is hard in the worst case, then Informal Theorem 3 builds one-way functions from non-trivial NIZKs for NP. The resulting one-way functions can be used, together with weak NIZKs, to obtain standard NIZKs by relying on known amplification theorems in [BG24, AK25].

For technical reasons we will still need to make the extremely mild complexity-theoretic assumption that NP $\not\subseteq$ (ioP/poly) $\setminus$ BPP. In other words, if NP languages are efficiently decidable infinitely often with advice, then they are efficiently decidable on all input lengths. This is an extremely mild complexity assumption: it is implied by NP $\not\subseteq$ ioP/poly, which is already a mild worst case assumption.

Corollary 1 (Informal)**.** *Suppose* NP $\not\subseteq$ (ioP/poly) $\setminus$ BPP*. Then non-trivial (computational)* NIZK *proofs imply standard* NIZK *proofs.*

Note that the corollary above applies only to NIZK proofs (with statistical soundness), since we currently do not have approaches to amplify arbitrary non-trivial NIZK arguments from assumptions weaker than public-key encryption.

A natural next question is whether similar amplification is possible for *interactive* ZK. Here, we note that one-way functions imply the existence of (private-coin) four-round ZK arguments [BJY97], yielding the following informal Corollary.

Corollary 2 (Informal)**.** *Suppose* $\mathsf{NP} \not\subseteq (\mathsf{ioP}/\mathsf{poly})\backslash\mathsf{BPP}$*. Then non-trivial public-coin constant-round (computational)* ZK *arguments imply four-round (private-coin)* ZK *arguments.*

1.1 Concurrent Work

Concurrent to our work, [CLV26] gave an alternative approach to building one-way functions from high-error zero-knowledge protocols. In the NIZK case, their results match ours exactly: assuming $\mathsf{NP} \not\subseteq \mathsf{ioP}/\mathsf{poly}$, any non-trivial NIZK argument requires a one-way function. In the interactive case, their result is incomparable to ours: for a k-message public-coin protocol, they require the stronger guarantee that $\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{s}}+(k-1)\varepsilon_{\mathsf{zk}} < 1-\frac{1}{p(|x|)}$, but are able to work with an arbitrary (polynomial) k.[2]

2 Technical Overview

For simplicity throughout this overview, we will highlight our techniques assuming that the completeness error $\varepsilon_{\mathsf{c}} = 0$. Formally we work with the general case where $\varepsilon_{\mathsf{c}} \neq 0$, but this is a straightforward extension of the analysis described in this overview.

In the following two subsections, we recall techniques used in prior works [OW93, CHK25] to show that zero-knowledge requires one-way functions, as well as barriers that prevent these techniques from working when $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} \approx 1$. These will build intuition towards our eventual approach. Readers that are already familiar with these works may skip ahead to our techniques described in Section 2.3.

We discuss prior ideas in the context of *non-interactive* zero-knowledge (NIZK), and then describe methods to close the gap in parameters in the non-interactive setting. However, we note that the more general interactive case requires the most significant new insights, discussed in Section 2.4.

2.1 Previous Techniques

[OW93] Techniques. The key tool used in [OW93] is Universal Extrapolation (UE): informally, UE lets us "reverse sample" the randomness that would generate a particular output of an efficient process. In more detail, given a polynomial-time sampler Samp and

[2]Note that we have rephrased this result slightly from how it appeared in [CLV26] to be in terms of the number of *messages* instead of the number of *rounds* (ie, pairs of messages). This is solely to avoid confusion, as in the rest of the paper we use the words message and round interchangeably.

an output y from its distribution, UE refers to the ability to efficiently and uniformly sample (up to small error) from the set of all inputs that Samp maps to y. UE is possible if and only if one-way functions do not exist [IL90].

Towards a contradiction, [OW93] assume the non-existence of one-way functions. Then using UE, they convert a NIZK $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ and simulator Sim for a language $\mathcal{L}$ into an algorithm $\mathcal{A}$ that decides $\mathcal{L}$. In particular, $\mathcal{A}$ samples a *real* CRS from Gen, then uses UE to find a random string r that would lead the *simulator* to output that same CRS. $\mathcal{A}$ then accepts if and only if V would accept the proof output by $\mathsf{Sim}(x; r)$.

To argue that $\mathcal{A}$ works, one has to show that it will accept $x \in \mathcal{L}$ with higher probability than it accepts $x \notin \mathcal{L}$. Consider each of these cases:

- If $x \notin \mathcal{L}$, $\mathcal{A}$ can only accept if it finds a proof *relative to a real* CRS that causes V to accept x. Soundness of the underlying NIZK will immediately tell us that this can happen with probability at most ε_{s}.

- If $x \in \mathcal{L}$, note that by completeness V will accept a real CRS and proof with probability 1 and so by zero-knowledge will accept a simulated CRS and proof with probability at least $1 - \varepsilon_{\mathsf{zk}}$.

  Now consider an algorithm $\mathcal{B}$ that is *almost* identical to $\mathcal{A}$, except that it starts by sampling a CRS from Sim instead of from Gen. We note first that this is (up to the small statistical error of the UE) identically distributed to simply running V on a simulated CRS and proof, so by the above $\mathcal{B}$ accepts $x \in \mathcal{L}$ with probability at least $1 - \varepsilon_{\mathsf{zk}}$. But note that since the only difference between $\mathcal{A}$ and $\mathcal{B}$ is in the original sampling of the CRS, and so by zero-knowledge their outputs must be $\varepsilon_{\mathsf{zk}}$-close. Thus, $\mathcal{A}$ accepts $x \in \mathcal{L}$ with probability at least $1 - 2\varepsilon_{\mathsf{zk}}$.

Thus, $\mathcal{A}$ is a correct algorithm for $\mathcal{L}$ as long as $1 - 2\varepsilon_{\mathsf{zk}}$ is noticeably larger than ε_{s}, which is equivalent to requiring that $\varepsilon_{\mathsf{s}} + 2\varepsilon_{\mathsf{zk}}$ is noticeably smaller than 1. Since one can implement $\mathcal{A}$ as long as one-way functions *don't* exist, this means that a NIZK for a hard language with $\varepsilon_{\mathsf{s}} + 2\varepsilon_{\mathsf{zk}} < 1$ *requires* the existence of a one-way function.

Note that in the analysis above, we first uses the zero-knowledge property to bound the difference between real and simulated proofs (costing one $\varepsilon_{\mathsf{zk}}$) and then again bound the difference between real and simulated crs (costing another $\varepsilon_{\mathsf{zk}}$). It is natural to wonder whether there may be a better way to analyze the same algorithm and show that $\mathcal{A}$ decides $\mathcal{L}$ whenever we have arbitrary *non-trivial* NIZKs, but in the following subsection (Section 2.2) we provide an explicit counterexample.

Techniques in CHK [CHK25]. The improvement in [CHK25] stems from the observation that one of the $\varepsilon_{\mathsf{zk}}$ errors occurs specifically when the analysis transitions the crs (and not the proof itself) from real to simulated. They then add a "CRS check" to the deciding algorithm $\mathcal{A}$ described above, which roughly predicts whether a CRS was sampled from the real or simulated distribution. This requires the use of Universal Approximation (UA), which intuitively allows one to approximate the probability of a given output coming from a given randomized process. Similar to the case of UE, [Imp92] showed that UA

is possible exactly when one-way functions don't exist[3].

In particular, [CHK25] modifies the algorithm $\mathcal{A}$ above by adding a step that estimates the probability that the CRS it samples came from Gen or from Sim, and immediately rejects if the latter is at least $\frac{1}{\sqrt{\varepsilon_s}}$ times larger than the former[4]. As in [OW93], one now has to show that the modified $\mathcal{A}$ accepts $x \in \mathcal{L}$ more often than $x \notin \mathcal{L}$.

- If $x \notin \mathcal{L}$, the above modification can only make $\mathcal{A}$ less likely to accept, so the acceptance probability is still upper-bounded by ε_s.
- If $x \in \mathcal{L}$, all the hybrids considered above have to be modified to also include the CRS checking step. It is possible to show that a real CRS fails the check with probability at most $\sqrt{\varepsilon_s}$ (ignoring a small error term from the UA), and thus a real CRS and proof will be accepted *and* pass the CRS check with probability at least $1 - \sqrt{\varepsilon_s}$. Thus, by zero-knowledge a *simulated* CRS and proof must pass both checks with probability at least $1 - \sqrt{\varepsilon_s} - \varepsilon_{zk}$.

  The key step in the [CHK25] analysis comes when considering $\mathcal{B}$, which as before is equivalent to $\mathcal{A}$ except that it starts with a simulated CRS. As in the prior analysis, this is identically distributed (up to the small statistical error of the UE) to sampling the CRS and proof from Sim directly, and so will accept $x \in \mathcal{L}$ with probability at least $1 - \sqrt{\varepsilon_s} - \varepsilon_{zk}$. However, instead of paying an *additive* ε_{zk} loss in acceptance probability between $\mathcal{B}$ and $\mathcal{A}$, [CHK25] notes that one can instead pay a *multiplicative* $\sqrt{\varepsilon_s}$ loss. Indeed, any CRS that is at most $\frac{1}{\sqrt{\varepsilon_s}}$ times more likely to come from Sim as from Gen can "contribute" at most $\frac{1}{\sqrt{\varepsilon_s}}$ times more the the acceptance probability of $\mathcal{B}$ as it does to $\mathcal{A}$; all other CRS are rejected by the checking step, and so "contribute" nothing in either algorithm. Thus, $\mathcal{A}$ accepts $x \in \mathcal{L}$ with probability at least $\sqrt{\varepsilon_s}$ times the acceptance probability of $\mathcal{B}$, which gives a final bound of $\sqrt{\varepsilon_s}\left(1 - \sqrt{\varepsilon_s} - \varepsilon_{zk}\right)$.

This means that $\mathcal{A}$ is a correct algorithm for $\mathcal{L}$ as long as $\sqrt{\varepsilon_s}\left(1 - \sqrt{\varepsilon_s} - \varepsilon_{zk}\right)$ is noticeably larger than ε_s; rearranging terms, this requires that $\varepsilon_{zk} + 2\sqrt{\varepsilon_s}$ is noticeably smaller than 1. As before, this says that a NIZK for a hard language with $\varepsilon_{zk} + 2\sqrt{\varepsilon_s} < 1$ must imply the existence of a one-way function.

2.2 Barriers to Improvement

We would ideally like a result that works for *any* non-trivial NIZK, meaning it should only need to assume that $\varepsilon_s + \varepsilon_{zk} < 1$. One might hope that a tighter analysis of the hybrids in [OW93] or [CHK25] could achieve such a result, but this turns out to be impossible: there exists a (contrived) protocol with $\varepsilon_s + \varepsilon_{zk} < 1$ for which neither algorithm $\mathcal{A}$ described above give advantage in distinguishing $x \in \mathcal{L}$ from $x \notin \mathcal{L}$!

[3] [CHK25] technically uses a stronger notion of UA than the one considered in [Imp92], but for our purposes we will ignore this subtlety.

[4] In fact, [CHK25] uses a slightly different algorithm in that they directly sample both the CRS and proof from Sim. However, the algorithm we describe here does appear as a hybrid in their analysis, and nothing in their techniques fundamentally changes if you treat that hybrid as the final algorithm for $\mathcal{L}$.

In particular, consider some language $\mathcal{L} \in \mathsf{P}$[5] and define a NIZK for $\mathcal{L}$ with errors $(\varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$ as follows:

- Gen : Output
$$\mathsf{crs} = \begin{cases} 0 & \text{w.p. } 1 - \varepsilon_{\mathsf{zk}}, \\ 1 & \text{w.p. } \varepsilon_{\mathsf{zk}}. \end{cases}$$
- $\mathsf{P}(\mathsf{crs})$ outputs as proof a bit $\pi = \mathsf{crs}$.
- V decides x on its own and if $x \in \mathcal{L}$, V outputs
$$\mathsf{V}(\mathsf{crs}, \pi, x) = \begin{cases} 1 & \text{if } \pi = \mathsf{crs}, \\ 0 & \text{if } \pi \neq \mathsf{crs}. \end{cases}$$
If $x \notin \mathcal{L}$, V outputs
$$\mathsf{V}(\mathsf{crs}, \pi, x) = \begin{cases} 1 & \text{w.p. } \varepsilon_{\mathsf{s}}, \\ 0 & \text{w.p. } 1 - \varepsilon_{\mathsf{s}}. \end{cases}$$
- $\mathsf{Sim}(x)$ outputs
$$(\mathsf{crs}, \pi) = \begin{cases} 00 & \text{w.p. } 1 - \varepsilon_{\mathsf{zk}}, \\ 01 & \text{w.p. } \varepsilon_{\mathsf{zk}}. \end{cases}$$

One can easily verify that the soundness error is exactly ε_{s} and the zero-knowledge error is exactly $\varepsilon_{\mathsf{zk}}$ by construction.

But now consider the behavior of the algorithm $\mathcal{A}$ from [OW93] if instantiated with this NIZK (assuming, for simplicity, that the UE has no errors):

- If $x \notin \mathcal{L}$, V will accept with probability ε_{s} regardless of what CRS and proof it is given, so $\mathcal{A}$ will accept with the same probability.
- If $x \in \mathcal{L}$, note that $\mathcal{A}$ can never sample the proof to be 1 if the CRS is 1, since Sim never outputs 11.[6] Thus, the only way for $\mathcal{A}$ to accept is if it samples the CRS as 0 (which happens with probability $1-\varepsilon_{\mathsf{zk}}$ from Gen) and the proof as 0 (which happens with probability $(1 - \varepsilon_{\mathsf{zk}})$ from Sim conditioned on the CRS being 0). Thus, $\mathcal{A}$ will only accept $x \in \mathcal{L}$ with probability $(1 - \varepsilon_{\mathsf{zk}})^2$.

Thus, at a minimum, we need to assume that $(1-\varepsilon_{\mathsf{zk}})^2$ is noticeably larger than ε_{s} in order for the [OW93] algorithm to have any chance of working. This rules out the possibility of a better analysis of the same algorithm giving us the optimal result[7].

[5]Note that while this language is by definition easy, that doesn't matter for the purposes of our counterexample; we are simply trying to show that *the particular algorithm* $\mathcal{A}$ doesn't decide $\mathcal{L}$.

[6]Technically, Sim can never output a CRS of 1, and so the behavior of $\mathcal{A}$ if it samples 1 from Gen is undefined. Informally, we can say that if the distribution of Sim is undefined, $\mathcal{A}$ simply fails. Alternatively, we can fix this formally by having Sim output 10 with some tiny probability δ, where we set δ to be small enough to not affect our analysis meaningfully.

[7]As an explicit example, the parameter setting of $\varepsilon_{\mathsf{zk}} = 0.5$ and $\varepsilon_{\mathsf{s}} = 0.25$ is non-trivial, but the [OW93] algorithm would accept x with probability 0.25 regardless of if $x \in \mathcal{L}$ or not.

Additionally, the same counterexample also rules out the possibility of getting a tight result from the modified decision algorithm $\mathcal{A}$ in [CHK25]. Indeed, if the parameters are set such that the CRS check rejects when the CRS is 0, $\mathcal{A}$ will *never* accept $x \in \mathcal{L}$. If instead the CRS check accepts when the CRS is 0, then it accepts *every* CRS[8], and thus (for this particular NIZK) the algorithm from [CHK25] is functionally equivalent to that from [OW93]!

2.3 Repetition to the Rescue

In order to improve on prior techniques, it will help to first understand *why* the above counterexample is bad for them. We note that the counterexample hinges on the fact that on $\mathsf{crs} = 0$ the simulator always has the *ability* to output a Good proof (i.e. 0), but still chooses to output a Bad proof (i.e. 1) some of the time. Here by Good we mean a proof that is statistically close to a real proof, and by Bad we mean a proof that is far from a real proof. This causes the distribution used by $\mathcal{A}$ to have $2\varepsilon_{\mathsf{zk}}$ error (once due to the distance between the real and simulated CRS and once due to the distance between the real and simulated proofs *conditioned on* a particular CRS), while still keeping the real and simulated (CRS, proof) distributions $\varepsilon_{\mathsf{zk}}$-close overall.

It turns out that this type of simulator behavior is the *only* reason prior techniques don't work with protocols where $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} \approx 1$. In particular, for every fixing of $x \in \mathcal{L}$, if every CRS was either Good (i.e., the simulator conditioned on this CRS *always* outputs an accepting proof for x) or Bad (i.e., the simulator conditioned on this CRS *never* outputs an accepting proof for x), then the algorithm $\mathcal{A}$ from [OW93] *will work* only assuming that $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} < 1$! This is because on any Bad CRS output by Gen, the real prover is guaranteed to output an accepting proof whereas the simulator never does. This allows us to infer that Gen must output a Good CRS with probability at least $1 - \varepsilon_{\mathsf{zk}}$, and thus $\mathcal{A}$ will accept $x \in \mathcal{L}$ with the same probability.

Of course, we cannot in general assume that the simulator always works on certain CRSes and always fails on others. However, we can achieve a similar effect by sampling *many* simulated proofs for the same CRS and checking whether *at least one* accepts.[9] In particular, this allows us to consider a CRS to be Good as long as the simulator conditioned on this CRS outputs an accepting proof for x with "not too small" probability, which means that our sampling will see *at least one* accepting proof with overwhelming probability. At the same time, we can still argue that we can't see a BadCRS too often, as the real prover will output an accepting proof for such a CRS, whereas the simulator "almost never" will.

In more detail, we fix an appropriate polynomial $p(\cdot)$ and consider a modified algorithm $\mathcal{A}'$ that repeats the [OW93] algorithm $p(|x|)$ times (all with respect to the same CRS, but reverse sampling fresh randomness each time) and accepts if *any* of the repetitions give an accepting proof.

[8] Note that Sim outputs the CRS as 1 with probability *smaller* than Gen does, so the CRS check can never reject this.

[9] We note that [Ps05] use a similar idea, where they categorize the crs according to whether there *exists* an accepting proof relative to it. However, their analysis is limited to the negligible-error setting and doesn't seem to naturally generalize to the interactive setting.

- For $x \notin \mathcal{L}$, $\mathcal{A}'$ still needs to find an accepting proof relative to a real CRS in order to accept. Since repeatedly running the [OW93] algorithm is a valid strategy to break soundness of the NIZK, we have that the probability that even one of the proofs accept is at most ε_{s}.

- For $x \in \mathcal{L}$, we call a CRS Good if Sim conditioned on outputting CRS gives an accepting proof for x with probability $\geq \frac{|x|}{p(|x|)}$ and Bad otherwise. Very roughly, as long as $p(|x|)$ is large enough, we can as above argue that Gen can only output a Bad CRS with probability (approximately) $\varepsilon_{\mathsf{zk}}$, since doing so contributes directly to the zero-knowledge error of the NIZK[10]. And if Gen outputs a Good CRS, $\mathcal{A}'$ will find an accepting proof for x with overwhelming probability. Thus, up to small errors, $\mathcal{A}'$ will accept $x \in \mathcal{L}$ with probability at least $1 - \varepsilon_{\mathsf{zk}}$.

As in prior analysis, we just need the probability that $\mathcal{A}'$ accepts $x \in \mathcal{L}$ to be noticeably larger than the probability it accepts $x \notin \mathcal{L}$. Rearranging terms, this only requires that $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}}$ be noticeably smaller than 1—meaning we can work with *any* non-trivial NIZK!

We have swept some technical details under the rug, but the above ideas coarsely suffice to obtain OWFs from non-trivial NIZKs. In the following section, we discuss barriers and methods to extend these ideas to the interactive setting.

2.4 Generalizing to Interactive Zero Knowledge

As discussed in the introduction, our primary goal is to obtain an implication from non-trivial *honest-verifier* ZK to OWFs, since this gives us the strongest possible theorem statement. First, we note that techniques from [OW93] naturally extend to the interactive case, but get further from the desired result as the number of rounds increases: to get a one-way function from a k round protocol, they require $\varepsilon_{\mathsf{s}} + k\varepsilon_{\mathsf{zk}} < 1$. Similarly, simple extensions of the [CHK25] techniques require $(\varepsilon_{\mathsf{s}})^{1/k} + \varepsilon_{\mathsf{zk}} < 1$.

A natural goal is therefore to see if we can extend the techniques that gave us a tight result for NIZKs to the interactive setting, and more specifically, eliminate the dependence on k in the above constraints.

In order to generalize our techniques to the interactive setting, it will help to take a slightly different view of our proof in the non-interactive case. We can treat our algorithm $\mathcal{A}$ as defining a malicious prover $\widetilde{\mathsf{P}}$, then simulating a run of $\widetilde{\mathsf{P}}$ and V to decide whether to accept x. In the analysis above, we treated $\widetilde{\mathsf{P}}$ as using UE to sample many possible simulated proof strings, and outputting an accepting one if found. But note that we equally well could phrase $\widetilde{\mathsf{P}}$ as choosing whichever sampled proof string has the maximum probability of being accepted. Phrased this way, we can now begin to extend this idea to the interactive case.

[10] Specifically, note that Sim will almost never output a Bad CRS with an accepting proof, since by definition the probability Sim's proof is accepting conditioned on the CRS being bad is at most $\frac{1}{p(|x|)}$. Since we are assuming perfect completeness however, P will always output an accepting proof, even when the CRS from Gen is Bad. Thus, if Gen outputs a Bad CRS with probability much higher than $\varepsilon_{\mathsf{zk}}$, we can break zero-knowledge with a distinguisher that checks if the CRS is Bad but the proof is still accepting. Here, note that we can (probably approximately) estimate whether a CRS is Good or Bad *efficiently* by using UE to sample many possible simulated proofs relative to it, and checking what fraction of them are accepting.

An Inefficient Prover. We would like to define a malicious prover $\widetilde{\mathsf{P}}$ such that our algorithm $\mathcal{A}$ to decide $\mathcal{L}$ can simply simulate an interaction between $\widetilde{\mathsf{P}}$ and V on x and output the result. Unlike the non-interactive case, $\widetilde{\mathsf{P}}$ can't immediately check if the message it is considering would lead to V accepting or rejecting, since this decision may depend on randomness sampled at a later point in the protocol. Thus, all we can hope for is to have $\widetilde{\mathsf{P}}$ choose the "best" message available to it, which means choosing the message that maximizes the probability that V accepts *if we continue to apply the prover strategy $\widetilde{P}$ in future rounds*. For now, we will assume that $\widetilde{P}$ can (inefficiently) find such messages, and we will later see how to make $\widetilde{P}$ efficient.

More formally, suppose for simplicity that we are starting from a protocol that satisfies statistical soundness and zero-knowledge. We can (recursively) define $\widetilde{\mathsf{P}}$ that, given x and a partial transcript τ, considers all possible next messages m in the support of Sim and outputs whichever one maximizes the probability of V accepting x if it interacts with $\widetilde{\mathsf{P}}$ starting from transcript $\tau \| m$. Here, defining this success probability based *only* on the transcript and not on V's internal state limits us to only working with public-coin protocols.

Let $\mathcal{A}(x)$ be the algorithm that runs $\widetilde{\mathsf{P}}$ and V on input x and outputs V's decision bit; we can now argue that $\mathcal{A}$ correctly decides $\mathcal{L}$.

- If $x \notin \mathcal{L}$, statistical soundness tells us that no prover can convince V with probability greater than ε_{s}, and so in particular $\widetilde{\mathsf{P}}$ cannot. Thus, $\mathcal{A}(x)$ will accept with probability at most ε_{s}.
- If $x \in \mathcal{L}$, we can argue that $\widetilde{\mathsf{P}}$ performs "almost as well as" the honest prover P. In particular, we note that since $\widetilde{\mathsf{P}}$ always chooses the best possible message in the support of Sim, the only way for P to choose a better message is if it sends a message that Sim never would. But from statistical zero-knowledge, we know that this can happen with probability at most $\varepsilon_{\mathsf{zk}}$![11] Thus, the probability that $\widetilde{\mathsf{P}}$ convinces V is at least $(1 - \varepsilon_{\mathsf{zk}})$, which tells us that $\mathcal{A}$ accepts with probability at least $(1 - \varepsilon_{\mathsf{zk}})$.

As in the NIZK case, this immediately gives us an algorithm for $\mathcal{L}$ as long as $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}}$ is noticeably smaller than 1.

Making $\widetilde{\mathsf{P}}$ Efficient. So far, we required $\widetilde{\mathsf{P}}$ to find the *best* possible next prover message in the support of Sim, which may not be efficient to compute. This inefficiency meant that our eventual decision algorithm $\mathcal{A}$ is inefficient, which is a problem. To fix these issues, we need to modify how we define $\widetilde{\mathsf{P}}$ so that it has a polynomial run time, but still works well enough to argue that $\mathcal{A}(x)$ will accept with good probability when $x \in \mathcal{L}$.

Taking inspiration from the NIZK case, we can have $\widetilde{\mathsf{P}}$ use UE to sample many possible next messages from Sim, and simply choose the best message *among those it sampled*. While this won't guarantee that we get the best *possible* next message, it does guarantee

[11] Importantly, we are applying zero-knowledge *globally* to the entire transcript instead of on a message-by-message basis. This is the key reason why we avoid the $O(k)\varepsilon_{\mathsf{zk}}$ term in the analysis from [OW93].

that (with overwhelming probability) we get a message that is "one of the best". In particular, suppose we order all the messages in the support of Sim by the probability that $\widetilde{\mathsf{P}}$ will convince V if it chooses that message next. For any δ, if we have $\widetilde{\mathsf{P}}$ sample $\frac{n}{\delta}$ many possible next messages, it will sample at least one in the top δ fraction of this order with overwhelming probability.

There is however one catch to this strategy that only arises in the interactive case: in order to choose the best message among those sampled, $\widetilde{\mathsf{P}}$ has to be able to compute the probability that each message will lead to it convincing V. Note though that as long as V is public coin (and so stateless), $\widetilde{\mathsf{P}}$ can get a very good estimate of this probability, as $\widetilde{\mathsf{P}}$ can simulate an interaction between itself and V "in its own head". A Chernoff bound says that $\widetilde{\mathsf{P}}$ only needs to run $\mathrm{poly}(n, \frac{1}{\delta})$ such simulations in order to (with overwhelming probability) estimate its success probability to within an additive δ factor for any δ. Note, however, that since $\widetilde{\mathsf{P}}$ has to simulate itself a polynomial number of times on a transcript that is only one message longer, its run time will grow by a *multiplicative* factor of $\mathrm{poly}(n)$ for each round. This will give us a final run time of $\mathrm{poly}(n)^k$, which constrains us to only work with protocols where k is a constant.

Putting this all together, we get an *efficient* $\widetilde{\mathsf{P}}$ that with overwhelming probability chooses a next message that has success probability at most 2δ smaller than some message in the top δ fraction of messages in the support of Sim. We claim that as long as δ is set to a sufficiently small inverse polynomial, our prior analysis is robust enough to work with these weaker guarantees on $\widetilde{\mathsf{P}}$.

In more detail, note that all we have to change in the above analysis is the argument that P can't pick a "better" message than $\widetilde{\mathsf{P}}$ too commonly. Since our weaker guarantee on $\widetilde{\mathsf{P}}$ is that it picks a message (approximately) in the top δ fraction of possible messages from the support of Sim, we just have to bound the probability that P picks a message that is better than a $(1-\delta)$ fraction of the support of Sim.

By a union bound over each round, the probability that Sim ever outputs such a message is at most δk, so by (statistical) zero-knowledge, the probability that a real transcript contains such a message is at most $\varepsilon_{\mathsf{zk}} + \delta k$. By our guarantees on $\widetilde{\mathsf{P}}$, we know that it loses at most 2δ success probability in each round compared to P conditioned on not picking such a "too good" message. Thus, since P will always convince V to accept $x \in \mathcal{L}$, we have that $\widetilde{\mathsf{P}}$ convinces V with probability at least $1 - \varepsilon_{\mathsf{zk}} - 3\delta k$.

Finally, in order for $\mathcal{A}$ to be correct, we will need ε_{s} to be noticeably smaller than $1 - \varepsilon_{\mathsf{zk}} - 3\delta k$. As long as $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}}$ is noticeably smaller than 1, we can pick an appropriate inverse-polynomial δ that makes this happen.

2.4.1 Working with Computational Zero-Knowledge

Up until this point, we've been applying zero-knowledge with respect to a distinguisher $\mathcal{D}$ that *perfectly* answers questions about the support of Sim—in particular, one that checks if a given prover message better than the bottom $(1-\delta)$ fraction of the support of Sim. Here recall that a message a is better than b if the probability that $\widetilde{P}$ convinces V is higher if $\widetilde{P}$ chooses a in its next round, compared with b.

If we want to work with protocols that only have computational zero-knowledge, we need to come up with an efficient distinguisher that still works well enough for our proof.

Our efficient distinguisher will try to *approximate* the fraction of messages from Sim that a given message is better than. To do this, it will sample a large number of possible next messages from the support of Sim and compare them to the given message. Formally, it will sample $\mathrm{poly}(n, \frac{1}{\delta})$ such messages, and check if at most a 2δ fraction of them are better than the given prover message. By a Chernoff bound, we have that if the given prover message is in fact better than a $(1-\delta)$ fraction of all possible messages in the support of Sim, this check will pass with overwhelming probability.

As in the previous section, we note that while we may not be able to compute the *exact* success probability efficiently, the same estimation procedure can with overwhelming probability get us within an additive factor of δ for any inverse-polynomial δ. Thus, except with negligible probability, we know that our distinguisher will say "real" if *any* prover message in its input transcript is at least 2δ better than the bottom $(1-\delta)$ fraction of messages in the support of Sim.

By a careful analysis, we can show that that this efficient distinguisher still works for the analysis in the previous section. Indeed, there are only two points where we had to use properties of the distinguisher: first to argue that it rarely outputs "real" on a simulated transcript, and second to argue that $\widetilde{\mathsf{P}}$ does "almost as well" as P conditioned on the distinguisher saying "simulated".

- For the former case, we claim that the efficient distinguisher says "real" on a transcript from Sim with probability at most $2\delta k$. For any given round, we know that the prover message in the transcript was generated by Sim, and hence comes from the same distribution as the messages the distinguisher samples. Thus, the probability that the message in the transcript is in the top 2δ fraction of the sampled messages is just 2δ, regardless of what procedure we use to compare them.

- For the latter case, we note that the only change between this section and the last is that the efficient distinguisher may allow P to have messages that have up to 2δ higher success probability than the bottom $(1-\delta)$ fraction of messages in the support of Sim. Since in the last section we already allowed $\widetilde{\mathsf{P}}$ to output messages that have up to 2δ *lower* probability than the top δ fraction of messages, we essentially are just doubling the error from there.

In both cases, we see that setting δ to be half as large as the setting of δ in the previous section suffices to make the same analysis go through.

2.4.2 Working with Private Coins

The above proof critically relies on the zero-knowledge protocol we start with being public-coin so that we can efficiently simulate V starting from an arbitrary transcript and thus estimate the probability of it accepting. However, we note that at least intuitively, we shouldn't have to make this assumption: if V is able to computationally hide information about its randomness in a private-coin protocol, intuitively the map from V's randomness to its messages is a one-way function.

We formalize this intuition in Appendix A, showing that either infinitely-often auxiliary-input uniform-secure OWFs exist or every constant-round private-coin protocol can be converted into an equivalent public-coin protocol.

We note that, unlike in the case of public-coin protocols, we obtain *auxiliary-input* one-way functions. This is because in our transformation, in order to achieve soundness the verifier has to resample its randomness such that it is consistent with the full transcript. Since the verifier does not have access to the malicious prover's code, there does not immediately seem to be a way to construct an efficient sampler such that UE on that sampler will allow us to do this. We get past this by allowing the sampler to take the transcript as auxiliary input, but this limits our result to achieving auxiliary-input one-way functions. Since this auxiliary input comes from the prover messages and not from the worst-case instance, switching to average-case hardness does not seem to improve the situation. Nonetheless, our results rule out information-theoretic constructions of (constant-round) non-trivial ZK even in the private-coin setting.

3 Preliminaries

We use the following notation:

- $[n]$ represents the set $\{1, 2, \ldots, n\}$ and $[n]_0$ represents the set $\{0, 1, \ldots, n\}$.
- We call a function μ negligible if it is smaller than any inverse polynomial, that is if for every polynomial p there exists an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $\mu(n) < \frac{1}{p(n)}$. We let negl denote an unspecified negligible function.
- We call a function μ noticeable if it is asymptotically larger than some inverse polynomial, that is if there exists a polynomial p and $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $\mu(n) \geq \frac{1}{p(n)}$.
- $a <_n b$ represents that a is noticeably less than b, that is that there exists a polynomial p and $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $a(n) < b(n) - \frac{1}{p(n)}$.

In this paper, our protocols will have (honest) participants that can be described as uniform Turing machines, whereas adversaries will (w.l.o.g.) be non-uniform Turing Machines, or equivalently, families of polynomial-sized circuits.

3.1 One-Way Functions

Definition 1 (One-way Functions)**.** *Fix an alphabet Σ. We say that a family of functions $\{f_x\}_{x \in \Sigma^*}$ with $f_x : \{0,1\}^{m_i(|x|)} \to \{0,1\}^{m_o(|x|)}$ is one-way if*

- ***[Efficiency]** There exists a Turing machine M and polynomial p such that for all $x \in \Sigma^*$ and $r \in \{0,1\}^{m_i(|x|)}$, $M(x, r)$ outputs $f_x(r)$ in time at most $p(|x|)$*

- ***[One-Wayness]*** *For all non-uniform PPT machines $\mathcal{A}$, all polynomials p, and all sufficiently large n, there exists an $x \in \Sigma^n$ with*

$$\Pr\left[f_x(\mathcal{A}(x, f_x(r))) = f_x(r) | r \xleftarrow{\$} \{0,1\}^{m_i(|x|)}\right] \leq \frac{1}{p(n)}$$

Definition 2 (Weak One-way Functions)**.** *Fix an alphabet Σ. We say that a family of functions $\{f_x\}_{x \in \Sigma^*}$ with $f_x : \{0,1\}^{m_i(|x|)} \to \{0,1\}^{m_o(|x|)}$ is weakly one-way if*

- ***[Efficiency]*** *There exists a Turing machine M and polynomial p such that for all $x \in \Sigma^*$ and $r \in \{0,1\}^{m_i(|x|)}$, $M(x,r)$ outputs $f_x(r)$ in time at most $p(|x|)$*
- ***[Weak One-Wayness]*** *For all non-uniform PPT machines $\mathcal{A}$ there exists a polynomial p such that for sufficiently large n there exists an $x \in \Sigma^n$ with*

$$\Pr\left[f_x(\mathcal{A}(x, f_x(r))) = f_x(r) | r \xleftarrow{\$} \{0,1\}^{m_i(|x|)}\right] \leq 1 - \frac{1}{p(n)}$$

Remark 1. *A few remarks about these definitions are in order:*

- *The definition above can be used to define both (standard) one-way functions, and one-way functions with auxiliary input. If we take $\Sigma = \{0,1\}$, these definitions correspond to (weak) one-way functions with auxiliary inputs; if we take $\Sigma = \{1\}$, these definitions correspond to (weak) one-way functions with no auxiliary input. In this paper, if we refer to an "auxiliary input (weak) one-way function", we mean one with respect to $\Sigma = \{0,1\}$; if we don't otherwise specify Σ, we default to $\Sigma = \{1\}$. A similar remark applies to all our definitions that make reference to Σ.*
- *Since we require that M is computable in time polynomial in $|x|$ (as opposed to in the size of its full input), we can assume WLOG that m_i and m_o are polynomially-bounded functions.*

While weak one-way functions may at first seem much weaker than one-way functions, the two notions are existentially equivalent.

Theorem 4 ([Yao82])**.** *Fix an alphabet Σ. One-way functions exist with respect to Σ if and only if weak one-way functions do.*

3.2 Universal Extrapolation

Definition 3 (Sampler)**.** *Fix an alphabet Σ. We say that a probabilistic polynomial time machine M is a sampler from m_i input bits to m_o output bits if for all $x \in \Sigma^*$, $M(x)$ uses at most $m_i(|x|)$ bits of randomness and for all $r \in \{0,1\}^{m_i(|x|)}$, $M(x;r) \in \{0,1\}^{m_o(|x|)}$.*

Remark 2. *Note that m_i and m_o can WLOG be bounded by the run time of M; since M runs in time polynomial in $|x|$, we can freely assume that m_i and m_o are polynomially-bounded functions of n.*

Definition 4 (Infinitely-Often Universal Extrapolation)**.** *Fix an alphabet Σ. We say that Infinitely-Often Universal Extrapolation (io-UE) is possible if for every sampler M from m_i input bits to m_o output bits and all polynomials p, there exists a non-uniform PPT machine N such that for infinitely many n, the following two distributions are $\frac{1}{p(n)}$-close in statistical distance for every $x \in \Sigma^n$:*

1. *Sample $r \xleftarrow{\$} \{0,1\}^{m_i(n)}$ and output $r \| M(x;r)$*
2. *Sample $r \xleftarrow{\$} \{0,1\}^{m_i(n)}$ and output $N(x, M(x;r)) \| M(x;r)$*

Theorem 5 ([IL90])**.** *Fix an alphabet Σ. io-UE is possible with respect to Σ if and only if one-way functions do* not *exist with respect to Σ.*

3.3 Zero-Knowledge

Definition 5 ($(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK)**.** *An $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK argument for an* NP *relation $\mathcal{R}$ is a pair of uniform PPT interactive algorithms (P, V) such that*

- ***Syntax***
  - *P is given input (x, w) and V is given input x.*
  - *P and V interact over the course of k rounds, creating a transcript τ.*
  - *After the interaction, V outputs a single bit. We let $\langle \mathsf{P}(w), \mathsf{V} \rangle(x)$ denote this output.*
- ***Completeness**: For every $(x, w) \in \mathcal{R}$,*

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

- ***Soundness**: For any* non-uniform *PPT algorithm $\widetilde{\mathsf{P}}$, we have that for $x \notin \mathcal{L}_{\mathcal{R}}$ such that $|x|$ is large enough,*

$$\Pr\left[\langle \widetilde{\mathsf{P}}, \mathsf{V} \rangle(x) = 1\right] \leq \varepsilon_{\mathsf{s}}(|x|)$$

- ***Zero-Knowledge**: There exists a PPT simulator* Sim *such that for all non-uniform PPT distinguishers $\mathcal{D}$ and all $(x, w) \in \mathcal{R}$ with $|x|$ large enough,*

$$\left| \Pr\left[\mathcal{D}(r, \tau) = 1 \mid (r, \tau) \leftarrow \mathsf{Sim}(x)\right] - \Pr\left[\mathcal{D}(r, \tau) = 1 \middle| \begin{array}{c} r \xleftarrow{\$} \{0,1\}^m \\ \tau \leftarrow \langle \mathsf{P}(w), \mathsf{V} \rangle(x; r) \end{array}\right] \right| \leq \varepsilon_{\mathsf{zk}}(|x|)$$

  where we let m be the number of random bits used by V and $\langle \mathsf{P}(w), \mathsf{V} \rangle(x; r)$ denote the transcript generated by P and V interacting, fixing V's random tape to r.

Remark 3. *We say a ZK argument for $\mathcal{L}$ is* public-coin *if the every message from V is simply a fresh random string. In particular, note that this implies without loss of generality that V does not maintain an internal state between rounds.*

Definition 6 ($(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-NIZK)**.** *An $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-NIZK for an* NP *relation $\mathcal{R}$ is a tuple of polynomial-time algorithms $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ such that*

- ***Syntax***
  - $\mathsf{crs} \leftarrow \mathsf{Gen}(1^n)$*:* Gen *is a randomized algorithm that on input an instance size (in unary) outputs a common reference string.*
  - $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$*:* P *is a randomized algorithm that on input a crs, instance, and witness for that instance outputs a proof.*
  - $b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$*:* V *is a* deterministic *algorithm that on input a crs, instance, and proof for that instance outputs a bit denoting acceptance or rejection.*
- ***Completeness****: For every* $(x, w) \in \mathcal{R}$*,*

$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, \pi) = 1 \middle| \begin{matrix}\mathsf{crs} \leftarrow \mathsf{Gen}(1^{|x|}) \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\end{matrix}\right] \geq 1 - \varepsilon_{\mathsf{c}}(|x|)$$

- ***Soundness****: For any* non-uniform *PPT algorithm* $\widetilde{\mathsf{P}}$*, we have that for* $x \notin \mathcal{L}_{\mathcal{R}}$ *such that* $|x|$ *is large enough,*

$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, \tilde{\pi}) = 1 \middle| \begin{matrix}\mathsf{crs} \leftarrow \mathsf{Gen}(1^{|x|}) \\ \tilde{\pi} \leftarrow \widetilde{\mathsf{P}}(\mathsf{crs}, x)\end{matrix}\right] \leq \varepsilon_{\mathsf{s}}(|x|)$$

- ***Zero-Knowledge****: There exists a PPT simulator* Sim *such that for all non-uniform PPT distinguishers* $\mathcal{D}$ *and all* $(x, w) \in \mathcal{R}$ *with* $|x|$ *large enough,*

$$\left|\Pr\left[\mathcal{D}(\mathsf{crs}, \pi) = 1 \mid (\mathsf{crs}, \pi) \leftarrow \mathsf{Sim}(x)\right] - \Pr\left[\mathcal{D}(\mathsf{crs}, \pi) = 1 \middle| \begin{matrix}\mathsf{crs} \leftarrow \mathsf{Gen}(1^{|x|}) \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\end{matrix}\right]\right| \leq \varepsilon_{\mathsf{zk}}(|x|)$$

Remark 4. *We will often refer to a (non-interactive) zero-knowledge argument as being for the language defined by the relation* $\mathcal{R}$*, rather than for the relation itself.*

Remark 5. *We have defined soundness for NIZKs as being* computational *and* non-adaptive*. One can also consider* statistical *soundness, where our guarantee needs to hold even against inefficient* $\widetilde{\mathsf{P}}$*. Additionally, one can consider* adaptive *soundness, where the adversary is allowed to choose the instance after seeing the CRS. Formally, our soundness requirement would be replaced by*

$$\Pr\left[\begin{matrix}|x| = n \wedge x \notin \mathcal{L}_{\mathcal{R}} \\ \wedge\ \mathsf{V}(\mathsf{crs}, x, \tilde{\pi}) = 1\end{matrix} \middle| \begin{matrix}\mathsf{crs} \leftarrow \mathsf{Gen}(1^n) \\ (x, \tilde{\pi}) \leftarrow \tilde{\mathsf{P}}(\mathsf{crs})\end{matrix}\right] \leq \varepsilon_{\mathsf{s}}(n)$$

for sufficiently large n*.*

3.4 Complexity Classes

Definition 7 (Infinitely-Often BPP)**.** *Let* ioBPP *be the class of languages* $\mathcal{L}$ *such that there exists a PPT algorithm* $\mathcal{A}$*, functions* $a, b : \mathbb{N} \to [0, 1]$*, and a polynomial* p *such that for infinitely many* n*,*

- *For all* $x \in \mathcal{L} \cap \{0, 1\}^n$*,* $\Pr[\mathcal{A}(x) = 1] \geq a(n)$

- *For all* $x \in \{0,1\}^n - \mathcal{L}$, $\Pr[\mathcal{A}(x) = 1] \leq b(n)$
- $a(n) - b(n) \geq \frac{1}{p(n)}$

We let ioBPP/poly *be defined similarly, except that* $\mathcal{A}$ *is allowed to be non-uniform.*

Definition 8 (Infinitely-Often P/poly)**.** *Let* ioP/poly *be the class of languages* $\mathcal{L}$ *such that there exists a non-uniform polynomial-time algorithm* $\mathcal{A}$ *such that for infinitely many* n*,*

- *For all* $x \in \mathcal{L} \cap \{0,1\}^n$, $\mathcal{A}(x) = 1$
- *For all* $x \in \{0,1\}^n - \mathcal{L}$, $\mathcal{A}(x) = 0$

Remark 6. *The textbook proof that* BPP $\subseteq$ P/poly *also says that* ioBPP/poly $\subseteq$ ioP/poly*, so in particular* ioBPP/poly $=$ ioP/poly*. Since* ioP/poly *is a more standard complexity class, we will state our results in terms of it, even when the algorithms we give are in fact probabilistic.*

3.5 Chernoff Bounds

We will need a few different standard instantiations of Chernoff bounds for binomial random variables, as listed below.

Theorem 6 (Chernoff Bounds)**.** *For every* $i \in [m]$*, let* X_i *be an independent Bernoulli random variable that takes value* 1 *with probability* p*, and let* $X := \sum_i X_i$*. Then:*

1. *for any* $0 < \delta < 1$, $\Pr[X \geq (1+\delta)mp] \leq \exp\left(-\frac{\delta^2 mp}{3}\right)$.
2. *for any* $0 < \delta < 1$, $\Pr[X \leq (1-\delta)mp] \leq \exp\left(-\frac{\delta^2 mp}{2}\right)$.
3. *for any* $0 < \Delta$, $\Pr[|X - mp| \geq \Delta] \leq 2\exp\left(-\frac{\Delta^2}{m}\right)$.

3.6 Lemmas From [LMP24]

In order to bridge the gap from auxiliary-input one-way functions to standard one-way functions, we recall the following definition and lemmas from [HN24] as presented in [LMP24].

Definition 9 (Infinitely-Often Average-Case BPP, [LMP24] Definition 2.1)**.** *Let* $\mathcal{L}$ *be a language and* $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ *be an efficiently sampleable distribution. We say* $(\mathcal{L}, \mathcal{D}) \in$ ioAvgBPP/poly *if there exists a non-uniform PPT* $\mathcal{A}$ *such that for infinitely many* $n \in \mathbb{N}$*:*

- *For all* $x \in \mathrm{Supp}(\mathcal{D}_n)$, $\Pr[\mathcal{A}(x) \in \{\bot, \mathcal{L}(x)\}] \geq 0.9$
- $\Pr\left[\mathcal{A}(x) = \bot \middle| \, x \xleftarrow{\$} \mathcal{D}_n\right] \leq 0.25$

Lemma 1 ([LMP24] Lemma 3.5)**.** *Suppose auxiliary-input one-way functions exist. Then there is a language* $\mathcal{L} \in$ NP *and a polynomial* $m(\cdot)$ *such that* $(\mathcal{L}, \{U_{m(n)}\}_{n\in\mathbb{N}}) \notin$ ioAvgBPP/poly*.*

In order to fit the next lemma from [LMP24] into our framework, it will help to introduce some new notation.

Definition 10 (Decision-To-Inversion Reduction)**.** *Fix a language $\mathcal{L}$. A decision-to-inversion reduction for $\mathcal{L}$ is a PPT machine $R^{(\cdot)}$, a function family $\{f_x\}_{x\in\{0,1\}^*}$, and a polynomial $p(\cdot)$ such that for any non-uniform PPT $\mathcal{A}$ and sufficiently large[12] x such that $\mathcal{A}(x,\cdot)$ inverts f_x with probability at least $\frac{1}{p(|x|)}$, $\Pr\left[R^{\mathcal{A}}(x) = \mathcal{L}(x)\right] \geq 1 - \frac{1}{p(|x|)}$.*

Lemma 2 (Implicit in proof of [LMP24] Lemma 3.1)**.** *Let $(\mathcal{L},\mathcal{D}) \notin \mathsf{ioAvgBPP/poly}$. If there exists a decision-to-inversion reduction for $\mathcal{L}$ with $p(n) = \omega(1)$, then one-way functions exist.*

Since Lemma 2 is not stated in exactly this form in [LMP24], we include a formal proof of it in Appendix B, noting that the proof essentially mirrors what already appears in [LMP24].

4 OWF From Non-Trivial NIZK

We begin with the simpler case of NIZKs.

Theorem 7. *Suppose there exists a language $\mathcal{L} \notin \mathsf{ioP/poly}$ such that $\mathcal{L}$ has an $(\varepsilon_{\mathsf{c}},\varepsilon_{\mathsf{s}},\varepsilon_{\mathsf{zk}})$-NIZK with $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} <_n 1$. Then there exists an auxiliary-input one-way function.*

Proof. Let $(\mathsf{Gen},\mathsf{P},\mathsf{V})$ be the $(\varepsilon_{\mathsf{c}},\varepsilon_{\mathsf{s}},\varepsilon_{\mathsf{zk}})$-NIZK for $\mathcal{L}$ guaranteed by the theorem statement, and $p(n)$ be a polynomial such that for sufficiently large n, $\varepsilon_{\mathsf{c}}(n)+\varepsilon_{\mathsf{s}}(n)+\varepsilon_{\mathsf{zk}}(n) < 1-\frac{1}{p(n)}$.[13] Suppose for the sake of contradiction that auxiliary-input one-way functions don't exist. Define:

- M as an auxiliary-input sampler where $M(x;r)$ computes $(\mathsf{crs},\pi) \leftarrow \mathsf{Sim}(x;r)$ and outputs crs.
- N as the io-UE machine guaranteed by Theorem 5 with respect to sampler M and polynomial $20p(n)$.
- Γ as the (infinite) set of n such that the guarantees of N hold.

We now claim that Algorithm 1 decides $\mathcal{L}$ infinitely often, contradicting that $\mathcal{L} \notin \mathsf{ioP/poly}$.

To analyze this algorithm, we first consider what it does when $x \in \{0,1\}^n - \mathcal{L}$. In this case, $\mathcal{A}$ can only accept if it finds an accepting proof for x relative to an honestly-generated CRS. Since $\mathcal{A}$ is efficient, soundness tells us that this happens with probability at most $\varepsilon_{\mathsf{s}}(n)$.

Next, we consider what $\mathcal{A}$ does when $x \in \{0,1\}^n \cap \mathcal{L}$ for some $n \in \Gamma$. Let p_{crs} be the probability that a single iteration of the for loop in $\mathcal{A}$ accepts. We call crs "bad" if $p_{\mathsf{crs}} \leq \frac{1}{20p(n)}$, and "good" if $p_{\mathsf{crs}} \geq \frac{1}{10p(n)}$. The core of our analysis comes from the following claim:

Claim 1. *Suppose $x \in \{0,1\}^n \cap \mathcal{L}$ for some $n \in \Gamma$. Then $\Pr[\mathsf{crs}$ is bad $\mid \mathsf{crs} \leftarrow \mathsf{Gen}(1^n)] \leq \varepsilon_{\mathsf{zk}}(n) + \varepsilon_{\mathsf{c}}(n) + \frac{1}{4p(n)}$.*

Algorithm 1: Algorithm $\mathcal{A}$ to decide $\mathcal{L}$

Input: $x \in \{0,1\}^n$
Output: Decision bit b, with $b = 1$ representing $x \in \mathcal{L}$

Compute $\mathsf{crs} \leftarrow \mathsf{Gen}(1^n)$
for $_ \leftarrow 1$ **to** $20n \cdot p(n)$ **do**
  Compute $r' \leftarrow N(x, \mathsf{crs})$
  Compute $(\mathsf{crs}', \pi') \leftarrow \mathsf{Sim}(x; r')$
  if $\mathsf{V}(\mathsf{crs}, x, \pi') = 1$ **then**
    Output $b = 1$
  end
end
Output $b = 0$

Algorithm 2: Distinguisher $\mathcal{D}$

Input: $x \in \{0,1\}^n$, (crs, π)
Output: Decision bit b, with $b = 0$ representing that (crs, π) was generated by Sim

for $_ \leftarrow 1$ **to** $20n \cdot p(n)$ **do**
  Compute $r' \leftarrow N(x, \mathsf{crs})$
  Compute $(\mathsf{crs}', \pi') \leftarrow \mathsf{Sim}(x; r')$
  Run $\mathsf{V}(\mathsf{crs}, x, \pi')$
end
Let count be the number of iterations where V accepted
if *count* $\leq 1.5n$ *and* $\mathsf{V}(x, \mathsf{crs}, \pi) = 1$ **then**
  Output $b = 1$
else
  Output $b = 0$
end

Proof. Consider the distinguisher $\mathcal{D}$ defined by Algorithm 2. We first claim that if $\mathcal{D}$ is given (crs, π) drawn from $\mathsf{Sim}(x)$, it will output 1 with probability at most $\frac{1}{5p(n)}$. First note that if we change π to instead be the proof we get from using randomness $r' \leftarrow N(x, \mathsf{crs})$, our guarantees on N tell us that this probability can change by at most $\frac{1}{20p(n)}$. Now there are two cases to consider: either crs is good, or it is not good.

In the case where crs is good, we note that each iteration of the for loop has $\mathsf{V}(\mathsf{crs}, x, \pi') = 1$ with probability at least $\frac{1}{10p(n)}$. Thus, bound 2 in Theorem 6 (with respect to $\delta = \frac{1}{4}$) tells us that the final value of count can be below $1.5n$ with probability at most $\exp(-n/16)$, which is at most $\frac{1}{20p(n)}$ for sufficiently large n.

In the case where crs is not good, we know that the probability that we get an accepting proof by computing $r' \leftarrow N(x, \mathsf{crs})$ and $(\mathsf{crs}', \pi') \leftarrow \mathsf{Sim}(x; r')$ is at most $\frac{1}{10p(n)}$. But note that this is exactly how our given proof is computed! Since $\mathcal{D}$ can only output 1 if $\mathsf{V}(x, \mathsf{crs}, \pi) = 1$, we have that when crs is not good this happens with probability at most $\frac{1}{10p(n)}$. Combining all of the above, we get that $\mathcal{D}$ outputs 1 on a simulated (crs, π) with probability at most $\frac{1}{5p(n)}$ as desired.

Now we turn to what $\mathcal{D}$ does when given (crs, π) generated by Gen and P. We claim here that it will output 1 with probability at least $\Pr[\mathsf{crs} \text{ is bad} \mid \mathsf{crs} \leftarrow \mathsf{Gen}(1^n)] - \varepsilon_{\mathsf{c}}(n) - \frac{1}{20p(n)}$. Indeed, there are only two ways for $\mathcal{D}$ to output 0: either $\mathsf{V}(x, \mathsf{crs}, \pi)$ rejects, or the final value of count is at most $1.5n$. The former can happen with probability at most $\varepsilon_{\mathsf{c}}(n)$ by completeness. For the latter, if crs is bad, we know that each iteration of the for loop has $\mathsf{V}(\mathsf{crs}', x, \pi') = 1$ with probability at most $\frac{1}{20p(n)}$. Pessimistically assuming that this probability is *exactly* $\frac{1}{20p(n)}$, bound 1 of Theorem 6 (with respect to $\delta = \frac{1}{2}$) tells us that the final value of count can exceed $1.5n$ with probability at most $\exp(-n/12)$; for sufficiently large n, this can be bounded by $\frac{1}{20p(n)}$.

Putting this all together, we have that $\mathcal{D}$ distinguishes a real (crs, π) from a simulated one with advantage at least $\Pr[\mathsf{crs} \text{ is bad} \mid \mathsf{crs} \leftarrow \mathsf{Gen}(1^n)] - \varepsilon_{\mathsf{c}}(n) - \frac{1}{4p(n)}$. But by zero-knowledge, we know that this advantage can be at most $\varepsilon_{\mathsf{zk}}(n)$. Rearranging terms gives the desired claim. □

We can now argue that $\mathcal{A}$ must accept $x \in \mathcal{L}$ with good probability. Suppose that crs is not bad. Then each iteration of the for loop must have $\mathsf{V}(\mathsf{crs}, x, \pi') = 1$ with probability at least $\frac{1}{20p(n)}$. Thus, the probability that *no* iteration of the for loop outputs 1 is at most $\left(1 - \frac{1}{20p(n)}\right)^{20n \cdot p(n)} \leq \exp(-n)$, which for sufficiently large n is at most $\frac{1}{4p(n)}$. Thus, the probability that $\mathcal{A}$ accepts $x \in \{0,1\}^n \cap \mathcal{L}$ for $n \in \Gamma$ is at least the probability that crs is not bad minus $\frac{1}{4p(n)}$; plugging in Claim 1, this is at least $1 - \varepsilon_{\mathsf{zk}}(n) - \varepsilon_{\mathsf{c}}(n) - \frac{1}{2p(n)}$.

Finally, this all means that we can instantiate Definition 7 with respect to $a(n) = 1 - \varepsilon_{\mathsf{zk}}(n) - \varepsilon_{\mathsf{c}}(n) - \frac{1}{2p(n)}$ and $b(n) = \varepsilon_{\mathsf{s}}(n)$. The above shows that we satisfy the first two points of the definition for all $n \in \Gamma$. For sufficiently large n, we know that $\varepsilon_{\mathsf{c}}(n) + \varepsilon_{\mathsf{s}}(n) + \varepsilon_{\mathsf{zk}}(n) < 1 - \frac{1}{p(n)}$, so $a(n) - b(n) > \frac{1}{2p(n)}$, satisfying the third point. Thus, we have that $\mathcal{L} \in \mathsf{ioBPP/poly}$, so by Remark 6, $\mathcal{L} \in \mathsf{ioP/poly}$. □

[12] Note here that we explicitly allow the notion of "sufficiently large" to depend on $\mathcal{A}$.

[13] In what follows, it will help to assume WLOG that $p(n) = \omega(n^2)$.

Remark 7. *Theorem 7 can also be phrased in the following way: if $\mathcal{L}$ has an $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-NIZK with $\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{s}}+\varepsilon_{\mathsf{zk}} <_n 1$, then $\mathcal{L}$ has a decision-to-inversion reduction (Definition 10). Indeed, all we need to prove the correctness of Algorithm 1 is for N to satisfy the io-UE guarantees on auxiliary input x. Additionally, we note that the proof of Theorem 5 constructs a candidate auxiliary-input one-way function $\{f_x\}_x$ and reduces inverting f_x to satisfying io-UE guarantees on auxiliary input x. Putting these two together gives the desired reduction R.*[14]

Given the above remark, the techniques from [HN24, LMP24] immediately allow us to improve Theorem 7 from an *auxiliary-input* one-way function to a standard one-way function.

Corollary 3. *Suppose that every $\mathcal{L} \in \mathsf{NP}$ has an $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-NIZK with $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} <_n 1$ and $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$. Then OWFs exist.*

Proof. Since $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$, we know there is a language $\mathcal{L} \in \mathsf{NP}$ (that thus has an appropriate NIZK) where $\mathcal{L} \notin \mathsf{ioP/poly}$. Theorem 7 thus tells us that there exists an auxiliary-input one-way function. We can then plug this into Lemma 1 to get a new language $\mathcal{L}' \in \mathsf{NP}$ and a polynomial $m(\cdot)$ such that $(\mathcal{L}', \{U_{m(n)}\}_{n\in\mathbb{N}}) \notin \mathsf{ioAvgBPP/poly}$. But since $\mathcal{L}' \in \mathsf{NP}$ (and thus has an appropriate NIZK), Remark 7 tells us that there is a decision-to-inversion reduction for $\mathcal{L}'$. Hence, Lemma 2 tells us that one-way functions exist. □

5 OWF From Non-Trivial Public-Coin ZK

We now turn to the more general case of interactive zero-knowledge.

Theorem 8. *Suppose there exists a language $\mathcal{L} \notin \mathsf{ioP/poly}$ such that $\mathcal{L}$ has a constant-round, public coin $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK argument with $\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{s}}+\varepsilon_{\mathsf{zk}} <_n 1$. Then there exists an auxiliary-input one-way function.*

Before proving this theorem, it will help to define some tools that will assist us in the proof. In all of the sections below, let

- $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ be the $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK argument for $\mathcal{L}$ guaranteed by the theorem statement with simulator Sim,
- k be the (constant) number of rounds of interaction in this argument,
- $p(n)$ be a polynomial such that for sufficiently large n, $\varepsilon_{\mathsf{c}}(n)+\varepsilon_{\mathsf{s}}(n)+\varepsilon_{\mathsf{zk}}(n) < 1-\frac{1}{p(n)}$,
- $p_{\mathsf{Est}}(n)$ as the polynomial $256k^2 \cdot n \cdot p(n)^2$,
- M as an auxiliary-input sampler where $M(x; r)$ interprets $r = r_1 \| r_2$ for $r_1 \in [k]$ and outputs the first $r_1 - 1$ rounds of the transcript generated by $\mathsf{Sim}(x; r_2)$,

[14] Technically, Algorithm 1 doesn't quite give a decision-to-inversion reduction as written, since we are only guaranteed a polynomial gap between the probability of accepting $x \in \mathcal{L}$ versus $x \notin \mathcal{L}$. However, standard BPP amplification can convert this into a reduction that outputs the correct answer with high probability.

- N be the io-UE machine guaranteed (assuming auxiliary-input one-way functions don't exist) by Theorem 5 with respect to sampler M and polynomial $8k^2 \cdot p(n)$, and
- Γ as the (infinite) set of n such that the guarantees of N hold.

5.1 Prover $\widetilde{\mathsf{P}}$

We start by defining a malicious prover strategy $\widetilde{\mathsf{P}}$ for $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$. Intuitively, $\widetilde{\mathsf{P}}$ will, given the current partial transcript of interaction with V, use N to sample "many" possible next prover messages. In order to choose which to use, $\widetilde{\mathsf{P}}$ will estimate the probability of each leading to an accepting transcript if it continues to use this strategy in future rounds, and pick the one with the largest success probability. We formally define the behavior of $\widetilde{\mathsf{P}}$ in Algorithm 3; we've separated out the definition of the success probability estimation procedure Est as Algorithm 4, as this will see use elsewhere.

Algorithm 3: Malicious Prover $\widetilde{\mathsf{P}}$

Input: $x \in \{0,1\}^n$, partial transcript τ with P sending the next message
Output: m as the next message to send

Let i be the number of rounds already completed in τ
for $_ \leftarrow 1$ **to** $16k \cdot n \cdot p(n)$ **do**
  Compute $r' \leftarrow N(x, \tau)$
  Parse $r' = r'_1 \| r'_2$
  Compute τ' as the first $i+1$ messages of $\mathsf{Sim}(x; r'_2)$
  Compute $\mathsf{Est}(x, \tau')$
end
Let τ^* be the transcript with the highest value of $\mathsf{Est}(x, \tau')$ sampled above
Output m^* as the $(i+1)$st message in τ^*

Algorithm 4: Success Probability Estimator Est

Input: $x \in \{0,1\}^n$, (possibly partial) transcript τ
Output: Estimate of the probability that V will accept when interacting with $\widetilde{\mathsf{P}}$ starting on transcript τ

for $_ \leftarrow 1$ **to** $12 p_{\mathsf{Est}}(n)^4$ **do**
  Simulate[15] interaction between V and $\widetilde{\mathsf{P}}$ starting from partial transcript τ
end
Output fraction of iterations where V accepted

Remark 8. *We note that while the definitions of $\widetilde{\mathsf{P}}$ and Est refer to each other, $\widetilde{\mathsf{P}}$ always calls Est on a strictly longer transcript than it started with, so these definitions are not circular.*

[15] We note that this simulation is possible because V is public-coin, and thus has no internal state. In fact, this is the *only* reason we need to limit our proof to public-coin protocols.

We now prove some useful properties of $\widetilde{\mathsf{P}}$ and Est.

Lemma 3. *$\widetilde{\mathsf{P}}$ and Est run in polynomial time.*

Proof. Define $T_{\widetilde{\mathsf{P}}}(i)$ as the run time of $\widetilde{\mathsf{P}}$ when it is given a transcript with i messages in it. We note that this is a decreasing function, as $\widetilde{\mathsf{P}}$ needs to do less work estimating its success probability when the transcript is closer to completion.

Our first claim is that there exists a polynomial $q(\cdot,\cdot)$ such that $T_{\widetilde{\mathsf{P}}}(i) \leq q(k,n) \cdot T_{\widetilde{\mathsf{P}}}(i+2)$ and $T_{\widetilde{\mathsf{P}}}(k-1) \leq q(k,n)$. Indeed, ignoring various (additive) polynomial overheads, when $\widetilde{\mathsf{P}}$ is given a transcript with i messages, it simply needs to run Est on transcripts with $i+1$ messages a total of $16k \cdot n \cdot p(n)$ times. Each call to Est needs to simulate $12p_{\mathsf{Est}}(n)^4$ interactions between $\widetilde{\mathsf{P}}$ and V, each of which needs to run $\widetilde{\mathsf{P}}$ at most k times, each time starting with a transcript that has at least $i+2$ messages. Combining these two statements, we get the first part of our claim. For the second part, we note that Est on a full transcript simply needs to check if V accepts $12p_{\mathsf{Est}}(n)^4$ times, each of which takes polynomial time independent of $\widetilde{\mathsf{P}}$. Since $\widetilde{\mathsf{P}}$ on input a transcript with $k-1$ messages simply needs to run Est on a full transcript $16k \cdot n \cdot p(n)$ times, we get the second half of the claim.

Given this claim, we can say that $T_{\widetilde{\mathsf{P}}}(0) \leq q(k,n)^{O(k)}$. Since k is a constant[16] and $T_{\widetilde{\mathsf{P}}}$ is a decreasing function, we have that $\widetilde{\mathsf{P}}$ always runs in polynomial time. Furthermore noting that Est only ever needs to run $\widetilde{\mathsf{P}}$ a polynomial number of times (along with some other polynomial additive overhead), we have that Est also always runs in polynomial time. □

Lemma 4. *On any input τ, the output of* Est *is within an additive distance of $\frac{1}{p_{\mathsf{Est}}(n)}$ of the true acceptance probability with probability at least $1 - \frac{1}{p_{\mathsf{Est}}(n)}$.*

Proof. Let p_τ

be the true acceptance probability given that $\widetilde{\mathsf{P}}$ and V start with the transcript τ. An application of bound 3 in Theorem 6 (with respect to $\Delta = 12p_{\mathsf{Est}}(n)^3$) tells us that

$$\begin{aligned}\Pr\left[|\mathsf{Est}(x,\tau) - p_\tau| \geq \frac{1}{p_{\mathsf{Est}}(n)}\right] &\leq 2\exp\left(-\frac{144p_{\mathsf{Est}}(n)^6}{12p_{\mathsf{Est}}(n)^4}\right) \\ &= 2\exp\left(-12p_{\mathsf{Est}}(n)^2\right) \leq \frac{1}{p_{\mathsf{Est}}(n)}\end{aligned}$$

□

Before stating our next property, we need to introduce some notation.

Definition 11. *Fix a partial transcript τ where the prover is going next, and let i be the number of messages in τ. For any possible next prover message m, let p_m be the probability that V would accept if it interacted with $\widetilde{\mathsf{P}}$ starting on the transcript $\tau \| m$. We define $q_{threshold}$ as the threshold*

[16] We note that this is the only reason we need to limit our proof to constant round protocols.

such that $q_{threshold} \geq p_{m'}$ *for at least* $1 - \frac{1}{16k \cdot p(n)}$ *fraction (by probability) of messages* m' *sampled using* N*. More formally, let* $q_{threshold}$ *be the smallest value such that*

$$\Pr\left[q_{threshold} \geq p_{m'} \;\middle|\; \begin{array}{c} r' = r'_1 \| r'_2 \leftarrow N(x, \tau) \\ \tau' \leftarrow \mathsf{Sim}(x; r'_2) \\ m' = \textit{the } (i+1)\textit{st message in } \tau' \end{array}\right] > 1 - \frac{1}{16k \cdot p(n)}$$

We call m *"okay" if* $p_m \geq q_{threshold} - \frac{2}{p_{\mathsf{Est}}(n)}$*, "good" if* $p_m \geq q_{threshold}$*, and "very good" if* $p_m \geq q_{threshold} + \frac{2}{p_{\mathsf{Est}}(n)}$*.*

Lemma 5. *Let* τ *be any partial transcript where the prover sends the next message. If we run* $\widetilde{\mathsf{P}}(x, \tau)$ *to get a next prover message* m*, then* m *is "okay" with probability at least* $1 - \frac{1}{16k \cdot p(n)} - e^{-n}$*.*

Proof. We note that $\widetilde{\mathsf{P}}$ makes $16k \cdot n \cdot p(n)$ calls to Est. By Lemma 4 and a union bound, we know that the probability that any of these calls is more than $\frac{1}{p_{\mathsf{Est}}(n)}$ away from the true success probability is at most $\frac{16k \cdot n \cdot p(n)}{p_{\mathsf{Est}}(n)} = \frac{1}{16k \cdot p(n)}$.

But we now note that since each iteration of the for loop in $\widetilde{\mathsf{P}}$ samples from exactly the distribution of messages used in Definition 11, each iteration chooses a "good" message with probability at least $\frac{1}{16k \cdot p(n)}$. Thus, the probability that *no* iteration chooses a "good" message is at most $(1 - \frac{1}{16k \cdot p(n)})^{16k \cdot n \cdot p(n)} \leq e^{-n}$.

Now note that if neither of the above bad events happen, the message m^* output by $\widetilde{\mathsf{P}}$ must be "okay".[17] Thus, we have that $\widetilde{\mathsf{P}}$ outputs an "okay" message with probability at least $1 - \frac{1}{16k \cdot p(n)} - e^{-n}$, as desired. □

5.2 Distinguisher $\mathcal{D}$

We next define a distinguisher $\mathcal{D}$ that we will use in our proof. Intuitively, $\mathcal{D}$ will check each prover message to see if it is "much better" than "most" next messages that we would get by using N. In our proof, we will use the fact that this distinguisher can only have $\varepsilon_{\mathsf{zk}}(n)$ advantage in distinguishing a real from a simulated transcript in order to argue that our algorithm cannot lose too much success probability. We formally define $\mathcal{D}$ as Algorithm 5.

We now prove some useful properties of $\mathcal{D}$.

Lemma 6. *For large enough* $n \in \Gamma$ *and* $x \in \{0,1\}^n$*,* $\Pr[\mathcal{D}(x, \tau) = 1 \mid \tau \leftarrow \mathsf{Sim}(x)] \leq \frac{1}{4p(n)}$*.*

Proof. We first define a sequence of hybrids for how we generate τ. For $i \in [k]_0$, let $\mathrm{Hyb}_i(x)$ be defined as in Algorithm 6. We note immediately that Hyb_k is identical to Sim. Furthermore, the guarantees on N tell us that $\mathrm{Hyb}_i(x)$ has statistical distance at most $\frac{1}{8k \cdot p(n)}$ from $\mathrm{Hyb}_{i+1}(x)$.[18] Thus, we have that $\mathsf{Sim}(x)$ has statistical distance at most $\frac{1}{8p(n)}$ from $\mathrm{Hyb}_0(x)$.

[17] In the worst case, we could underestimate the success probability of every "good" message we sampled by $\frac{1}{p_{\mathsf{Est}}(n)}$, and choose a different message due to overestimating its success probability by $\frac{1}{p_{\mathsf{Est}}(n)}$.

[18] Note that this loses a factor of k compared to the overall guarantee on N since we are specifically asking N to work on a partial transcript with i messages, where M outputs such a transcript with probability $\frac{1}{k}$.

Algorithm 5: Distinguisher $\mathcal{D}$

Input: $x \in \{0,1\}^n$, (full) transcript τ
Output: Decision bit b, with $b = 0$ representing that τ was generated by Sim

for *each* P *round* i **do**
  Let τ_i be the first i rounds of τ and τ_{i-1} the first $i-1$ rounds
  Compute $\mathsf{Est}(x, \tau_i)$
  for $_ \leftarrow 1$ **to** $8k \cdot n \cdot p(n)$ **do**
    Compute $r' \leftarrow N(x, \tau_{i-1})$
    Parse $r' = r'_1 \| r'_2$
    Compute τ' as τ_{i-1} concatenated with the ith message of $\mathsf{Sim}(x; r'_2)$
    Compute $\mathsf{Est}(x, \tau')$
  end
  Let count be the number of iterations where $\mathsf{Est}(x, \tau') \geq \mathsf{Est}(x, \tau_i)$
  if *count* $< n$ **then**
    Output $b = 1$
  end
end
Output $b = 0$

Algorithm 6: Hybrid Transcript Generation Hyb_i

Input: $x \in \{0,1\}^n$
Output: (Full) transcript τ

Let τ be the first i rounds of $\mathsf{Sim}(x)$
for $j \leftarrow i+1$ **to** k **do**
  Compute $r' \leftarrow N(x, \tau)$
  Parse $r' = r'_1 \| r'_2$
  Append the jth message in $\mathsf{Sim}(x; r'_2)$ to τ
end
Output τ

To complete the proof, we now claim that $\Pr[\mathcal{D}(x,\tau)=1 \mid \tau \leftarrow \mathrm{Hyb}_0(x)] \leq \frac{1}{8p(n)}$. Indeed, consider the iteration of $\mathcal{D}$ corresponding to prover round i. We note that because τ was sampled from Hyb_0, τ_i is drawn from the same distribution as each τ' that $\mathcal{D}$ samples in this iteration. Thus, if we order τ_i and each sampled τ' by their respective values of Est (breaking ties randomly), they will be in a random order. Since it is only possible for $\mathcal{D}$ to output 1 in this iteration if τ_i is in the top n of this order, we have that the probability of this happening is at most $\frac{n}{8k\cdot n\cdot p(n)} = \frac{1}{8k\cdot p(n)}$. Taking a union bound over all prover rounds, we get our desired claim. □

Lemma 7. *Let τ be a (full) transcript and $n = |x|$ sufficiently large. If any prover message in τ is "very good" (as in Definition 11, with respect to the partial transcript up to that message), $\mathcal{D}(x,\tau)$ will output $b = 1$ in the iteration corresponding to that round with probability at least $1 - \frac{1}{16k\cdot p(n)}$.*

Proof. Let i be a prover round where the prover's message m_i is "very good". We first note that the corresponding iteration of $\mathcal{D}$ only makes $8k \cdot n \cdot p(n) + 1$ calls to Est. Thus, by Lemma 4 and a union bound, we have that all values returned by Est are within a $\frac{1}{p_{\mathsf{Est}}(n)}$ additive distance of their corresponding correct values with probability at least $1 - \frac{8k\cdot n\cdot p(n)}{p_{\mathsf{Est}}(n)} = 1 - \frac{1}{32k\cdot p(n)} - \frac{1}{p_{\mathsf{Est}}(n)}$. For the remainder of the proof, we will assume this is the case.

Since every call to Est is within an additive $\frac{1}{p_{\mathsf{Est}}(n)}$ distance of the true acceptance probability, we have that for any τ' sampled in the iteration of $\mathcal{D}$ corresponding to round i, we can only have $\mathsf{Est}(x,\tau') \geq \mathsf{Est}(x,\tau_i)$ if $p_{m'} \geq p_{m_i} - \frac{2}{p_{\mathsf{Est}}(n)}$, where m' is the ith message of τ'. But since m_i is "very good", we know that this happens with probability at most $\frac{1}{16k\cdot p(n)}$ over the sampling of τ'. Pessimistically assuming that this probability is exactly $\frac{1}{16k\cdot p(n)}$, bound 1 in Theorem 6 (with respect to $\delta = \frac{1}{2}$) will tell us that the final value of count can only exceed $\frac{3n}{4}$ with probability at most $\exp(-n/24)$; for sufficiently large n, this is at most $\frac{1}{p_{\mathsf{Est}}(n)}$. Since $\mathcal{D}$ will output 1 if count $< n$ (and so in particular if count $\leq \frac{3n}{4}$), this tells us that as long as no call to Est fails, $\mathcal{D}$ will output 1 with probability at least $1 - \frac{1}{p_{\mathsf{Est}}(n)}$.

Applying a union bound to the above two paragraphs, we get that overall $\mathcal{D}$ outputs 1 with probability at least $1 - \frac{1}{32k\cdot p(n)} - \frac{2}{p_{\mathsf{Est}}(n)} \geq 1 - \frac{1}{16k\cdot p(n)}$ as desired. □

5.3 Algorithm $\mathcal{A}$

We can now describe our algorithm for deciding $\mathcal{L}$, as described in Algorithm 7.

Lemma 8. *For all $x \in \{0,1\}^n - \mathcal{L}$, $\Pr[\mathcal{A}(x) = 1] \leq \varepsilon_{\mathsf{s}}(n)$.*

Proof. By Lemma 3, we know that $\widetilde{\mathsf{P}}$ is an efficient prover strategy. Thus, soundness tells us that the probability that V accepts when interacting with $\widetilde{\mathsf{P}}$ on $x \notin \mathcal{L}$ is at most $\varepsilon_{\mathsf{s}}(|x|)$. □

Lemma 9. *For all sufficiently large $n \in \Gamma$ and all $x \in \{0,1\}^n \cap \mathcal{L}$, $\Pr[\mathcal{A}(x) = 1] \geq 1 - \varepsilon_{\mathsf{c}}(n) - \varepsilon_{\mathsf{zk}}(n) - \frac{1}{2p(n)}$.*

Algorithm 7: Algorithm $\mathcal{A}$ for $\mathcal{L}$

Input: $x \in \{0,1\}^n$
Output: Decision bit b

Simulate an interaction between $\widetilde{\mathsf{P}}$ and V on input x
if V *accepts* **then**
  Output $b = 1$
else
  Output $b = 0$
end

Proof. Fix n to be large enough that the guarantees of all the above lemmas hold, and x as any element in $\{0,1\}^n \cap \mathcal{L}$. We define a sequence of hybrid experiments as follows. For each $i \in [k]_0$, let $\mathrm{Hyb}^1_i(x)$:

- Create a transcript τ by running the honest P and V on x for the first i rounds, and completing the remaining rounds by running $\widetilde{\mathsf{P}}$ and V.[19]
- Run $\mathcal{D}(x, \tau)$. For every $j \in [k]_0$, let R_j be the event that $\mathcal{D}$ outputs 1 in or before the iteration corresponding to round j.

To simplify notation in our proof, for each $i \in [k]$, we also define $\mathrm{Hyb}^2_i(x)$ as:

- Create a transcript τ by running the honest P and V on x for the first i rounds, and completing the remaining rounds by running $\widetilde{\mathsf{P}}$ and V.
- Run $\mathcal{D}(x, \tau)$. For every $j \in [k]_0$, let R_j be the event that $\mathcal{D}$ outputs 1 in or before the iteration corresponding to round j.
- Create a second transcript τ' by taking the first $i-1$ rounds of τ and completing the remaining rounds by running $\widetilde{\mathsf{P}}$ and V.

Note that Hyb^1_0 (ignoring $\mathcal{D}$) is exactly the same as $\mathcal{A}$, so our goal is to prove that V is unlikely to reject in that hybrid. On the other hand, Hyb^1_k involves generating an honest transcript, which we know can only be rejected with probability at most $\varepsilon_{\mathsf{c}}(n)$. Thus, it suffices for us to show that as we move from Hyb^1_k to Hyb^1_0[20] the probability that V rejects cannot increase by much. Specifically, we aim to prove

Claim 2. *Fix any $i \in [k]$. Then we have that*

$$\Pr_{\mathrm{Hyb}^1_{i-1}}[\mathsf{V}(x,\tau) = 0 \wedge \overline{R_{i-1}}] \leq \Pr_{\mathrm{Hyb}^1_i}[\mathsf{V}(x,\tau) = 0 \wedge \overline{R_i}] + \frac{1}{4k \cdot p(n)} + \Pr_{\mathrm{Hyb}^2_i}[\overline{R_{i-1}} \wedge R_i]$$

[19] As in our definition of $\widetilde{\mathsf{P}}$, this is possible because V is stateless.
[20] The hybrids count down for ease of notation.

Indeed, by repeated application of the claim, we get that

$$\begin{aligned}\Pr[\mathcal{A}(x)=0] &= \Pr_{\mathrm{Hyb}_0^1}[\mathsf{V}(x,\tau)=0] = \Pr_{\mathrm{Hyb}_0^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_0}] \\ &\leq \Pr_{\mathrm{Hyb}_k^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_k}] + \frac{1}{4p(n)} + \sum_{i\in[k]} \Pr_{\mathrm{Hyb}_i^2}[\overline{R_{i-1}} \wedge R_i] \\ &\leq \Pr_{\mathrm{Hyb}_k^1}[\mathsf{V}(x,\tau)=0] + \frac{1}{4p(n)} + \sum_{i\in[k]} \Pr_{\mathrm{Hyb}_i^2}[\overline{R_{i-1}} \wedge R_i] \end{aligned} \tag{1}$$

(For the second equality, we used the fact that R_0 is always false.) The first term is at most $\varepsilon_{\mathsf{c}}(n)$ as noted above. For the last term, we note that the events R_i and R_{i-1} are defined over the first i messages of τ, which are sampled using the honest P and V. Thus, we have that $\Pr_{\mathrm{Hyb}_i^2}[\overline{R_{i-1}} \wedge R_i] = \Pr_{\mathrm{real}}[\overline{R_{i-1}} \wedge R_i]$, where real is the probability space where we sample the entire transcript τ from P and V, then run $\mathcal{D}(x,\tau)$. Since $R_{i-1} \subseteq R_i$ for all $i \in [k]$, the resulting sum telescopes; that is, $\sum_{i\in[k]} \Pr_{\mathrm{real}}[\overline{R_{i-1}} \wedge R_i] = \Pr_{\mathrm{real}}[\overline{R_0} \wedge R_k]$, which by the definition of R_0 and R_k is just $\Pr_{\mathrm{real}}[\mathcal{D}(x,\tau)=1]$. Plugging this all back into (1), we get

$$\begin{aligned}\Pr[\mathcal{A}(x)=0] &\leq \varepsilon_{\mathsf{c}}(n) + \Pr_{\mathrm{real}}[\mathcal{D}(x,\tau)=1] + \frac{1}{4p(n)} \\ &\leq \varepsilon_{\mathsf{c}}(n) + \varepsilon_{\mathsf{zk}}(n) + \Pr[\mathcal{D}(x,\tau)=1 \mid \tau \leftarrow \mathsf{Sim}(x)] + \frac{1}{4p(n)} \\ &\leq \varepsilon_{\mathsf{c}}(n) + \varepsilon_{\mathsf{zk}}(n) + \frac{1}{2p(n)}\end{aligned}$$

where the first inequality comes from zero-knowledge and the second comes from Lemma 6. All that now remains is to prove the claim. We do so by proving the following two claims.

Claim 3. *Fix any $i \in [k]$. Then we have that*

$$\Pr_{\mathrm{Hyb}_{i-1}^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_{i-1}}] \leq \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] + \Pr_{\mathrm{Hyb}_i^2}[\overline{R_{i-1}} \wedge R_i]$$

Claim 4. *Fix any $i \in [k]$. Then we have that*

$$\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] \leq \Pr_{\mathrm{Hyb}_i^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i}] + \frac{1}{4k \cdot p(n)}$$

Note that Claim 2 trivially follows from Claim 3 and Claim 4.

Proof of Claim 3. We first note that

$$\Pr_{\mathrm{Hyb}_{i-1}^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_{i-1}}] = \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_{i-1}}]$$

Indeed, in both cases we are checking if V accepts a transcript sampled by taking the first $i-1$ messages from P and V with the remainder from $\widetilde{\mathsf{P}}$ and V, as well as if $\mathcal{D}$ outputs 1

within the first $i-1$ messages of that transcript. But now we can write

$$\begin{aligned}\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_{i-1}}] &= \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] + \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_{i-1}} \wedge R_i] \\ &\leq \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] + \Pr_{\mathrm{Hyb}_i^2}[\overline{R_{i-1}} \wedge R_i]\end{aligned}$$

which yields the desired claim. □

Proof of Claim 4. First note that if i is a verifier round, we have that

$$\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] = \Pr_{\mathrm{Hyb}_i^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i}]$$

Indeed, the only place where these two hybrids could differ is at round i. But note that $\overline{R_i}$ is not affected by this difference, since $\mathcal{D}$ only looks at prover rounds. Additionally, τ and τ' are identically distributed (both use V to sample the ith message), so the event that V rejects is also unaffected. This means that the statement trivially holds when i is a verifier round, so for the remainder of the proof we will assume that i is a prover round.

Recalling the definitions of "okay" and "very good" from Definition 11, we have that

$$\begin{aligned}\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] = \quad & \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i} \wedge m_i \text{ is "very good"}] \\ & + \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i} \wedge m_i \text{ is not "very good"} \wedge m_i' \text{ is "okay"}] \\ & + \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i} \wedge m_i \text{ is not "very good"} \wedge m_i' \text{ is not "okay"}] \\ \leq \quad & \Pr_{\mathrm{Hyb}_i^2}[\overline{R_i} \wedge m_i \text{ is "very good"}] \\ & + \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i} \wedge m_i \text{ is not "very good"} \wedge m_i' \text{ is "okay"}] \\ & + \Pr_{\mathrm{Hyb}_i^2}[m_i' \text{ is not "okay"}]\end{aligned}$$

where m_i is the ith message in τ and m_i' is the ith message in τ'. We note that the first term is at most $\frac{1}{16k \cdot p(n)}$ by Lemma 7; the third term is at most $\frac{1}{16k \cdot p(n)} + e^{-n}$ by Lemma 5. For the second term, we note that since m_i is not "very good" but m_i' is "okay", the probability that V rejects τ' can be at most $\frac{4}{p_{\mathsf{Est}}(n)}$ higher than the probability that V rejects τ. Thus, the second term can be bounded as

$$\begin{aligned}&\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i} \wedge m_i \text{ is not "very good"} \wedge m_i' \text{ is "okay"}] \\ &\leq \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i} \wedge m_i \text{ is not "very good"} \wedge m_i' \text{ is "okay"}] + \frac{4}{p_{\mathsf{Est}}(n)} \\ &\leq \Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i}] + \frac{4}{p_{\mathsf{Est}}(n)} = \Pr_{\mathrm{Hyb}_i^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i}] + \frac{4}{p_{\mathsf{Est}}(n)}\end{aligned}$$

where the equality comes from the fact that τ is sampled from the same distribution in the two hybrids. Plugging this all in, we get that

$$\Pr_{\mathrm{Hyb}_i^2}[\mathsf{V}(x,\tau')=0 \wedge \overline{R_i}] \leq \Pr_{\mathrm{Hyb}_i^1}[\mathsf{V}(x,\tau)=0 \wedge \overline{R_i}] + \frac{4}{p_{\mathsf{Est}}(n)} + \frac{1}{8k \cdot p(n)} + e^{-n}$$

Noting that $\frac{4}{p_{\mathsf{Est}}(n)} + \frac{1}{8k \cdot p(n)} + e^{-n} \leq \frac{1}{4k \cdot p(n)}$ for sufficiently large n, we get the desired claim. □

□

5.4 Putting It All Together

Proof of Theorem 8. Suppose for the sake of contradiction that ai-OWFs don't exist, and hence Γ is infinite. We then have that Algorithm 7 satisfies the requirements of Definition 7 with respect to $a(n) = 1 - \varepsilon_{\mathsf{c}}(n) - \varepsilon_{\mathsf{zk}}(n) - \frac{1}{2p(n)}$ and $b(n) = \varepsilon_{\mathsf{s}}(n)$. Indeed, Lemma 9 immediately tells us that we satisfy the first two points for infinitely many n. For the third point, we have that for sufficiently large n,

$$a(n) - b(n) = 1 - \varepsilon_{\mathsf{c}}(n) - \varepsilon_{\mathsf{zk}}(n) - \varepsilon_{\mathsf{s}}(n) - \frac{1}{2p(n)} \geq \frac{1}{2p(n)}$$

where the inequality comes from the definition of $p(n)$. Thus, we have that $\mathcal{L} \in \mathsf{ioBPP/poly}$, so by Remark 6, we have that $\mathcal{L} \in \mathsf{ioP/poly}$. This provides our contradiction. □

Remark 9. *Theorem 8 can also be phrased in the following way: if $\mathcal{L}$ has a constant-round, public-coin $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK argument with $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} <_n 1$, then $\mathcal{L}$ has a decision-to-inversion reduction (Definition 10). The reasoning for this is identical to that in Remark 7.*

Given the above remark, the techniques from [HN24, LMP24] immediately allow us to improve Theorem 8 from an *auxiliary-input* one-way function to a standard one-way function.

Corollary 4. *Suppose that every $\mathcal{L} \in \mathsf{NP}$ has a constant-round, public-coin $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$-ZK argument with $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} <_n 1$ and $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$. Then OWFs exist.*

Proof. Since $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$, we know there is a language $\mathcal{L} \in \mathsf{NP}$ (that thus has an appropriate argument) where $\mathcal{L} \notin \mathsf{ioP/poly}$. Theorem 8 thus tells us that there exists an auxiliary-input one-way function. We can then plug this into Lemma 1 to get a new language $\mathcal{L}' \in \mathsf{NP}$ and a polynomial $m(\cdot)$ such that $(\mathcal{L}', \{U_{m(n)}\}_{n \in \mathbb{N}}) \notin \mathsf{ioAvgBPP/poly}$. But since $\mathcal{L}' \in \mathsf{NP}$ (and thus has an appropriate argument), Remark 9 tells us that there is a decision-to-inversion reduction for $\mathcal{L}'$. Hence, Lemma 2 tells us that one-way functions exist. □

6 Applications to Amplification

Similar to [CHK25], we can plug the one-way function we get from a high-error zero-knowledge protocol into prior work in order to get (almost) unconditional amplification of these high-error protocols. Below, we separately consider the cases of non-interactive zero-knowledge and interactive zero-knowledge.

6.1 Non-Interactive Zero-Knowledge

In the non-interactive case, a recent line of works [GJS19, BKP+24, BG24, AK25] has explored what assumptions are needed in order to amplify (adaptively-sound) NIZKs with high errors to have negligible errors, with [BG24, AK25] only requiring one-way functions in certain cases. If we instantiate these results with the one-way function we get from Corollary 3, we get that we can amplify such NIZKs (almost) unconditionally. Formally, we recall the following theorem from [AK25], rephrased slightly to fit our terminology.

Theorem 9 ([AK25], Theorem 8.27)**.** *Suppose OWFs exist. Then if there exists an* $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$*-NIZK* proof *with adaptive and statistical soundness for a language* $\mathcal{L}$ *with (1)* ε_{c} *negligible and (2)* $\varepsilon_{\mathsf{s}} + \varepsilon_{\mathsf{zk}} <_n 1$*, then there exists an* $(\varepsilon'_{\mathsf{c}}, \varepsilon'_{\mathsf{s}}, \varepsilon'_{\mathsf{zk}})$*-NIZK with adaptive and statistical soundness for* $\mathcal{L}$ *with* $\varepsilon'_{\mathsf{c}}$*,* $\varepsilon'_{\mathsf{s}}$*, and* $\varepsilon'_{\mathsf{zk}}$ *all negligible.*

Combining this with Corollary 3 immediately gives us the following result.

Corollary 5. *Suppose that* $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$ *or* $\mathsf{NP} \subseteq \mathsf{BPP}$*. Suppose further that all of* NP *has* $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$*-NIZKs with adaptive and statistical soundness where (1)* ε_{c} *is negligible and (2)* $\varepsilon_{\mathsf{zk}} + \varepsilon_{\mathsf{s}} <_n 1$*. Then all of* NP *has* $(\varepsilon'_{\mathsf{c}}, \varepsilon'_{\mathsf{s}}, \varepsilon'_{\mathsf{zk}})$*-NIZKs with adaptive and statistical soundness where* $\varepsilon'_{\mathsf{c}}$*,* $\varepsilon'_{\mathsf{s}}$*, and* $\varepsilon'_{\mathsf{zk}}$ *are all negligible.*

Proof. First, suppose $\mathsf{NP} \subseteq \mathsf{BPP}$. In this case, every $\mathcal{L}$ in NP has a trivial NIZK (P sends nothing and V checks the instance on their own), meaning the claim holds trivially.

Otherwise, suppose $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$. By Corollary 3, we get that one-way functions exist; we can then use Theorem 9 to amplify the NIZK guaranteed for any $\mathcal{L} \in \mathsf{NP}$ to have negligible errors. □

6.2 Interactive Zero-Knowledge

In the interactive case, a classic result of [BJY97] shows that one-way functions alone suffice to construct a 4-round interactive zero-knowledge protocol with negligible errors. Similar to the non-interactive case, we can instantiate this result using the one-way function we get from Corollary 4 in order to (almost) unconditionally "amplify" (actually, just build from scratch) interactive zero-knowledge. Formally, we recall the following theorem from [BJY97].

Theorem 10 ([BJY97], Theorem 1.1)**.** *Suppose OWFs exist. Then for any* $\mathcal{L} \in \mathsf{NP}$*, there exists a four-round* $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$*-ZK argument for* $\mathcal{L}$ *with* ε_{c}*,* ε_{s}*, and* $\varepsilon_{\mathsf{zk}}$ *all negligible.*

Combining this with Corollary 4 immediately gives us the following result.

Corollary 6. *Suppose that* $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$ *or* $\mathsf{NP} \subseteq \mathsf{BPP}$*. Suppose further that all of* NP *has a constant-round, public-coin* $(\varepsilon_{\mathsf{c}}, \varepsilon_{\mathsf{s}}, \varepsilon_{\mathsf{zk}})$*-ZK argument with* $\varepsilon_{\mathsf{c}} + \varepsilon_{\mathsf{zk}} + \varepsilon_{\mathsf{s}} <_n 1$*. Then all of* NP *has a 4-round* $(\varepsilon'_{\mathsf{c}}, \varepsilon'_{\mathsf{s}}, \varepsilon'_{\mathsf{zk}})$*-ZK argument with* $\varepsilon'_{\mathsf{c}}$*,* $\varepsilon'_{\mathsf{s}}$*, and* $\varepsilon'_{\mathsf{zk}}$ *all negligible.*

Proof. First, suppose $\mathsf{NP} \subseteq \mathsf{BPP}$. In this case, every $\mathcal{L}$ in NP has a trivial zero-knowledge protocol (P sends nothing and V checks the instance on their own), meaning the claim holds trivially.

Otherwise, suppose $\mathsf{NP} \not\subseteq \mathsf{ioP/poly}$. By Corollary 4, we get that one-way functions exist; we can then use Theorem 10 to construct the desired zero-knowledge argument for NP. □

Acknowledgments. JH, DK, and KT were supported in part by NSF CAREER CNS-2238718, NSF CNS-2247727, a Google Research Scholar award and an award from Visa Research. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

[AK25] Benny Applebaum and Eliran Kachlon. NIZK amplification via leakage-resilient secure computation. Cryptology ePrint Archive, Paper 2025/995, 2025.

[BG24] Nir Bitansky and Nathan Geier. Amplification of non-interactive zero knowledge, revisited. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IX*, volume 14928 of *Lecture Notes in Computer Science*, pages 361–390. Springer, 2024.

[BJY97] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. Cryptology ePrint Archive, Paper 1997/002, 1997.

[BKP+24] Nir Bitansky, Chethan Kamath, Omer Paneth, Ron D. Rothblum, and Prashant Nalini Vasudevan. Batch proofs are statistically hiding. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 435–443. ACM, 2024.

[CHK25] Suvradip Chakraborty, James Hulett, and Dakshita Khurana. On weak nizks, one-way functions and amplification. In *Annual International Cryptology Conference*, pages 580–610. Springer, 2025.

[CLV26] Rohit Chatterjee, Yunqi Li, and Prashant Nalini Vasudevan. Weak zero-knowledge and one-way functions. Cryptology ePrint Archive, Paper 2026/299, 2026.

[GJS19] Vipul Goyal, Aayush Jain, and Amit Sahai. Simultaneous amplification: The case of non-interactive zero-knowledge. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 608–637. Springer, 2019.

[HN24] Shuichi Hirahara and Mikito Nanashima. One-way functions and zero knowledge. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1731–1738. ACM, 2024.

[IL90] R. Impagliazzo and L.A Levin. No better ways to generate hard np instances than picking uniformly at random. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 812–821 vol.2, 1990.

[Imp92] Russell Impagliazzo. *Pseudo-random generators for cryptography and for randomized algorithms*. PhD thesis, University of California, Berkeley, 1992.

[LMP24] Yanyi Liu, Noam Mazor, and Rafael Pass. A note on zero-knowledge for NP and one-way functions. Cryptology ePrint Archive, Paper 2024/800, 2024.

[Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*, pages 133–138. IEEE Computer Society, 1991.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way fuctions 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*, pages 3–17. IEEE Computer Society, 1993.

[Ps05] Rafael Pass and abhi shelat. Unconditional characterizations of non-interactive zero-knowledge. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*. Springer Berlin Heidelberg, 2005.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 80–91, 1982.

Supplementary material

Disclaimer

Case studies, comparisons, statistics, research and recommendations are provided "AS IS" and intended for informational purposes only and should not be relied upon for operational, marketing, legal, technical, tax, financial or other advice. Visa Inc. neither makes any warranty or representation as to the completeness or accuracy of the information within this document, nor assumes any liability or responsibility that may result from reliance on such information. The Information contained herein is not intended as investment or legal advice, and readers are encouraged to seek the advice of a competent professional where such advice is required.

These materials and best practice recommendations are provided for informational purposes only and should not be relied upon for marketing, legal, regulatory or other advice. Recommended marketing materials should be independently evaluated in light of

your specific business needs and any applicable laws and regulations. Visa is not responsible for your use of the marketing materials, best practice recommendations, or other information, including errors of any kind, contained in this document.

A Private to Public Coin in the Absence of ai-OWFs

Theorem 11. *If there exists an argument system that satisfies the following properties:*

- *It satisfies correctness up to ϵ_c error.*
- *It satisfies soundness up to ϵ_s error.*
- *There exists a simulator such that the output of the simulator is at most ϵ_{zk} distinguishable from the transcript of an honest execution of the protocol. Note that this is weaker than honest-verifier zero-knowledge since the simulator does not simulate the entire view of the verifier, only the transcript.*
- *For some constant $t \in \mathbb{N}$ the first t verifier messages are uniformly random strings.*

Then either io-uniform-ai-OWFs exist, or for every polynomial q, there exists an argument system that satisfies the following properties:

- *It satisfies correctness up to $\epsilon_c + 1/q(\lambda)$ error.*
- *It satisfies soundness up to $\epsilon_s + 1/q(\lambda)$ error.*
- *There exists a simulator such that the output of the simulator is at most $\epsilon_{zk} + 1/q(\lambda)$ distinguishable from the transcript of an honest execution of the protocol.*
- *The first $t + 1$ verifier messages are uniformly random strings.*

Additionally, the new prover, verifier, and simulator run in time polynomial in the original prover, verifier, and simulator runtimes.

Proof. Without loss of generality, we model the verifier V as sampling its first t messages $r_1, \ldots, r_t$ uniformly at random. Then, before sending the $(t+1)^{\text{th}}$ message, V samples internal randomness r. For $i \geq t+1$, the i^{th} verifier message is computed by a deterministic polynomial-time machine V_i that takes as input the transcript τ and the randomness r. Additionally suppose there exist polynomials p_1 and p_2 such that the size of the transcript just before the $(t+1)^{\text{th}}$ verifier message is $p_2(\lambda)$ and $|r| = p_1(\lambda)$.

Define $f(a, x) = V_{t+1}(a, x)$. If f is a $1/2q(\lambda)$-distributional (infinitely-often, uniform-secure) auxiliary-input one-way function we are done, since by [Imp92] it may be amplified to construct an (infinitely-often, uniform-secure) auxiliary-input one-way function. Suppose not, i.e., there exists a PPT machine A such that for all large enough λ, for any $a \in \{0,1\}^{p_1(\lambda)}$, and for $x \leftarrow \{0,1\}^{p_2(\lambda)}$,

$$\{V_{t+1}(a, x), A(a, V_{t+1}(a, x))\} \approx_{\frac{1}{2q(\lambda)}} \{V_{t+1}(a, x), x\}.$$

where the approximation symbol represents statistical indistinguishability. In particular,

$$\{V_{t+1}(a,x), x, A(a, V_{t+1}(a,x))\} \approx_{\frac{1}{q(\lambda)}} \{V_{t+1}(a,x), A(a, V_{t+1}(a,x)), x\}. \tag{2}$$

We proceed by a sequence of hybrids, where the final hybrid will represent a proof system where the first $t+1$ verifier messages are uniformly random.

Hybrid 0: Identical to the honest protocol.

Hybrid 1: The verifier behaves identically to the honest verifier except it appends $A(\tau, V_{t+1}(\tau, r))$ to its $(t+1)^{\text{th}}$ message, where τ is the transcript of the first t messages. This appended value is ignored by all subsequent verifier and honest prover steps.

The protocol preserves correctness since the appended value is ignored by both parties. It preserves soundness since the appended value can be sampled by the malicious prover themselves and is ignored by the verifier. The simulator for (honest-verifier) zero-knowledge can run the simulator from Hybrid 0 and then sample $A(V_{t+1}(\tau, r))$ and append to the $(t+1)^{\text{th}}$ verifier message in the simulated transcript. Since the honest transcript distribution in Hybrid 1 can be sampled by sampling an honest transcript in Hybrid 0 and running the same sample-and-append step, switching between Hybrid 0 and Hybrid 1 preserves simulator security.

Hybrid 2: Identical to Hybrid 1 except the $(t+1)^{\text{th}}$ message is computed as $V_{t+1}(\tau, r) \| r$, and all subsequent verifier steps use $A(\tau, V_{t+1}(\tau, r))$ in place of r. In other words, after computing $V_{t+1}(\tau, r)$ the roles of r and $A(\tau, V_{t+1}(\tau, r))$ are switched.

By (2), swapping $A(a, V_{t+1}(\tau, r))$ and r in the transcript after $V_{t+1}(\tau, r)$ can be detected with advantage at most $\frac{1}{q(\lambda)}$, even by unbounded adversaries. Therefore, completeness, soundness, and simulation security are all preserved upto $1/q(\lambda)$ error. For simulation security we crucially use the fact that the simulator only simulates the transcript instead of the full view of V, since (2) does not hold if given access to the randomness of A, which is part of the view of V.

Hybrid 3: Identical to Hybrid 2 except the $(t+1)^{\text{th}}$ message consists only of the randomness r. $A(a, V_{t+1}(\tau, r))$ is computed before the $(t+2)^{\text{th}}$ verifier message instead of before the $(t+1)^{\text{th}}$ message.

Since $V_{t+1}(\tau, r)$ can be computed from the transcript, this change does not affect completeness, soundness, or simulation security. The verifier now samples the first $t+1$ messages uniformly at random, so we are done. □

Corollary 7. *The existence of a constant-round $(\epsilon_c, \epsilon_s, \epsilon_{zk})$ (honest-verifier) zero-knowledge argument implies either io-uniform-ai-OWFs exist, or for every polynomial q, there exists a public-coin constant-round $(\epsilon_c + 1/q(\lambda), \epsilon_s + 1/q(\lambda), \epsilon_{zk} + 1/q(\lambda))$ (honest-verifier) zero-knowledge argument.*

Proof. Follows from repeated application of Theorem 11 along with the following observations:

- While there is polynomial blowup in runtime each time the theorem is applied, since there are constant rounds the final runtimes are polynomial.
- While errors grow by an inverse polynomial additive term, we can set the to be small enough to obtain our desired final bounds.
- While the theorem only guarantees a weakening of honest zero-knowledge, for public-coin protocols, both notions are the same. Therefore we recover honest-verifier zero-knowledge for the final protocol.

□

B Proof of Lemma 2

Here we provide an explicit proof of Lemma 2, noting that it is closely mirrors the proof of Lemma 3.1 in [LMP24].

Proof of Lemma 2. Let $(R^{(\cdot)}, \{f_x\}_{x\in\{0,1\}^*}, p(\cdot))$ be the decision-to-inversion reduction for $\mathcal{L}$ guaranteed by the statement of the lemma. We define the function $g(r_1\|r_2) = \mathcal{D}(r_1)\|f_{\mathcal{D}(r_1)}(r_2)$, and claim that g is a weak one-way function.

Indeed, suppose for the sake of contradiction that g is not a weak one-way function. This means that there exists an algorithm $\mathcal{A}$ that inverts g with probability at least 0.75 on infinitely many input sizes. Using $\mathcal{A}$, we define an algorithm $\mathcal{B}$ to decide $(\mathcal{L}, \mathcal{D})$ as follows. On input x, $\mathcal{B}$ first samples a random r_2 and runs $\mathcal{A}(x, f_x(r_2))$. If $\mathcal{A}$ fails to find a preimage of $f_x(r_2)$, $\mathcal{B}$ immediately halts and outputs $\perp$. Otherwise, $\mathcal{B}$ outputs $R^{\mathcal{A}}(x)$.

We now claim that $\mathcal{B}$ puts $(\mathcal{L}, \mathcal{D}) \in \mathsf{ioAvgBPP/poly}$, which is a contradiction. Fix any n where $\mathcal{A}$ succeeds in inverting g and that is sufficiently large for our guarantees on R to hold. To show that the first point of Definition 9 holds, we fix an $x \in \mathrm{Supp}(\mathcal{D}_n)$ and consider two possible cases. If $\mathcal{A}(x, \cdot)$ inverts f_x with probability at most $\frac{1}{p(n)}$, we have that the first step of $\mathcal{B}(x)$ will cause it to output $\perp$ with probability at least $1 - \frac{1}{p(n)}$. If instead $\mathcal{A}(x, \cdot)$ inverts f_x with probability at least $\frac{1}{p(n)}$, our guarantees on R tell us that (as long as the first step doesn't output $\perp$), $\mathcal{B}(x)$ will output $\mathcal{L}(x)$ with probability at least $1 - \frac{1}{p(n)}$. Thus in either case, we have that $\mathcal{B}(x) \in \{\perp, \mathcal{L}(x)\}$ with probability at least $1 - \frac{1}{p(n)}$, which for sufficiently large n is at least 0.9.

For the second point of Definition 9, we note that $\mathcal{B}(x)$ can only output $\perp$ if $\mathcal{A}(x, \cdot)$ fails to invert f_x. Thus the probability of this happening over a random $x \xleftarrow{\$} \mathcal{D}_n$ is exactly the probability that $\mathcal{A}$ fails to invert a random output of g, which we know to be at most 0.25.

Putting this all together, we have reached a contradiction, and so must have that g is in fact a weak one-way function. By Theorem 4, this means that one-way functions exist. □