Electronic Colloquium on Computational Complexity, Report No. 95 (2025)

Gödel in Cryptography: Effectively Zero-Knowledge Proofs for NP with No Interaction, No Setup, and Perfect Soundness

Rahul Ilango
MIT
rilango@mit.edu

Abstract

A zero-knowledge proof demonstrates that a fact (like that a Sudoku puzzle has a solution) is true while, counterintuitively, revealing nothing else (like what the solution actually is). This remarkable guarantee is extremely useful in cryptographic applications, but it comes at a cost. A classical impossibility result by Goldreich and Oren [J. Cryptol. '94] shows that zero-knowledge proofs must necessarily sacrifice basic properties of traditional mathematical proofs — namely perfect soundness (that no proof of a false statement exists) and non-interactivity (that a proof can be transmitted in a single message).

Contrary to this impossibility, we show that zero-knowledge with perfect soundness and no interaction is effectively possible. We do so by defining and constructing a powerful new relaxation of zero-knowledge. Intuitively, while the classical zero-knowledge definition requires that an object called a simulator actually exists, our new definition only requires that one cannot rule out that a simulator exists (in a particular logical sense). Using this, we show that *every falsifiable security property of (classical) zero-knowledge can be achieved with no interaction, no setup, and perfect soundness.* This enables us to remove interaction and setup from (classical) zero-knowledge in essentially all of its applications in the literature, at the relatively mild cost that such applications now have security that is "game-based" instead of "simulation-based."

Our construction builds on the work of Kuykendall and Zhandry [TCC '20] and relies on two central, longstanding, and well-studied assumptions that we show are also necessary. The first is the existence of non-interactive witness indistinguishable proofs, which follows from standard assumptions in cryptography. The second is Krajíček and Pudlák's 1989 conjecture that *no optimal proof system exists.* This is one of the main conjectures in the field of proof complexity and is the natural finitistic analogue of the impossibility of Hilbert's second problem (and, hence, also Gödel's incompleteness theorem). Our high-level idea is to use these assumptions to construct a prover and verifier where no simulator exists, but the non-existence of a simulator is independent (in the logical sense of unprovability) of an arbitrarily strong logical system. One such logical system is the standard axioms of mathematics: ZFC.

ISSN 1433-8092

Contents

1 Introduction

Imagine Alice wants to convince Bob that the Sudoku puzzle he is working on actually has a solution. Alice could simply reveal a solution to Bob. But this would ruin the puzzle for Bob. Ideally, Alice could prove to Bob that the puzzle is solvable without revealing any additional information (like how to solve it). This is exactly what *zero-knowledge proofs* [GMR89] guarantee.

In a zero-knowledge proof, a prover (like Alice) convinces a verifier (like Bob) that a statement is true, while — counterintuitively — revealing nothing. The crux of Goldwasser, Micali, and Rackoff's definition is a mathematical formalization of "revealing nothing." Their beautiful idea is to do so via the notion of a *simulator*.

A simulator is an efficient algorithm that the verifier can use to simulate — on its own — any interaction the verifier could possibly have with the prover. This guarantees that the verifier learns "nothing new" by interacting with the prover. Intuitively, if you already know everything someone else is going to say, then you cannot learn anything by talking to them.

Remarkably, a proof of any statement can be transformed into a zero-knowledge proof, under a standard assumption in cryptography (one-way functions exist) [GMW86]. This is extremely useful, and zero-knowledge proofs have become a basic tool in cryptography, important in theoretical and practical cryptographic systems alike. However, using zero-knowledge proofs comes at a cost.

Drawbacks of zero-knowledge. All existing zero-knowledge proofs diverge from the traditional notion of a proof (such as Alice simply revealing a Sudoku solution to Bob) in two undesirable ways.

- **Interaction**: The prover and verifier need to interact — either explicitly, via sending messages back and forth, or implicitly, via trusted setup.[1] (What is often called "non-interactive" zero-knowledge (NIZK) [BFM88; BSMP91] requires trusted setup.) In contrast, a traditional proof is a single string that the prover can send in a single message.
- **Imperfect Soundness:** The verifier sometimes accepts "proofs" of false statements. In contrast, traditional proofs offer *perfect soundness*: if there is a proof of a statement, it must be true.

One might hope that these drawbacks are avoidable. Unfortunately, Goldreich and Oren [GO94] show that neither interaction nor imperfect soundness can be removed from zero-knowledge proofs.[2] Moreover, this impossibility seems conceptually inherent and not easily avoidable, as Barak, Ong, and Vadhan [BOV07] remark.

> "Given the aforementioned impossibility results [GO94], reducing the interaction further [than trusted setup] seems unlikely. Indeed, a truly noninteractive proof system, in which the prover sends a single proof string to the verifier, seems to be inherently incompatible with the intuitive notion of 'zero-knowledge': from such a proof, the verifier gains the ability to prove the same statement to others."

In fact, there are difficulties even if one significantly relaxes the guarantees of zero-knowledge. For example, it was open (until this paper) to construct proofs — without interaction or trusted setup — that are even just *witness hiding*, meaning that a proof of an NP-statement does not reveal a witness to it [FS90; KZ20].

This work. This paper's main contribution is to define and construct a powerful new relaxation of zero-knowledge that bypasses these barriers. To give a taste, our results imply that *every falsifiable*[3] *security*

[1] An example of trusted setup is that both parties share a common reference/random string (crs) that is trusted to have been sampled from a distribution [BFM88; BSMP91]. As noted in previous work [BOV07], this is itself a limited form of interaction.

[2] It is easy to see that perfect soundness actually suffices to remove interaction. If the verifier is perfectly sound, then it is also sound when its random coins are set to be all zero. But then one can collapse the interaction to a single prover message.

[3] As we describe later, falsifiability [Nao03] roughly means that one can check in polynomial (or, in our case, even exponential) time if a given adversary breaks the security property. This is true for many common security definitions, including one-way functions, semantically-secure encryption, and indistinguishability obfuscation.

property of (classical) zero-knowledge is achievable with no interaction, no setup, and perfect soundness. This enables us to remove setup and interaction from zero-knowledge in essentially all of its applications in the literature, at the mild cost that the end applications now have security that is "game-based" (instead of perhaps "simulation-based"). To explain our new definition, we first need the notion of a *logical system* [CR79].

Background: logical systems. For us, a logical system is specified by a deterministic Turing machine $\mathcal{L}$. You can think of $\mathcal{L}$ as taking as input a string called a statement (for now, do not worry about what a statement actually is) and a string called a proof, and accepting or rejecting depending on whether the proof is "valid." There are just two requirements on a Turing machine in order for it to be a logical system:

- *Efficiency:* $\mathcal{L}$ must run in polynomial time.
- *Soundness:* whenever a $\mathcal{L}$ accepts a proof of a statement, that statement must actually be true.[4]

We stress that this definition of a logical system is extremely general. Almost any notion of a mathematical proof fits into this framework, including ZFC (Zermelo-Fraenkel with Choice, the standard axioms in mathematics), or even ZFC with extra cryptographic axioms added, such as "language L requires circuits of size at least $n^{\log n}$ for all $n \geq 256$," assuming they meet our soundness criterion. See Section 3.1 for more details.

Finally, before we proceed, we modify our terminology. What we have been calling a *logical system* so far is actually called a *proof system* in the field of proof complexity, as defined by Cook and Reckhow [CR79]. Henceforth, we adopt the proof complexity meaning. To avoid confusion, we will use different terms (usually *prover* and *verifier*) for cryptographic proof systems.

Our new definition. We can now discuss our new relaxation of zero-knowledge. Recall that being (classically) zero-knowledge means that (a quantitative form of) the statement $X =$ "a simulator exists for the prover" is true. We can also think of X as a sequence of statements $X_\lambda =$ "a simulator exists for the prover on security parameter λ."[5]

Our new definition — *effectively zero-knowledge to* $\mathcal{L}$ — replaces the requirement that X_λ is *actually true* with a weaker *relaxation* of truth. In particular, if X_λ is actually true, then we know $\mathcal{L}$ cannot prove $\neg X_\lambda$ (because $\mathcal{L}$ is sound). We relax this property. Effectively zero-knowledge to $\mathcal{L}$ roughly says that any $\mathcal{L}$-proof of $\neg X_\lambda$ has length $\lambda^{\omega(1)}$. Loosely speaking, this means $\mathcal{L}$ cannot efficiently rule out that the prover has a simulator.

This turns out to be an extremely powerful guarantee. As we will see, there is some formal sense in which it implies the prover is guaranteed to have every consequence of being zero-knowledge that (a) is "polynomial-time observable" and (b) has polynomial-length $\mathcal{L}$-proofs.

But isn't that impossible? At first glance, the unprovability guarantee above may seem too strong to be true. For example, shouldn't one be able to use Goldreich and Oren's impossibility result [GO94] to prove $\neg X =$ "the prover P has no simulator" in any sufficiently powerful $\mathcal{L}$?

But [GO94] only yields the disjunction "either the verifier V is not perfectly sound or the prover P has no simulator." This is okay for us, as it will turn out that our construction is "dual-mode" in the sense that $\mathcal{L}$ cannot determine whether (a) V is perfectly sound and P has no simulator or (b) P has a simulator and V is not sound.

[4] There are some technicalities here related to (a) what constitutes a statement and (b) what it means for an arbitrary statement to be true. Our actual requirement is soundness on just a small class of statements related to computation. See Section 3.1 for a formal definition.

[5] For readers not familiar with cryptography, the security parameter λ roughly plays the role that the input length does in complexity theory.

Our results. We now state our main result, which holds under three central, longstanding, and well-studied assumptions from complexity theory, proof complexity, and cryptography. We will give a detailed discussion of these assumptions and the meaning of "falsifiable" in a few paragraphs. Below, *zero-interaction*[6] *provers and verifiers* refer to uniform polynomial-time algorithms that use no interaction (the prover sends one message to the verifier) and involve no setup.

Theorem 1.1 (Zero-Interaction Effectively Zero-Knowledge Proofs (ZIZK))**.** *Assume* $\mathsf{P} = \mathsf{BPP}$*, no infinitely often optimal proof system exists, and non-interactive witness indistinguishable proofs*[7] *(NIWIs) for* SAT *exist. Then both of the following hold (in fact, the second implies the first):*

1. *For every falsifiable security property* Π *of (classical) zero-knowledge, there is a zero-interaction prover and verifier for* SAT *with perfect soundness and with property* Π*.*
2. *For every proof system* $\mathcal{L}$*, there is a zero-interaction prover and verifier for* SAT *that is effectively zero-knowledge to* $\mathcal{L}$*.*

Moreover, subexponentially-secure[8] *versions of (1) and (2) hold assuming subexponential versions of the second and third assumptions hold (see Theorem 2.1 and Theorem 3.1 for a formal statement).*

We view Theorem 1.1 and especially (2) as showing that zero-knowledge with zero-interaction and perfect soundness is effectively possible. For example, one way to use (2) is as follows. Suppose you have an application — such as maliciously secure multi-party computation — where one achieves a (possibly simulation-based) security property $\Pi^\star$ via interactive zero-knowledge proofs. By using (2), one can remove all interaction from the proofs, while still preserving every falsifiable security property Π that is logically implied by $\Pi^\star$ (in the logical system of one's choice). We interpret this as saying that we can remove interaction from zero-knowledge, at the mild cost that applications now have game-based security. Indeed, Theorem 1.1 suggests that — with a few exceptions[9] — interaction is only necessary in zero-knowledge if one wants simulation-based guarantees.

We also offer a few extensions of our result.

- *A Concrete Candidate:* We give a concrete zero-interaction prover and verifier for SAT that we conjecture is effectively zero-knowledge to ZFC (Conjecture 3.14) and that we further informally conjecture enjoys every "natural" falsifiable security property of zero-knowledge (Conjecture 2.12).
- *The First Construction of Non-Interactive Witness Hiding Proofs.* An immediate corollary of our results is the first construction of non-interactive witness hiding proofs with a uniform prover and verifier (Corollary 2.8). Previously, the best construction required a *non-uniform* prover and verifier [KZ20].
- *A Generic Transformation from* $\mathsf{Search\text{-}NP}$ *to* TFNP *via ZIZKs.* TFNP [MP91] is the class of NP search problems where a solution is guaranteed to exist. A natural way [HNY17] to try to convert a hard $\mathsf{Search\text{-}NP}$ problem to a hard TFNP problem is to include a proof π that a solution exists. But this transformation may not preserve hardness, as knowing π could make the problem easier.

  An immediate corollary of Theorem 1.1 (in the subexponential regime) is that we can use the above approach to generically transform any $\mathsf{Search\text{-}NP}$ problem to a corresponding TFNP problem, while preserving worst-case circuit complexity almost exactly (Corollary 7.28). In other words, under the subexponential assumptions in Theorem 1.1, *every* $\mathsf{Search\text{-}NP}$ problem (including $\mathsf{Search\text{-}SAT}$) is equally hard, even given a proof that there is a solution.

[6] We use the term *zero-interaction* because *non-interactive* zero-knowledge [BFM88; BSMP91] is used for the case where the prover and verifier have access to trusted setup, which is a limited form of interaction. As an analogy, non-alcoholic drinks can have a small amount of alcohol, as opposed to zero alcohol. We thank Neekon Vafa for suggesting this name.

[7] Our definition of a NIWI differs mildly from the usual definition in that we require its indistinguishability ϵ to be at most some *efficiently computable* negligible function. This is a very weak requirement. See Definition 4.3 for the details.

[8] In this paper, subexponential refers to $2^{\lambda^{\Omega(1)}}$.

[9] One trivial exception is the aforementioned attack [BOV07] that "the verifier gains the ability to prove the same statement to others." More generally, our results do not preserve aspects of zero-knowledge related to *deniability*. We do not view this as a significant deficiency because non-interactive zero-knowledge with trusted setup (NIZKs) have the same deficiency [Pas03a].

- *A Generic Transformation from* UP *to* $\mathsf{NP} \cap \mathsf{coNP}$ *via ZIZKs.* We also show an analogous statement for UP and $\mathsf{NP} \cap \mathsf{coNP}$ languages, building on [GIK+23], who gave a similar result in the random oracle model. Specifically, under the subexponential versions of the assumptions in Theorem 1.1, we show that for every UP language there is a corresponding $\mathsf{NP} \cap \mathsf{coNP}$ *language* with matching worst-case circuit complexity.

- *A Strengthening in the Non-Uniform Setting:* We construct a *non-uniform* zero-interaction prover and verifier that (under plausible assumptions) is effectively zero-knowledge to *every* proof system and hence has every falsifiable property of zero-knowledge (Theorem 7.6).

  Actually, this particular "extension" predates our main result and fell out as an application of a result in a different work [Ila25] by the author. After realizing we could prove this paper's main theorem (Theorem 1.1), we decided to split into two papers: this one (focused on zero-knowledge) and [Ila25] (focused on the meta-mathematics of lower bounds in complexity theory). We did keep a variant of the non-uniform construction (Theorem 7.6) as an application in [Ila25] because the proof is short and it naturally falls out of it.[10] But [Ila25] explicitly defers to this paper on both how to use this object and how to make it significantly more applicable (by making it uniform).

We now discuss the meaning of falsifiability and also our assumptions.

What does falsifiable mean? A security property is falsifiable [Nao03] if one can (somewhat) efficiently test if a given adversary breaks the security property. To illustrate, we give a few examples. We advise the reader to skip an example if they are not familiar with the underlying cryptographic object.

- *Being a One-Way Function*: The test checks if the adversary inverts the function on random inputs.

- *Being an Encryption Scheme*: The adversary's description includes two messages m_1 and m_2. The test checks if the adversary distinguishes between random encryptions of m_1 and m_2.

- *Being a Witness Indistinguishable Prover [FS90]*: The adversary's description includes a formula φ and two satisfying assignments w and w'. The test checks if the adversary distinguishes between the prover's proofs of "φ is satisfiable" generated using w from those generated using w'.

- *Being an Indistinguishability Obfuscator [BGI+12]*: The adversary's description includes circuits C_0 and C_1 of the same size. The test first uses brute force to verify that C_0 and C_1 compute the same function, and, if so, then tests if the adversary distinguishes between obfuscations of C_0 and C_1.

  In contrast to the previous examples, here the test runs in exponential time (to do the brute force check), rather than polynomial time.

Along these lines, we give a definition of a falsifiable security property of zero-knowledge (Definition 2.6). Conclusion (1) in Theorem 1.1 applies if the test runs in polynomial time. The subexponential version of (1) even handles tests that run in $2^{\mathsf{poly}(n)}$ time (by setting the security parameter $\lambda = \mathsf{poly}(n)$ to be a sufficiently large polynomial). The latter is very general and captures, to the best of our knowledge, all indistinguishability-based security properties in the literature.

Plausibility of our assumptions. All of the assumptions in Theorem 2.1 are longstanding, well-studied, and (in our opinion) very plausible. Moreover, these assumptions are "win-win." This means that breaking one constitutes an unexpected and fundamental discovery in proof complexity, cryptography, or complexity theory. We now discuss each assumption in detail.

- *No Optimal Proof System:* This is one of the main conjectures in the field of proof complexity. For example, it is one of the three[11] major problems mentioned on proof complexity's Wikipedia page [Wik24].

[10] For completeness, we also give a proof of Theorem 7.6 in this paper.

[11] The other two are whether $\mathsf{NP} = \mathsf{coNP}$ and automatability (whether one can efficiently generate short proofs whenever they exist).

It is also well-studied [KP89; BG98; MT98; KM98; Meß99; BFFM00; Sad02; KMT03; GSSZ04; Pud06; Bey07; Sad07; Sad08; BKM09; BS09; Bey10; CF10; BKM11; BS11; HIMS12; Pud13; CFM14; Kra14; PS19; DG20; Kha22; Kha24].

The notion of an optimal proof system was first defined by Krajíček and Pudlák [KP89], who also conjectured their non-existence. Very roughly speaking, a proof system is *optimal* if it has shorter proofs (up to polynomial blowup) of every coNP statement, when compared to any other proof system.[12] Somewhat amusingly, the definition of an optimal proof system and the definition of a zero-knowledge proof were published in journals in the same year [KP89; GMR89].

We encourage the reader to think of the non-existence of optimal proof systems as a stronger version of $\mathsf{NP} \neq \mathsf{coNP}$. Recall $\mathsf{NP} \neq \mathsf{coNP}$ (roughly) says that, for every proof system $\mathcal{L}$, there exist unsatisfiable formulas ψ_n that lack $\mathsf{poly}(n)$-length $\mathcal{L}$-proofs of unsatisfiability. In contrast, the non-existence of an optimal proof system turns out to be equivalent [KP89] to saying something stronger: that there is a P-*uniform*[13] sequence of such ψ_n. In fact, this uniform sequence is what we use in our construction.

Intriguingly, optimal proof systems are closely related to old questions in mathematical logic. One of Krajíček and Pudlák's main results [KP89] is that if an optimal proof system exists, then there is a single proof system that can prove the "finitary consistency" of all other proof systems. As Krajíček and Pudlák remark, this means that if an optimal proof system exists, then "we could realize the Hilbert program in a modified, finitistic sense. We conjecture that this is not possible."

In other words, it is natural to view the existence of an optimal proof system as a scaled-down version of Hilbert's second problem, which Gödel's incompleteness theorem [Göd31] famously resolved negatively. Indeed, the non-existence of optimal (respectively, subexponentially optimal) proof systems is immediately implied by Pudlák's (respectively, Mycielski's) conjectured finitary analogue of Gödel's incompleteness theorem [Pud86].

Optimal proof systems are also related to complexity theory questions. Krajíček and Pudlák [KP89] show that if no optimal proof system exists, then $\mathsf{NE} \neq \mathsf{coNE}$. Köbler, Messner, and Torán [KMT03] show that no optimal proof system exists if $\mathsf{NP} \cap \mathsf{SPARSE}$ lacks a complete problem.[14]

As is usual when using hardness assumptions in cryptography, we need to rule out even an infinitely often upper bound. For example, "SAT is not in P infinitely often" is necessary for one-way functions to exist. Similarly, we need to assume that no infinitely often optimal proof system exists. This assumption appears just as plausible as the non-infinitely-often version.

- *Non-interactive Witness Indistinguishable Proofs (NIWIs)*: NIWIs (henceforth, we just say NIWIs instead of NIWIs for SAT) are themselves a powerful relaxation of zero-knowledge achievable with perfect soundness and zero interaction [BOV07]. In more detail, a NIWI consists of a zero-interaction prover and verifier with perfect soundness and the following witness indistinguishable (WI) guarantee: for all $\varphi(w) = \varphi(w') = 1$, the following two distributions are computationally indistinguishable
  - the proof π of "φ is satisfiable" generated by the prover when given witness w, and
  - the proof π of "φ is satisfiable" generated by the prover when given witness w'.[15]

  At first glance (similar to indistinguishability obfuscation [BGI+12]), it is not clear how useful WI is. For example, if φ only has one witness, then WI provides no guarantee whatsoever. Nevertheless, WI turns out to be quite powerful [FS90; FLS90], as we will also see in this paper.

  There are several different constructions of NIWIs from various widely-believed cryptographic assumptions. For example, NIWIs exist
  - if indistinguishability obfuscation and one-way permutations exist [BP15],

[12] See Definition 3.8 and Conjecture 3.10 for formal definitions.
[13] This means that one can produce ψ_n in uniform polynomial-time given 1^n.
[14] SPARSE is the set of languages L with $\mathsf{poly}(n)$ many n-bit YES instances.
[15] See Definition 4.3 for a formal definition of NIWIs.

- if a trapdoor one-way permutation exists and E[16] requires $2^{\Omega(n)}$-size non-deterministic circuits [BOV07],
- if indistinguishability obfuscation and one-way functions exist and E requires $2^{\Omega(n)}$-size non-deterministic circuits [BOV07; BP15], or
- assuming certain hardness [BF03] for bilinear groups [GOS12].

- $\mathsf{P} = \mathsf{BPP}$: This is one of the main conjectures in complexity theory, and it follows from widely-believed assumptions. For example, a celebrated result by Impagliazzo and Wigderson [IW97] shows that $\mathsf{P} = \mathsf{BPP}$ if E requires $2^{\Omega(n)}$-size circuits.

  As is often done (e.g., Goldreich [Gol11]), we write $\mathsf{P} = \mathsf{BPP}$ to mean that their promise classes coincide, i.e., $\mathsf{Promise\text{-}P} = \mathsf{Promise\text{-}BPP}$. This abuse of notation is done because all known approaches[17] to showing that $\mathsf{P} = \mathsf{BPP}$ (for languages) also imply that $\mathsf{Promise\text{-}P} = \mathsf{Promise\text{-}BPP}$.

Necessity of our assumptions. Since witness indistinguishability is a falsifiable property of zero-knowledge, the existence of NIWIs is necessary for conclusion (1) of Theorem 1.1. Somewhat surprisingly, we show that the non-existence of infinitely often optimal proof systems is also necessary for conclusion (1) of Theorem 1.1, assuming injective one-way functions exist with, say, quasipolynomial security (see Theorem 7.22).[18]

Avoiding barriers. We now discuss how our results bypass the aforementioned barriers. Our prover does not actually have a simulator, so Goldreich and Oren's impossibility result [GO94] does not apply. Relatedly, the existence of a simulator is not falsifiable.

Next, avoiding the "inherent incompatibility" mentioned in [BOV07] is more subtle. Recall that the incompatibility is that if a prover is truly non-interactive, then "the verifier gains the ability to prove the same statement to others." Indeed, this is true in our construction. However, somewhat remarkably, our definition seems to cleanly separate this "obviously broken" security property from other security properties of zero-knowledge. In particular, it turns out that the "obviously broken" security property is unprovable in $\mathcal{L}$, essentially for the same reason that the non-existence of a simulator is unprovable in $\mathcal{L}$.

The high-level construction. We now give an overview of our construction. As we describe in detail in Section 3.5, our construction builds on a long sequence of ideas in the cryptographic literature, including the notion of witness indistinguishability by Feige and Shamir [FS90], the "OR proof" construction of Feige, Lapidot, and Shamir [FLS90], the realizability of ZAPs and NIWIs by Dwork and Naor [DN07] and Barak, Ong, and Vadhan [BOV07] respectively, and especially Kuykendall and Zhandry's approach [KZ20] to constructing non-interactive witness hiding proofs. The construction works as follows.

1. *Fix a proof system and get a hard sequence of unsatisfiable formulas.* Fix an arbitrary proof system. To be concrete, we fix ZFC in this overview. The proof complexity assumption we use will guarantee the following. There is a P-uniform sequence[19] of unsatisfiable formulas ψ_λ such that, for each λ, the length of any ZFC-proof that "ψ_λ is unsatisfiable" is $\lambda^{\omega(1)}$. (In fact, these ψ_λ correspond to the consistency of another proof system, which hints at the relationship with Gödel's incompleteness theorem.)

2. *Construct the prover.* We construct the prover $\mathcal{P}$ as follows. Given a SAT instance φ and a witness w, output a NIWI proof that "either φ is satisfiable or ψ_λ is satisfiable."

3. *Construct the verifier.* Given a formula φ and a purported proof π, the verifier accepts if π is a NIWI proof of "either φ is satisfiable or ψ_λ is satisfiable" for some λ.

4. *Perfect soundness.* A NIWI proof enjoys perfect soundness (meaning there are no proofs of false statements). Thus, since the ψ_λ are all unsatisfiable (by construction), a NIWI proof that "either φ is satisfiable or ψ_λ is satisfiable" implies that φ is satisfiable. Thus, we get perfect soundness.

[16] E refers to the complexity class $\mathsf{DTIME}[2^{O(n)}]$.
[17] See Chen and Tell [CT23] for a discussion of this.
[18] If one only considers "natural" falsifiable properties, then this may be avoidable. See Section 2.5.
[19] This means one can produce ψ_λ in uniform polynomial time given 1^λ.

5. *Effectively zero-knowledge.* Pretend for a second that ψ_λ is satisfiable at x_λ (of course, ψ_λ is unsatisfiable so this is not the case, but pretend). Then it turns out there is a simple circuit that simulates the prover. On input φ, the simulator outputs a NIWI proof π that "either φ is satisfiable or ψ_λ is satisfiable," where it uses the witness x_λ with $\psi_\lambda(x_\lambda) = 1$ to generate the proof π.[20]

   To show that this is indeed a simulator, we need to prove that the output of this simulator is indistinguishable from the output of the prover. This will follow from the witness indistinguishable (WI) guarantee of a NIWI. In more detail, the WI guarantee says that given a proof π of the statement "either φ is satisfiable or ψ_λ is satisfiable," one cannot tell whether π was generated using a witness for φ or a witness for ψ_λ. Hence, the output of this simulator is indistinguishable from the output of the prover (where one outputs a NIWI proof using a witness for φ).

   To summarize, if ψ_λ were satisfiable, then there is a simulator for the prover on security parameter λ. In the contrapositive, if there is no simulator for the prover on security parameter λ, then ψ_λ is unsatisfiable. But we choose the ψ_λ to lack short ZFC-proofs of unsatisfiability. Hence, ZFC does not have short proofs refuting the existence of a simulator for the prover on security parameter λ.[21]

   This also guarantees that certain "polynomial-time observable" consequences of having a simulator are true. In more detail, suppose ZFC has a short proof of

   $$\text{"the prover having a simulator on } \lambda \text{ implies } U(M) \text{ outputs 1 in time at most } t\text{,"} \tag{1}$$

   where U is a fixed efficient universal Turing machine. Our goal is to show that indeed $U(M) = 1$. For contradiction, suppose not. Then one gets a short proof that "the prover lacks a simulator on λ": just run $U(M)$ for t steps, see that it doesn't output one, and use the contrapositive of (1). But this contradicts the fact that there is no short proof of the non-existence of a simulator.

Interpretations. We believe our ideas could help achieve other "impossible" objects in cryptography and complexity theory. We state a few different interpretations that may be useful for future definitions.

- *Irrefutability of a Simulator and Other Properties:* Instead of requiring that a simulator actually exists, one can require that it is hard to prove that no simulator exists. One could also apply this to other properties beyond simulators. For example, if an object has dual-modes A and B, it might be hard to prove that it is not in mode B, even though the object is in mode A.

- *Proof-Theoretic Relaxations:* Often in cryptography, one would like an information-theoretic security definition D to hold, but it is impossible to achieve. To cope with this, one considers a computational relaxation $\tilde{D}$ of the definition that is achievable.

  In our setting, we relax a definition in a different way. If a definition D holds, then every security property ρ that is implied by D holds. To relax this, instead of hoping that every ρ implied by D holds, we consider a proof-theoretically interesting subset of properties and only require that these hold.

  In fact, this particular perspective already features prominently in a line of work [JJ22; JKLV24; JKLM25; MDS25] initiated by Jain and Jin [JJ22], who were the first to use proof complexity in cryptography, to our knowledge. Loosely speaking, [JJ22] shows that one can get better guarantees from indistinguishability obfuscation if one has a short proof that two circuits are equivalent in a particular proof system (extended Frege). Follow-up work builds on this framework to construct succinct non-interactive arguments (SNARGs) [JKLV24; JKLM25] and practical obfuscation candidates [MDS25].

  One interesting direction to explore is whether some proof-theoretic relaxation of the random oracle heuristic [BR93] might hold.

[20] This simulator is non-uniform, as it needs to know x_λ. This suffices for most applications (since security is usually against non-uniform adversaries). We discuss how to handle cases where one needs a uniform simulator in Section 2.4.

[21] Actually, this step requires ZFC to be able to prove the security of the NIWI. To avoid this, we actually choose ψ_λ to lack short proofs of unsatisfiability in the proof system (ZFC + an axiom for the NIWI's security).

- *Axioms as a Resource:* The impossibility of truly non-interactive zero-knowledge is (roughly speaking) because of tension between soundness and the simulator. What we have done, in some sense, is create a setup where the security definition does not have access to the axiom needed to discern soundness. As a result, there is no tension between security and soundness because, in the eyes of security, soundness might not even hold. In some ways, this is analogous to a technique in cryptography called "complexity leveraging" but on axioms.

  This suggests that, in settings where one wants to avoid an "uninteresting" counterattack on a security definition, one might be able to proof-theoretically concoct a security notion where it is impossible to even talk about the "uninteresting" counterattack because of a lack of axioms. Perhaps then the new definition is achievable.

Related work. We now discuss related work. There are several prior works [BP04; BOV07; BL18; KZ20] studying relaxations of zero-knowledge achievable with zero-interaction. One line of work [BP04; BL18], which we have not yet discussed, achieves (weak [Pas03b]) zero-knowledge with zero-interaction by relaxing statistical soundness to *computational* soundness (for comparison, we consider perfect soundness). Specifically, Barak and Pass [BP04] give a construction with uniform soundness (i.e., no uniform algorithm can prove a false statement). Bitansky and Lin [BL18] extend this to a weak form of soundness against non-uniform adversaries (the number of false statements an adversary can generate is bounded by its amount of non-uniformity).

There is a long line of work beginning with [Pap94], and especially inspired by [BPR15], that shows TFNP hardness using cryptographic objects, including witness indistinguishable proofs [HNY17]. We point the reader to [HKKS20] for a good overview of this line of work. In contrast to previous work, our TFNP result applies to any worst-case hard Search-NP problem.

As we mentioned earlier, we are not the first to use proof complexity in cryptography [JJ22; JKLV24; JKLM25; MDS25]. However, to our knowledge, our work is the first that uses *hardness* assumptions from proof complexity and also the first that considers very strong proof systems, like ZFC.

The next two sections. At this point, we would like to give a more detailed discussion of our results and how to use them. However, one difficulty is that our main theorem requires some proof complexity background and a few minor (but subtle) modifications of standard cryptographic definitions. As a result, we decided that the best approach is a longer exposition that both formally states our definitions and provides several examples. In contrast, the main technical content of this paper is actually quite short (a few pages).

We split the exposition into two parts. In the first part (Section 2), we warm up with a weaker version of our results, which suffices for most applications and requires no proof complexity background. In the second part (Section 3), we review the necessary proof complexity background and formally state our main result. We discuss some open questions in Section 3.6.

2 An Exposition of Our Results: Part 1 (No Proof Complexity)

In this section, we will formally state a weaker version of our main result and give a few examples of how to use it. As a preview, we state the theorem now. We define the terminology it uses in the next subsection.

Theorem 2.1 (Weak Version of Main Result)**.** *Assume*

- $\mathsf{P} = \mathsf{BPP}$*,*
- *NIWIs exist (respectively, subexponentially secure NIWIs exist), and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

For every falsifiable property of zero-knowledge Π*, there exists a perfectly sound prover for which* Π *holds (respectively, holds with subexponential security).*

2.1 Definitions: Provers, Simulators, and Falsifiable Properties of Zero-Knowledge

We will now spend about a page spelling out several definitions. Most of these definitions are standard in cryptography with minor variations (mostly, we need non-asymptotic "pointwise" versions). We encourage the reader to go through all of them, but a seasoned cryptographer can likely focus on the parts we have highlighted in fuchsia and skim the rest.

We start with our notion of a prover. For brevity, a prover will refer to both a prover and its verifier.[22] Since all provers in this paper will use zero interaction, we also omit writing zero-interaction.

Definition 2.2 (Provers)**.** *A* prover $\mathcal{P}$ *consists of two uniform polynomial-time algorithms* $\mathcal{P}.\mathsf{prove}$ *and* $\mathcal{P}.\mathsf{verify}$ *with the following behavior:*

- $\mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda)$ *is randomized and takes as input a formula* φ *with* $|\varphi| \leq \lambda$.[23]
- $\mathcal{P}.\mathsf{verify}(\varphi, \pi, 1^\lambda)$ *is deterministic*[24] *and has* $\Pr_{\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda)}[\mathcal{P}.\mathsf{verify}(\varphi, \pi, 1^\lambda) = 1] = 1$ *if* $\varphi(w) = 1$.

Next, say a prover is perfectly sound if the verifier always rejects proofs of false statements.

Definition 2.3 (Perfect Soundness)**.** *A prover* $\mathcal{P}$ *is* perfectly sound *if* $\mathcal{P}.\mathsf{verify}(\varphi, \pi, 1^\lambda) = 0$ *for all* π, λ, *and unsatisfiable* φ.

Next, we define our notion of a simulator, which is just the non-asymptotic, pointwise version of the standard definition [GMR89]. Indeed, we stress that the definition below makes sense for any choice of natural numbers $\lambda, s, \frac{1}{\epsilon}$, which is why we call it pointwise.

Definition 2.4 (Pointwise Simulators)**.** *For* $\lambda, s, \frac{1}{\epsilon} \in \mathbb{N}$, *we say a prover* $\mathcal{P}$ *has an* s-size ϵ-indistinguishable simulator on λ *if there exists an* s*-size probabilistic circuit* Sim_λ *such that for all* φ *of size at most* λ *with* $\varphi(w) = 1$

$$Sim_\lambda(\varphi) \approx_\epsilon \mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda).$$

Here $\approx_\epsilon$ denotes computational indistinguishability (adversary circuits of size at most $\frac{1}{\epsilon}$ cannot distinguish with advantage ϵ).[25]

Many natural security definitions are equivalent to bounding the probability adversaries of a certain size can win a game. These are often referred to as falsifiable security properties, as defined by Naor [Nao03]. We will need a version of falsifiable security properties for provers.

Definition 2.5 (Game)**.** *A* game *is a (not necessarily efficient) randomized algorithm* $G(\mathcal{P}, A, 1^\lambda)$ *with*

- Input: *a prover (represented by its code), an adversary circuit* A, *and a security parameter* 1^λ.
- Output: *a bit (think of* G *as outputting* 1 *when the adversary "breaks" a security property).*

Now we are ready to define falsifiable properties of zero-knowledge. Roughly speaking, these are games with upper bounds on an adversary's success probability whenever there is a simulator.

Definition 2.6 (Falsifiable Property of Zero-Knowledge)**.** *A* falsifiable property of zero-knowledge Π *is a tuple* (G, Δ) *where* G *is a game and* Δ *is a polynomial-time computable function such that if* $\mathcal{P}$ *has an* s*-size* ϵ*-indistinguishable simulator on* λ, *then for all* A

$$\Pr[G(\mathcal{P}, A, 1^\lambda) = 1] \leq \Delta(\lambda, A, s, \epsilon).$$[26]

We say Π holds (respectively, holds with subexponential security) on a prover $\mathcal{P}^\star$ *if there exist functions* $s^\star = \mathsf{poly}(\lambda)$ *and* $\tau^\star = \frac{1}{\epsilon^\star} = \lambda^{\omega(1)}$ *(respectively,* $2^{\lambda^{\Omega(1)}}$*) such that for all* λ *and all* A

$$\Pr[G(\mathcal{P}^\star, A, 1^\lambda) \text{ outputs } 1 \text{ in time at most } \tau^\star\ ] \leq \Delta(\lambda, A, s^\star, \epsilon^\star) + \epsilon^\star.$$

[22] As we mentioned earlier, usually one would call this a "proof system," but we reserve that term for the notion of proof systems in proof complexity.

[23] We require that $|\varphi| \leq \lambda$ because it makes Definition 2.4 much cleaner.

[24] A verifier can always be derandomized if $\mathsf{P} = \mathsf{BPP}$.

[25] See Definition 4.2 for a more detailed definition of computational indistinguishability.

[26] We stress that the values λ, s, ϵ that Δ takes as input are concrete numbers $\lambda, s, \frac{1}{\epsilon} \in \mathbb{N}$.

As we will see in the next example, the requirement that Δ is efficiently computable is quite mild.

2.2 Illustrative Example of Definitions: Witness Hiding

We now give an example, both to illustrate the above definitions and to set up a future application. Informally, a prover is *witness hiding* [FS90] if getting a proof that φ is satisfiable does not help one find a satisfying assignment to φ. One way to formalize this is as follows [KZ20].

- Let $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ be a P-samplable distribution[27] on formulas and corresponding witnesses (i.e., the distribution outputs a pair (φ, w) with $\varphi(w) = 1$).
- *The Witness Hiding Game for $\mathcal{D}$* is the game $G(\mathcal{P}, A, 1^\lambda)$ given by:

  1. Sample $(\varphi, w) \leftarrow \mathcal{D}_\lambda$ and $\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda)$.
  2. Output 1 if and only if $A(\varphi, \pi)$ is a satisfying assignment to φ.

  We say a prover $\mathcal{P}$ is *witness hiding for $\mathcal{D}$* if there is an $S(\lambda) = \lambda^{\omega(1)}$ such that

  $$\Pr[G(\mathcal{P}, A, 1^\lambda)] \leq \frac{1}{S(\lambda)}$$

  whenever $|A| \leq S(\lambda)$.
- Of course, we can only hope to be witness hiding if it is actually hard to solve $\mathsf{Search\text{-}SAT}$ on $\mathcal{D}$.

  We say $\mathcal{D}$ is a *hard* $\mathsf{Search\text{-}SAT}$ *distribution* if there exists a polynomial-time computable function $S(\lambda) = \lambda^{\omega(1)}$ such that for all λ and all adversary circuits A of size $S(\lambda)$ we have that

  $$\Pr_{(\varphi,w)\leftarrow\mathcal{D}_\lambda}[\varphi(A(\varphi)) = 1] < \frac{1}{S(\lambda)}.$$

  We note that polynomial-time computability is an extremely mild requirement on S. For example, let $\log^\star$ denote the iterated logarithm. For every $S = \lambda^{\Omega(\log^\star \log^\star \lambda)}$, there is a constant $c > 0$ such that S is at least the polynomial-time computable superpolynomial function $c\lambda^{c \log^\star \log^\star \lambda}$.

It is easy to show that witness hiding is a falsifiable property of zero-knowledge, as long as the corresponding distribution is hard. We give a rigorous proof of this and encourage the reader to go through it, as it helps illustrate our definitions.

Proposition 2.7. *Let $\mathcal{D}$ be a hard* $\mathsf{Search\text{-}SAT}$ *distribution and G be its corresponding game. Then (G, Δ) is a falsifiable property of zero-knowledge for some $\Delta = (\lambda^{-\omega(1)} + \epsilon) \cdot \mathsf{poly}(|A|, \lambda, s)$.*

Proof. Fix a hard $\mathsf{Search\text{-}SAT}$ distribution $\mathcal{D}$ with corresponding polynomial-time computable function $S_0 = S_0(\lambda) = \lambda^{\omega(1)}$. Let $\mathcal{P}$ be a prover with an s-size ϵ-indistinguishable simulator Sim_λ on λ. Fix an adversary A. Then we have

$$\begin{aligned}\Pr_G[G(\mathcal{P}, A, 1^\lambda) = 1] &\leq \Pr_{(\varphi,w)\leftarrow\mathcal{D}_\lambda}[\varphi(A(\varphi, Sim_\lambda(\varphi))) = 1] + \epsilon \cdot \mathsf{poly}(|A|, \lambda) \\ &\leq (\frac{1}{S_0(\lambda)} + \epsilon) \cdot \mathsf{poly}(|A|, \lambda, s)\end{aligned}$$

where the first line is by the simulator guarantee and the second is by the definition of S_0. □

[27] This means there is a probabilistic polynomial-time Turing machine that on input 1^λ samples from the distribution $\mathcal{D}_\lambda$.

2.3 The (Weaker) Result

We have now defined all the terminology needed to understand the weaker version of our main result.

Theorem 2.1 (Weak Version of Main Result)**.** *Assume*

- $\mathsf{P} = \mathsf{BPP}$*,*
- *NIWIs exist (respectively, subexponentially secure NIWIs exist), and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

For every falsifiable property of zero-knowledge Π*, there exists a perfectly sound prover for which* Π *holds (respectively, holds with subexponential security).*

The following corollary follows easily from Theorem 2.1, Proposition 2.7, and the definition of witness hiding.

Corollary 2.8. *Assume the assumptions in Theorem 2.1 hold. For every hard* Search-SAT *distribution, there exists a perfectly sound prover that is witness hiding for it.*

While the prover in Corollary 2.8 depends on the underlying distribution, we do not view this as a deficiency. This is because it seems likely that such a dependence is actually necessary (see the discussion in Appendix A) and because of what we will discuss in Section 2.5 (a single prover with every "natural" falsifiable property of zero-knowledge).

Exponential-time games: from Search-NP to TFNP. Theorem 2.1 is especially powerful in the subexponential regime. We illustrate this with another corollary: one can generically convert Search-NP problems to TFNP problems, while preserving worst-case hardness.

Corollary 2.9 (Informal Corollary 7.28)**.** *Assume the subexponential assumptions in Theorem 2.1 hold. For every* Search-NP *problem, there is a corresponding* TFNP *problem with matching worst-case hardness.*

We briefly sketch the proof of this corollary for the special case of Search-SAT (i.e., output a satisfying assignment to a given φ, whenever one exists). One can consider a natural TFNP version of Search-SAT, where one is given φ and also a proof π (in some chosen proof system) of "φ is satisfiable." To prove hardness of this TFNP version, we consider the following game, which outputs 1 if the given adversary solves the problem in the worst-case.

$G(\mathcal{P}, A, 1^\lambda)$:

1. Suppose A takes s-size formulas as input. For all φ of size s:
   (a) By brute force, find a satisfying assignment w to φ. If none exists, then go to the next φ.
   (b) Using $\mathsf{P} = \mathsf{BPP}$, deterministically estimate (to within additive error .01)
   $$\Pr_{\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda)}[A(\varphi, \pi) \text{ outputs a satisfying assignment to } \varphi].$$
   (c) Output 0 if the estimate is less than 2/3.
2. Output 1.

One can show (see Claim 7.29) that $\Pi = (G, \Delta)$ is a falsifiable property of zero-knowledge for some Δ that roughly corresponds to the worst-case circuit complexity of Search-SAT.

On the other hand, G runs in time $\mathsf{poly}(2^{s \log s}, |A|, \lambda)$. At first, this may seem bad because G runs in exponential time and Theorem 2.1 only offers subexponential security. But this just means we need to set the

security parameter λ appropriately (as is often done in proofs using subexponential security). In particular, by setting $\lambda = \mathsf{poly}(s)$ sufficiently large, the game will run in time $\mathsf{poly}(2^{\lambda^\delta}, |A|)$ for an arbitrarily small δ. Appealing to the subexponential version of Theorem 2.1 completes our sketch.

In the remainder of this section, we address two aspects of Theorem 2.1:

- *But does it work for X scenario?* In Section 2.4, we discuss how to use Theorem 2.1 and its extensions for applications where it is a priori not clear how to use it.
- *A single "universal" prover?* The prover in the statement of Theorem 2.1 can depend on the desired falsifiable property of zero-knowledge. Ideally, one would want a single "universal" prover with every falsifiable property of zero-knowledge. We discuss this in Section 2.5.

2.4 Using the Result

What if a security property needs both a simulator and soundness? We now address a potential concern with our definition, which is best explained with an example. Consider the following somewhat artificial protocol related to oblivious transfer.[28] Let $Commit$ be a commitment scheme[29] that is perfectly binding and computationally hiding. Let $\mathcal{P}$ be an arbitrary prover.

Example Protocol

Alice is given $x \leftarrow \{0,1\}^2$. Bob is given $b \leftarrow \{0,1\}$.

1. Bob sends Alice a commitment $c_b \leftarrow Commit(b)$.
2. Alice sends Bob a commitment $c_x \leftarrow Commit(x)$.
3. Bob sends the bit b and also a $\mathcal{P}$-proof that c_b was a commitment to b.
4. Alice sends the bit x_b and gives a $\mathcal{P}$-proof π_x that this is consistent with the commitment c_x.[a]

[a] In more detail, Alice proves that there exists a string consistent with the commitment whose b'th bit is x_b.

Now suppose that Alice wants to be sure that if Bob acts maliciously in step (3) — but is otherwise honest — he cannot learn $x_{b\oplus 1}$ with non-negligible advantage. This is easy to see if we replace $\mathcal{P}$ with a standard (interactive) zero-knowledge proof. Critically, however, this relies on interactive zero-knowledge having both soundness and a simulator. In particular, we need soundness so that Bob does not lie about b in step (3). On the other hand, we need a simulator in order to say that π_x does not leak $x_{1\oplus b}$ in step (4).

In contrast, our notion of a falsifiable property of zero-knowledge only captures properties that follow from just a simulator existing. As a result, the straightforward way to model this as a game G — where Bob is the adversary and winning corresponds to distinguishing $x_{b\oplus 1} = 0$ from $x_{b\oplus 1} = 1$ — does not fit into our framework of a falsifiable property of zero-knowledge.

But this is easily overcome. To do so, we consider a modified game G'. G' is exactly like G except that G' outputs zero if Bob lies in step (3). Note that G' knows b (it samples it and simulates running Bob on it), so it can easily tell if Bob lies. With this modification, $\Pi' = (G', \Delta)$ is easily shown to be a falsifiable property of zero-knowledge with $\Delta = 1/2 + (\lambda^{-\omega(1)} + \epsilon) \cdot \mathsf{poly}(|A|, s, \lambda)$.[30] Thus, Theorem 2.1 guarantees there is a perfectly sound prover $\mathcal{P}^\star$ on which Π' holds. Now, because $\mathcal{P}^\star$ is perfectly sound, the assumption that Bob does not lie at step (3) is without loss of generality. In other words, any adversary that succeeds on G with prover $\mathcal{P}^\star$ must also succeed on G' with prover $\mathcal{P}^\star$. Hence, $\Pi = (G, \Delta)$ holds for $\mathcal{P}^\star$. This completes the argument.

[28] The reader does not need to know what oblivious transfer [Rab05] is to understand the example.

[29] If the reader is not familiar with such schemes, think of $Commit$ as follows. It is a randomized algorithm with the property that the output $Commit(x)$ completely determines x information-theoretically but hides x computationally.

[30] We assume that the security of the commitment scheme is at least some efficiently computable superpolynomial function.

It is worth noting that, in this example, we used the fact that we can easily tell when Bob lies (because the game gets access to Bob's input and random coins). This is true for many examples. But, even if this is not true, one can generically use complexity leveraging to check whether a given statement is true via brute force. This incurs only a polynomial blow-up if the subexponentially-secure version of Theorem 2.1 holds.

What if a security property needs a uniform simulator? Our definitions view simulators as circuits, with no assumption about uniformity. This suffices for most settings in cryptography, as one usually considers security against non-uniform adversaries.

It turns out, however, that our results hold even if one restricts to simulators with only about $|\psi_\lambda|$ bits of non-uniformity, where ψ_λ are the sequence of formulas that parameterize our construction.[31] As a result, if one makes, say, subexponential assumptions, one can afford to set $|\psi_\lambda| = \mathsf{poly}\log\lambda$ and still get $\lambda^{\omega(1)}$ security. This means a version of our results holds even when restricting simulators to $\mathsf{poly}\log\lambda$ bits of non-uniformity (see Theorem 7.16). Similarly, one could also restrict to uniform simulators running in, say, quasipolynomial time by brute forcing over all potential satisfying assignments to ψ_λ.

2.5 A Single Universal Prover?

In Theorem 2.1, the prover depends on the precise falsifiable property one wants to hold. Could there be a "universal" prover with *every* falsifiable property of zero-knowledge? Unfortunately, this is likely impossible (see Proposition 7.10) because of the aforementioned attack [BOV07] that "a verifier gains the ability to prove the same statement to others." Indeed, a universal prover seems unlikely even for witness hiding (see Appendix A).

On the other hand, *our results suggest there is a single (uniform) prover that enjoys every "natural" falsifiable property of zero-knowledge!* In an attempt to make this formal, let $\mathsf{Natural}$ be the collection of all falsifiable properties of zero-knowledge whose underlying game has appeared in the literature prior to this work (and does not depend on our choice of prover or involve statements about its verifier[32]). We stress that, while the definition of $\mathsf{Natural}$ is not formal, everything we discuss below can be made formal modulo defining $\mathsf{Natural}$.

Definition 2.10. *Say a prover $\mathcal{P}_{ultimate}$ is an* ultimate prover *if every $\Pi \in \mathsf{Natural}$ holds for $\mathcal{P}_{ultimate}$.*

At this point, we will need a little bit of proof complexity, but we hope that what we say is intuitive enough to understand at a high level. It turns out that the prover guaranteed in Theorem 2.1 depends only on what axioms one needs in order to prove a formalization of the statement "Π is a falsifiable property of zero-knowledge." If one believes that a single set of polynomial-time decidable axioms (for example, ZFC) should prove all of these natural statements (perhaps up to some slack in Δ) — as appears to be the working hypothesis in complexity theory — then Theorem 2.1 gives a single prover that enjoys every "natural" falsifiable security property (up to the same slack in Δ).

Theorem 2.11 (Informal Version of Theorem 7.12)**.** *If $\mathsf{P} = \mathsf{BPP}$, NIWIs exist, and a single non-optimal proof system proves "Π is a falsifiable property of zero-knowledge" for every $\Pi \in \mathsf{Natural}$, then an ultimate prover exists.*

In fact, one can relax this even further. We actually do not need that, say, ZFC can prove $X =$"Π is a falsifiable property of zero-knowledge." It (roughly) suffices to have a uniform sequence of unsatisfiable formulas ψ_λ with the property that ZFC lacks a short proof of "if X, then ψ_λ is unsatisfiable." We formalize this in Theorem 7.14.

Furthermore, we propose an explicit candidate for such a sequence ψ_λ: formulas expressing[33] the consistency of ZFC_+ on λ-length proofs. Here, ZFC_+ denotes adding the axiom that ZFC is consistent to ZFC.

[31] Indeed, the non-uniformity precisely corresponds to a potential satisfying assignment to ψ_λ.

[32] We do this to eliminate obvious counterexamples. For example, the ability, given a proof of a statement, to prove the statement to others.

[33] See Section 3.4 for a formal definition.

Our choice is inspired by Pudlák's work [Pud86] on a finitary analogue of Gödel's incompleteness theorem, which we discuss in Section 3.4. Our intuition for this choice is very simple. We believe ZFC lacks short proofs of the consistency of ZFC_+ (see Conjecture 3.13). Why should a natural statement in cryptography help ZFC prove the consistency of ZFC_+?

Conjecture 2.12 (Informal)**.** *Instantiating our construction with formulas corresponding to the consistency of ZFC_+ and with a subexponentially secure NIWI construction that existed prior to this work*[34] *yields an ultimate prover.*

(This conjecture is not formal because the definition of Natural is not formal.)

3 An Exposition of Our Results: Part 2 (With Proof Complexity)

In this section, we will discuss our main result, stated below as a preview. In the coming subsections, we will explain the background needed to understand the statement.

Theorem 3.1 (Main Result)**.** *Assume*

- *NIWIs (respectively, subexponentially-secure NIWIs) exist, and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

Then for every proof system $\mathcal{L}$, there exists a perfectly sound prover $\mathcal{P}$ that is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$.

3.1 Proof Complexity Background: Proof Systems

To start, we need the notion of a (proof complexity) proof system, as defined by Cook and Reckhow [CR79]. Cook and Reckhow's definition is extremely broad. We will give a formal definition next, but essentially they just require the following.

- *Statements are Language Membership*: A "statement" is just an assertion of the form "$x \in L$" for some fixed but arbitrary language L. For example, L might be SAT, UNSAT (the complement of SAT), or HALT (the language for the Halting problem).[35]
- *Soundness*: If there is a proof of a statement, then that statement is true.
- *Polynomial-Time Checkability*: There is a polynomial-time Turing machine that checks if a given proof proves a given statement.

For our purposes, we do not require completeness.

Based on the properties above, one might expect the definition of a proof system to be a polynomial-time machine $\mathcal{L}$ that takes as input a purported proof π, a statement x, and either accepts (implying $x \in L$) or rejects. Cook and Reckhow's definition is essentially equivalent but slightly nicer to work with.

Definition 3.2 (Proof System [CR79])**.** *A* proof system for a language L *is a polynomial-time algorithm $\mathcal{L} : \{0,1\}^\star \to L$. We say there is an ℓ-*length $\mathcal{L}$-proof of x *if $\mathcal{L}(\pi) = x$ for some $|\pi| \leq \ell$.*

Essentially, $\mathcal{L}$ takes as input a "proof" $\pi \in \{0,1\}^\star$ and outputs a "theorem statement" $x \in L$. There is no notion of an "invalid proof," but there is also no guarantee that the output of $\mathcal{L}$ is "interesting." The fact that the range of $\mathcal{L}$ is contained in L enforces soundness.

[34] We add this constraint to avoid pathological NIWI constructions (e.g., ones that are only secure if ZFC_+ is consistent).

[35] In the literature, the term "Cook-Reckhow proof system" often refers to the special case where L is the coNP-complete language TAUT. However, Cook and Reckhow's original definition was for an arbitrary language L, which is the definition we use in this paper.

Examples of Cook-Reckhow Proof Systems. To illustrate the concept, we give a few examples.

- *Constant Output:* Let x be an element of language L. Then the constant Turing machine $\mathcal{L}_x$ given by $\mathcal{L}_x(\pi) = x$ is a proof system for L.
- *The* SAT *Verifier*: The polynomial-time Turing machine $\mathcal{L}$ with the following behavior is a proof system for SAT.

$$\mathcal{L}(\varphi, x) = \begin{cases} \varphi, & \text{if } \varphi(x) = 1 \\ \text{a fixed trivially satisfiable formula (e.g. } \psi = x_1), & \text{otherwise.} \end{cases}$$

- *ZFC*: Roughly, a proof π in ZFC consists of a sequence of lines $\phi_1, \ldots, \phi_\ell$, where each line is either an axiom or can be obtained by applying one of finitely many inference rules (e.g., modus ponens) to previous lines. The final line in the proof is the statement that π proves. The axioms of ZFC have the property that one can decide if ϕ is an axiom of ZFC in polynomial-time. Indeed, the only fact about ZFC that we will need for this bullet point is that one can check whether a given π is a valid proof in polynomial time.

  The polynomial-time Turing machine $\mathcal{L}$ with the following behavior is a proof system[36] for HALT (it is easily generalized to other languages).

$$\mathcal{L}(\pi, M, x) = \begin{cases} (M, x), & \text{if } \pi \text{ is a ZFC proof that ``}M(x) \text{ halts''} \\ \text{a fixed trivially halting machine ,} & \text{otherwise.} \end{cases}$$

- *The ultimate cryptographic system: ZFC + Axioms for {DDH, LWE, RSA, ...}*: Suppose one believes a certain cryptographic assumption Y is true. Currently, we may not know if Y is provable in ZFC. But one could always add Y as an axiom to ZFC. Indeed, assuming Y is true (so that we maintain soundness), then the polynomial-time Turing machine $\mathcal{L}$ with the following behavior is a proof system for HALT (it is easily generalized to other languages).

$$\mathcal{L}(\pi, M, x) = \begin{cases} (M, x), & \text{if } \pi \text{ is a ZFC-proof of ``if } Y\text{, then } M(x) \text{ halts''} \\ \text{a fixed halting machine,} & \text{otherwise.} \end{cases}$$

Conventions for Proof Systems in This Paper. We now make some choices specific to this paper.

1. *Choice of Language:* $\mathsf{P}^{\mathsf{HALT}}$. In this paper, we are free to choose as powerful a language L as we wish for our proof systems. Indeed, since we incur no cost whatsoever for our choice, we go somewhat overboard, as it makes our life as easy as possible. We choose L to be the natural complete problem for $\mathsf{P}^{\mathsf{HALT}}$: the circuit evaluation problem with HALT-oracle gates.[37] In other words, L is the set

$$\{(C, x) : C^{\mathsf{HALT}}(x) = 1\}.$$

   We abuse notation and refer to this language as $\mathsf{P}^{\mathsf{HALT}}$. We choose $\mathsf{P}^{\mathsf{HALT}}$ for two reasons.

   - *Analyzing Computation*: We will want to be able to prove statements about computation that are obviously encodable as instances of HALT. Choosing HALT instead of a time-bounded version lets us avoid having to keep track of time complexity.
   - *Closure Under Implications:* We want to be able to prove statements like "if X, then Y," where X and Y are themselves statements. Recall, "if X, then Y" really means "(not X) or Y." Thus, we want to choose a language L such that ($x \notin L$ or $y \in L$) is equivalent to $z \in L$ for some z that we can compute in, say, polynomial-time from x and y. A generic way to take a language L and give it this property is to consider circuit evaluation with L-oracle gates.

[36] Here, we are implicitly assuming that ZFC is sound on such statements, as we do throughout the paper.

[37] If the reader is uncomfortable with the fact that HALT is undecidable, the results in the paper can be made to go through by replacing $\mathsf{P}^{\mathsf{HALT}}$ with bounded-time variants of the Halting problem.

Henceforth, unless otherwise stated, a proof system refers to a proof system for $\mathsf{P}^{\mathsf{HALT}}$.

Definition 3.3 (Proof System)**.** *A proof system (with L omitted) refers to a proof system for $\mathsf{P}^{\mathsf{HALT}}$.*

2. *Notation for Statements: Double Quotes.* We usually put statements being proved in double quotes and write an English language or mathematical description of the statement (the corresponding encoding as an instance of $\mathsf{P}^{\mathsf{HALT}}$ should be clear).

3. *Sufficiently Strong.* It will be helpful to assume that the proof systems we consider are strong enough to prove some useful facts. We stress this can be done without loss of generality (essentially by just adding axioms). In particular, throughout the paper, we assume any proof system we consider:

   - *Can simulate ZFC.* It will be helpful that $\mathcal{L}$ is at least strong enough to carry out some basic reasoning. To be concrete, we assume that $\mathcal{L}$ is at least as strong as, say, ZFC.
     (Formally, if there is an ℓ-length $\mathcal{L}$-proof of "X" and there is an ℓ'-length ZFC-proof of "if X, then Y," then there is a $\mathsf{poly}(\ell, \ell')$-length $\mathcal{L}$-proof of "Y." Any proof system $\mathcal{L}$ can be transformed into one that has this property (see Proposition 4.1).)
   - *Knows $\mathsf{P} = \mathsf{BPP}$ (when we assume it).* Sometimes it will be helpful to assume that $\mathsf{P} = \mathsf{BPP}$, in which case there is a deterministic polynomial-time algorithm $\mathsf{Estimate}(M, x, 1^k)$ that takes as input the code of a probabilistic Turing machine M, a string x, and a parameter k and outputs a value v such that
     $$\left| v - \Pr_M[M(x) \text{ outputs 1 in time at most } k] \right| \leq \frac{1}{k} \tag{2}$$
     Without loss of generality, any proof system can be transformed into one that also can prove the correctness of $\mathsf{Estimate}$. In other words, for some constant c, it can prove
     $$\text{"for all } M, x \text{ and } k, \mathsf{Estimate}(M, x, 1^k) \text{ runs in time } (|M| + |x| + k)^c \text{ and satisfies (2)."}$$
     We note that the above statement can be encoded as an instance of $\mathsf{P}^{\mathsf{HALT}}$.

3.2 Relaxing Truth

A useful way of thinking about our results is via a certain way of relaxing truth. In particular, we give a definition that, roughly speaking, relaxes the requirement that

$$\text{"}X \text{ is true"}$$

to the requirement that

$$\text{every "}t\text{-time consequence of } X \text{ being true" is true.}$$

We will apply this notion to relax the requirement that a zero-knowledge simulator exists.

To formalize a t-time consequence, let U be an efficient universal Turing machine (recall, this means that U takes as input $M = (M_0, x)$, where M_0 is a description of a Turing machine and x is a string, and simulates running $M_0(x)$). Let $U^t(M)$ denote cutting off $U(M)$ after t steps and outputting zero if it did not already output something.

Definition 3.4 (t-time Indistinguishable From True)**.** *Let $\mathcal{L}$ be a proof system. Let X be a statement. We say that X is* t-time indistinguishable from true to $\mathcal{L}$ *if the following holds. For all M, if there is a t-length $\mathcal{L}$-proof that*
$$\text{"if } X, \text{ then } U^t(M) = 1,\text{"}$$
then $U^t(M) = 1$.

We make a few remarks on this definition.

- *We can specify t in binary.* We note that we can encode the statement "if X, then $U^t(M) = 1$" as a $\mathsf{poly}(|X|, \log t, |M|)$-length instance of $\mathsf{P}^{\mathsf{HALT}}$. Note that the dependency on t is logarithmic. This is important because otherwise the t-length proof requirement would be too stringent.
- *t will be large.* As we will see soon, t will be large (i.e., superpolynomial in our security parameter).
- *X is often false.* In this paper, we will usually be in the case where X is *false*. This means that (if $\mathcal{L}$ is sufficiently strong) one can prove that X implies anything, including false statements. This definition says that if there is a *short* proof that "X implies a t-time statement Y," then Y must be true.
- *We will usually write $M(y)$ instead of $U^t(M, y)$.* For readability, we will usually write $M(y)$ instead of $U^t(M, y)$.
- *Close relationship to unprovability.* This definition is essentially equivalent to "$\neg X$" lacking a short $\mathcal{L}$-proof.
  - If $\mathcal{L}$ is sufficiently strong and has a short proof that "X implies $U^t(M) = 1$" but $U^t(M) \neq 1$, then one gets a short proof of "$\neg X$."[38]
  - If "$\neg X$" has a short $\mathcal{L}$-proof and $\mathcal{L}$ is sufficiently strong, then $\mathcal{L}$ has a short vacuous proof of "if X, then $U^t(M) = 1$" for all M (even ones that trivially have $U^t(M) \neq 1$).

  We use the t-time indistinguishable from true definition rather than the unprovability definition because it is both more operationally useful in proofs and because we feel it more intuitively reflects what is happening in our results.

Example: Non-asymptotic One-Way Functions. Now we give an example to illustrate this definition. Recall that a circuit $C : \{0,1\}^n \to \{0,1\}^m$ is an S-secure one-way function if for every S-size adversary circuit A we have that

$$\Pr_{x \leftarrow \{0,1\}^n, y = C(x)}[C(A(y)) = y] < \frac{1}{S}.$$

We observe that if a statement X is indistinguishable from true and if X provably implies C is an S-secure one-way function, then C is indeed a one-way function (with a tiny security loss). This argument works more generally for non-asymptotic falsifiable properties.

Proposition 3.5. *Assume $\mathsf{P} = \mathsf{BPP}$. Let $\mathcal{L}$ be a (sufficiently strong) proof system. Let $S \in \mathbb{N}$ and C be a circuit.*[39] *If there is an ℓ-length $\mathcal{L}$-proof that "if X, then C is an S-secure one-way function" and if X is $\mathsf{poly}(\ell, S)$-time indistinguishable from true to $\mathcal{L}$, then C is an $S/3$-secure one-way function.*

Proof. Let $G(C, A)$ be the probabilistic Turing machine that, given C and A, samples $x \leftarrow \{0,1\}^n$, sets $y = C(x)$ and outputs 1 if $C(A(y)) = y$. Recall that f being an S-secure one-way function means that $\Pr[G(C, A) = 1] \leq \frac{1}{S}$ for all $|A| \leq S$.

Now fix an adversary A of size at most $S/3$. Using that $\mathcal{L}$ is sufficiently strong (enough to simulate ZFC), the assumed ℓ-length proof implies that there is also a $\mathsf{poly}(\ell, |C|, S) = \mathsf{poly}(\ell, S)$-length[40] $\mathcal{L}$-proof that

$$\text{"if } X\text{, then } \Pr[G(C, A) = 1] < \frac{1}{S}\text{."}$$

Next, since $\mathcal{L}$ is sufficiently strong (enough to show $\mathsf{P} = \mathsf{BPP}$), we get that there is a $\mathsf{poly}(\ell, S)$-length $\mathcal{L}$-proof that

$$\text{"if } X\text{, then } \mathsf{Estimate}(G, (C, A), 1^S) \leq \frac{2}{S}\text{."}$$

[38] The proof is just run $U^t(M)$, see that it does not output 1, and then use the contrapositive of "X implies $U^t(M) = 1$."

[39] We stress that S is a single number and C is a single circuit. We are not viewing them as asymptotic objects here.

[40] The fact that $|C| = \mathsf{poly}(\ell)$ follows from the following argument. By assumption there is an ℓ-length proof of "if X, then C is an S-secure one-way function." This means there exists a $\pi \in \{0,1\}^{\leq \ell}$ such that $\mathcal{L}(\pi) =$ "if X, then C is an S-secure one-way function." Hence, because $\mathcal{L}$ runs in polynomial time, it must be the case that $|C| = \mathsf{poly}(\ell)$.

Then because $\mathsf{Estimate}(G,(C,A),1^S)$ runs in time $\mathsf{poly}(S,|G|,|C|,|A|) = \mathsf{poly}(S,\ell)$ and because X is $\mathsf{poly}(\ell,S)$-time indistinguishable from true to $\mathcal{L}$, we get that

$$\mathsf{Estimate}(G,(C,A),1^S) \leq \frac{2}{S}$$

and hence that $\Pr[G(C,A)=1] \leq \frac{3}{S}$. □

3.3 Our Main Result

We will now formally state our relaxation of zero-knowledge. We note that it relies on definitions made in Section 2.1 and Section 3.2.

Definition 3.6 (Effectively Zero-Knowledge to $\mathcal{L}$)**.** *Let $\mathcal{P}$ be a prover, and let $\mathcal{L}$ be a proof system. We say $\mathcal{P}$ is* effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ *if for some $t = \lambda^{\omega(1)}$ (respectively, $t = 2^{\lambda^{\Omega(1)}}$) and $s = \mathsf{poly}(\lambda)$ we have that for all $\lambda \in \mathbb{N}$*

$$\text{“}\mathcal{P} \text{ has an } s(\lambda)\text{-size } \frac{1}{t(\lambda)}\text{-indistinguishable simulator on } \lambda\text{”}^{41} \tag{3}$$

is $t(\lambda)$-time indistinguishable from true to $\mathcal{L}$.

We now have all the background needed to understand the conclusion of our main result.

Theorem 3.1 (Main Result)**.** *Assume*

- *NIWIs (respectively, subexponentially-secure NIWIs) exist, and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

Then for every proof system $\mathcal{L}$, there exists a perfectly sound prover $\mathcal{P}$ that is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$.

We remark that Theorem 3.1 itself does not assume $\mathsf{P} = \mathsf{BPP}$. But it is often helpful to assume $\mathsf{P} = \mathsf{BPP}$ when one wants to use Theorem 3.1.

One drawback of Theorem 3.1 is that it is existential. It only guarantees the existence of such a $\mathcal{P}$, but it does not say how to construct $\mathcal{P}$. Under a variant of a conjecture of Khaniki [Kha24], we show a constructive version of Theorem 3.1 holds: given the code of $\mathcal{L}$, one can efficiently find a $\mathcal{P}$ that is effectively zero-knowledge to $\mathcal{L}$ (see Theorem 7.18). In the next subsection, we also describe a concrete candidate for a $\mathcal{P}$ that is effectively zero-knowledge to ZFC.

3.4 Proof Complexity Background: Optimal Proof Systems

In this subsection, we discuss the background needed to understand our assumption about optimal proof systems. A key notion in proof complexity is *simulation* [CR79], which gives a natural partial ordering on the power of proof systems.

Definition 3.7 (Simulating a Proof System [CR79])**.** *Let $\mathcal{L}$ and $\mathcal{L}'$ be proof systems for a language L. We say that $\mathcal{L}$* simulates *$\mathcal{L}'$ if there is a polynomial p such that the following holds. For every X, if there is an ℓ-length $\mathcal{L}'$-proof of “X,” there is also a $p(\ell)$-length $\mathcal{L}$-proof of “X.”*

An *optimal proof system* is one that simulates all other proof systems [KP89].

Definition 3.8 (Optimal Proof System [KP89])**.** *Let $\mathcal{L}$ be a proof system for a language L. We say that $\mathcal{L}$ is* optimal *if it simulates every proof system $\mathcal{L}'$ for L.*

[41] In Remark 5.1, we clarify potential ambiguities arising from notation in (3).

This definition can be straightforwardly extended to the infinitely often or subexponential settings (see Definition 6.7).

A natural question is whether an optimal proof system exists for a given language. An immediate consequence of Gödel's incompleteness theorem is that there is no optimal proof system for UNHALT, the complement of the halting problem.[42] There is an easy proof based on the undecidability of HALT.

Theorem 3.9 ([Göd31; Tur37])**.** *There is no optimal proof system for* UNHALT*.*

Proof. For every $(M, x) \in \mathsf{UNHALT}$, there exists a proof system for UNHALT that proves "$M(x)$ does not halt" (you can take the constant machine that always outputs (M, x)). Hence, if $\mathcal{L}$ were an optimal proof system for UNHALT, it must be the case that $(M, x) \in \mathsf{UNHALT}$ if and only if $\mathcal{L}(\pi) = (M, x)$ for some π.

Then given (M, x), one can solve the halting problem by running the following two algorithms in parallel:

- simulate $M(x)$ and accept if it ever halts
- try all possible π and reject if $\mathcal{L}(\pi) = (M, x)$.

This contradicts the undecidability of the halting problem. □

It turns out this means that for every proof system $\mathcal{L}$ for UNHALT, there is another proof system $\mathcal{L}'$ for UNHALT such that $\mathcal{L}'$ simulates $\mathcal{L}$ but not vice versa. In fact, Gödel shows that if $\mathcal{L}$ is a "sufficiently nice" proof system, then one can take $\mathcal{L}'$ to be $\mathcal{L}$ with the extra axiom that "$\mathcal{L}$ is consistent." Here, "$\mathcal{L}$ is consistent" means that it never proves both a statement and its negation (note that consistency is naturally encoded as an instance of UNHALT).

Proof Systems for UNSAT**.** The field of proof complexity is primarily interested in the proof systems for UNSAT, referred to as *propositional proof systems*.[43] One of the main conjectures in the field is that there is no optimal proof system for UNSAT. (We often omit saying "propositional" when referring to this conjecture to match the literature.)

Conjecture 3.10 (No Optimal (Propositional) Proof System [KP89])**.** *There is no optimal propositional proof system.*

The following is one of the main results in [KP89] and is very useful for us. It roughly says that if no optimal proof system exists, then there is a uniform sequence of (propositional) statements hard to prove in any given proof system.

Theorem 3.11 (Krajíček and Pudlák [KP89])**.** *Assume there is no optimal proof system. Then for every propositional proof system $\mathcal{L}$, there exists a* P*-uniform*[44] *sequence of unsatisfiable formulas ψ_λ such that $\mathcal{L}$ lacks* poly(λ)*-length $\mathcal{L}$-proofs of "ψ_λ is unsatisfiable."*

Indeed, the sequence ψ_λ corresponds to the consistency of a proof system $\mathcal{L}'$ that $\mathcal{L}$ does not simulate. As this suggests, Conjecture 3.10 is closely related to longstanding conjectures about extending Gödel's incompleteness theorem.

In more detail, in his 1986 paper, Pudlák [Pud86] investigates whether "scaled down" versions of Gödel's incompleteness theorem hold. Specifically, for a proof system $\mathcal{L}$, let $Con_\lambda(\mathcal{L})$ be the formula that is unsatisfiable if and only if $\mathcal{L}$ is consistent on proofs of length at most λ. Based on Gödel's incompleteness theorem, one might then expect that for all λ, there are, for example, no short ZFC-proofs that "$Con_\lambda(ZFC)$ is unsatisfiable." Somewhat surprisingly, Pudlák [Pud86] showed that this is *false*. But Pudlák [Pud86] conjectured that a modification of this should hold (see also [Pud17] and [Pud13, Conjecture 5]).

[42] In fact, it rules out optimality even under a much weaker notion of simulation, where statements provable in $\mathcal{L}'$ just need to be provable in $\mathcal{L}$, without any length considerations.

[43] In proof complexity, one usually considers the language of tautologies $\mathsf{TAUT} = \{\varphi : \varphi(x) = 1 \ \ \forall x\}$ instead of $\mathsf{UNSAT} = \{\varphi : \varphi(x) = 0 \ \ \forall x\}$. Both languages are coNP-complete, so the choice makes no difference. We use UNSAT because it is both more familiar to cryptographers and because it makes our construction slightly cleaner.

[44] This means that there is a polynomial-time algorithm that outputs ψ_λ on input 1^λ.

Conjecture 3.12 (Informal Finite Gödel Conjecture [Pud86, Problem 1])**.** *Let $\mathcal{L}$ be a "sufficiently nice" proof system. Let $\mathcal{L}_+$ be the proof system corresponding to $\mathcal{L}$ with the axiom that $\mathcal{L}$ is consistent added. Then there is no* $\mathsf{poly}(\lambda)$*-length $\mathcal{L}$-proof of "$Con_\lambda(\mathcal{L}_+)$."*

Conjecture 3.12 immediately implies the non-existence of optimal proof systems (see, e.g., [Pud13, Section 6.4]). Moreover, Mycielski conjectures (see [Pud86] and the discussion in [Pud13]) that the length of these proofs should be exponential in λ, which implies there are no subexponentially optimal proof systems. In particular, the following conjecture seems reasonable.

Conjecture 3.13 (Finite Gödel Conjecture for ZFC)**.** *The length of the shortest ZFC-proof that $Con_\lambda(ZFC_+)$ is $2^{\lambda^{\Omega(1)}}$.*

Motivated by this, we make the following conjecture.

Conjecture 3.14. *Our construction, instantiated with a subexponentially secure NIWI construction that existed prior to this work*[45] *and the sequence $Con_\lambda(ZFC_+)$, is effectively subexponentially zero-knowledge to ZFC.*

3.5 Cryptography Background: Ideas We Build On

We now give an overview of the main cryptographic ideas our construction builds on.

Witness Indistinguishability. Feige and Shamir [FS90] defined the notion of *witness indistinguishability* (WI). In contrast to zero-knowledge (ZK), WI has nice composition properties, which were partly Feige and Shamir's motivation. They also show it suffices for some important applications of ZK [FS90].

OR proofs. At first, the ability to do WI proofs seems much weaker than the ability to do ZK proofs. Contrary to this, Feige, Lapidot, and Shamir [FLS90] show how to get ZK guarantees by using WI. They do so using the following "OR proof" idea.

"OR proof" [FLS90]

Parameterized by a formula ψ.

1. The prover outputs a WI proof π that "$\varphi \vee \psi$ is satisfiable" using its witness for φ.

The key to their analysis is that

- **if ψ is satisfiable at, say, x:** then one can simulate π by giving a WI proof that "$\varphi \vee \psi$ is satisfiable" using the witness x. Thus, in this case, the protocol is zero-knowledge.
- **if ψ is unsatisfiable:** the construction will be sound because $\varphi \vee \psi$ being satisfiable implies that φ is satisfiable.

Feige, Lapidot, and Shamir [FLS90] then choose ψ to be the statement that a uniformly random string r is in the range of a length-doubling PRG G. Thus, with high probability, ψ is unsatisfiable, leading to soundness. But one can simulate the protocol by choosing $r = G(x)$.

Non-interactive Witness Indistinguishability. Building on the work of Dwork and Naor [DN07], Barak, Ong, and Vadhan [BOV07] show (perhaps surprisingly) that one can produce traditional mathematical proofs that are witness indistinguishable. In particular, they show (under plausible assumptions) that there is a zero-interaction prover that is perfectly sound and witness indistinguishable. Follow-up work [GOS12; BP15] gives constructions under different assumptions.

[45] We add this constraint to avoid pathological NIWI constructions (e.g., ones that are only secure if ZFC_+ is consistent).

A new analysis of the "OR proof." Recall that a proof π is witness hiding (WH) [FS90] if (roughly speaking) a proof π that "φ is satisfiable" does not help an adversary find a satisfying assignment to φ.

Kuykendall and Zhandry [KZ20] study the possibility of using NIWIs to achieve truly non-interactive WH proofs (NIWH), whose existence was unclear. One of their key ideas is a novel analysis of the OR proof above. They choose ψ to be unsatisfiable so that soundness holds. But they observe that if WH does not hold on a specific distribution $\mathcal{D}$ of instances ψ, then this can be used to certify that ψ is unsatisfiable (since if ψ is satisfiable, the proof is simulatable and hence witness hiding). But under the widely-believed complexity assumption $\mathsf{NP} \neq \mathsf{coNP}$ (and its randomized extension), every unsatisfiable formula ψ cannot have a certificate that it is unsatisfiable. Thus, there must exist *some* ψ such that this construction is witness hiding. Moreover, their argument generalizes to falsifiable security properties.

Kuykendall and Zhandry [KZ20] use this to construct *distribution-dependent non-uniform* NIWH. They remark (with notation changed to match the presentation above):

> "Unfortunately, we cannot use this protocol in a uniform setting as the ψ needed to achieve witness hiding may be hard to compute. Furthermore, the choice of ψ is not universal; it depends on the underlying distribution $\mathcal{D}$ from which the statements are drawn. Thus the construction is not a single witness hiding proof system for NP, but rather a family of proof systems, one for each hard distribution. Non-uniform protocols should be viewed as existential results: unlike common reference string protocols, the non-uniform model does not require the joint input to be sampleable.
>
> Nevertheless, this result at least suggests a fundamental difficulty of ruling out non-interactive witness hiding protocols. Indeed, ruling out such protocols in the non-uniform setting would yield a surprising complexity implication, coming close to showing that the polynomial hierarchy collapses. Given that non-interactive witness hiding cannot be ruled out, we believe our result is also strongly suggestive that it should be possible to actually find a non-interactive witness hiding proof system, under plausible computational assumptions. Finding an explicit procedure for generating appropriate ψ clearly would suffice to make this scheme uniform; however, it is unclear how to do so."

In some sense, we accomplish this via proof complexity. A key insight is to shift focus from $\mathcal{D}$ and the falsifiable property to the specific proof system in which one can certify ψ is unsatisfiable.

3.6 Conclusion and Open Questions

There are many questions left open by this work. We discuss a few here.

- **What security properties cannot be captured?** We suspect that there are at least a few natural examples of security properties of zero-knowledge that are possible to achieve with zero interaction and that are not captured in our framework. However, we currently lack such a natural example (i.e., all the natural examples we thought of are either impossible or can be captured in our framework).

  One general (but somewhat unnatural) class of properties we seem unable to handle is security properties that only follow from, say, a uniform simulator and that are only falsifiable in, say, subexponential-time.

- **What if optimal proof systems do exist?** Our construction relies on the assumption that optimal (propositional) proof systems *do not* exist. Can one get interesting cryptographic consequences if optimal proof systems *do* exist? For example, Kuykendall and Zhandry [KZ20] show that "best-possible" $\mathcal{L}$-proofs exist, in the following sense: one can efficiently generate an $\mathcal{L}$-proof of "φ is satisfiable" that, very loosely speaking, is "most zero-knowledge" among all short $\mathcal{L}$-proofs. If one chooses $\mathcal{L}$ to be an optimal proof system, then can one get more power out of this?

  Also, we note that even if optimal proof systems do exist, the construction we use can still offer a human ignorance guarantee [Rog06]. To illustrate, we could choose ψ in the construction to be a

formula whose unsatisfiability corresponds to a mathematical statement that is believed to be true, but is not yet proven. For instance, consider the Collatz function ($n \in \mathbb{N}$ maps to $n/2$ if n is even and $\frac{3n+1}{2}$ otherwise) underlying the famous Collatz conjecture in mathematics. One quantitative version of the Collatz conjecture supported by empirical evidence and probabilistic models [LW92; KL10] is that for all $n \in \mathbb{N}$, the Collatz function repeatedly applied to n reaches one after at most $c \ln n$ iterations for some constant c ([LW92; KL10] suggest $c \approx 41.68$).

Based on this, one could choose ψ_λ such that it is unsatisfiable if and only if the quantitative Collatz conjecture is true on all λ-bit numbers for, say, $c = 100$. Then any attack on the corresponding prover constitutes a proof — modulo cryptographic assumptions — that ψ_λ is unsatisfiable and hence that the quantitative Collatz conjecture on λ-bit numbers is true. To our knowledge, such proofs are not known for, say, $\lambda = 128$. Hence, such a prover seems to offer a human ignorance guarantee. Intriguingly, even if no optimal proof systems exist, constructions like this might still offer more practical efficiency.

- **VBB and other "impossible" objects.** Can one build on our ideas to achieve other "impossible" definitions? For example, Virtual Black Box Obfuscation [BGI+12] (VBB) is a dream object in cryptography that is ruled out by an arguably contrived impossibility result. To avoid this impossibility result, cryptographers use a different definition, called indistinguishability obfuscation (iO) [BGI+12], that is significantly restricted.

  On the other hand, a common informal analogy is that NIWIs are to (truly) non-interactive zero-knowledge as iO is to VBB. The idea is that both non-interactive zero-knowledge and VBB are impossible simulation-based definitions, and NIWIs and iO are indistinguishability-based relaxations that can be achieved. In this work, we show how to use NIWIs to essentially achieve non-interactive zero-knowledge. Can similar ideas show that (if iO exists) we can essentially achieve VBB?

  For example, one concrete approach is examining what happens if we pick a family of unsatisfiable formulas φ with some "nice" properties and look at the construction that obfuscates a circuit C by outputting the iO of $C \vee \varphi$.

  Also, as discussed in the previous bullet point, there is a sense [GK16; KZ20] in which NIWIs and iO both give the "best possible" security guarantees. On the other hand, Proposition 7.10 and Theorem 2.1 together suggest that truly best possible proofs are impossible, at least if there is no optimal proof system. Is there an analogue of this for iO, or is this a fundamental difference between the two settings? Perhaps this points toward further examining obfuscators which lack perfect functionality.

- **Applications of zero-interaction zero-knowledge.** We expect that there are more applications of zero-interaction zero-knowledge. What else can one achieve?

  In an upcoming follow-up work we use zero-interaction zero-knowledge to refute the "scaled-down Rice's theorem" conjecture from [BGI+12].

- **Choosing some satisfiable ψ_λ?** In our construction, we choose ψ_λ to always be unsatisfiable in order to get perfect soundness. One could, alternatively, consider a sequence of formulas that is mostly unsatisfiable but is rarely satisfiable. In this case, one would lose perfect soundness on a small fraction of λ, but perhaps by carefully choosing ψ_λ one could plausibly be effectively zero-knowledge to *every* proof system. This object could be useful, for example, in applications related to impossibility results, where it suffices for an attacker to work on infinitely many input lengths. One could also hope for other weakenings of soundness, like uniform soundness [BP04].

  A good test case is understanding whether such ψ_λ exist in the random oracle model.

- **Either a dream world for cryptography or proof complexity.** In what follows we are informal, but we suspect that this can be made formal. Suppose a prover $\mathcal{P}$ is effectively zero-knowledge to $\mathcal{L}$ (think, $\mathcal{L} = \text{ZFC}$). Then we know that if a falsifiable property Π does not hold for $\mathcal{P}$, then it must be that "Π is a falsifiable security property of zero-knowledge" lacks a short proof in $\mathcal{L}$. Indeed, the adversary

falsifying Π constitutes a proof — in the proof system $\mathcal{L}' = (\mathcal{L}$ + "$\mathcal{P}$ is effectively zero-knowledge to $\mathcal{L}$") — that "$\mathcal{L}$ lacks a short proof that 'Π is a falsifiable security property of zero-knowledge.' "

Hence, one of two things is true for all such Π. Either Π holds for $\mathcal{P}$ or $\mathcal{L}'$ can prove that "$\mathcal{L}$ lacks a short proof that 'Π is a falsifiable property of zero-knowledge' " (in a certain quantitative sense).

One can interpret this as follows: either we live in a cryptographic dream world — $\mathcal{P}$ has most natural falsifiable properties of zero-knowledge — or we live in a proof complexity dream world — for every proof system $\mathcal{L}$, there is another proof system $\mathcal{L}'$ in which we can prove concrete lower bounds on the length of $\mathcal{L}$-proofs for many natural statements in complexity theory.

This phenomenon seems worthy of further investigation. Is there some sense in which the proof complexity dream world is unlikely?

4 Preliminaries

We now discuss a few preliminaries not already covered in Sections 2 and 3. We assume basic background in cryptography, as can be found in Goldreich's textbooks [Gol01; Gol04].

The size $|C|$ of a circuit C is the number of wires in the circuit, including input wires. When we say a problem requires circuits of size s, we mean it requires circuits of size at least s to compute.

We write $\{0,1\}^{\leq n}$ for the set of binary strings of length at most n. When choosing parameters, we often say things such as: set $\alpha = n^{\omega(1)}$ sufficiently small. This means choose α to be a sufficiently slow growing superpolynomial function.

4.1 Proof Systems

One can always close a proof system under implications provable in another proof system.

Proposition 4.1 (Folklore?)**.** *Let $\mathcal{L}$ and $\mathcal{L}'$ be proof systems. There is a proof system $\mathcal{L}^\star$ with both of the following properties:*

- simulation of $\mathcal{L}$: *For all X, if there is an ℓ-length $\mathcal{L}$-proof of X, then there is a $\mathsf{poly}(\ell)$-length $\mathcal{L}^\star$-proof of X.*
- polynomial closure under $\mathcal{L}'$-deduction: *For all X and Y, if there is an ℓ_0-length $\mathcal{L}^\star$-proof of X and there is an ℓ_1-length $\mathcal{L}'$-proof of "if X, then Y," then there is a $\mathsf{poly}(\ell_0, \ell_1)$-length $\mathcal{L}^\star$-proof of Y.*

Proof. We construct $\mathcal{L}^\star$ as follows.

Proof System $\mathcal{L}^\star$

Given a string which we interpret as a tuple (π, π') :

1. If π is the empty string, output $\mathcal{L}(\pi')$.
2. Otherwise, if $\mathcal{L}^\star(\pi) = X$ and $\mathcal{L}'(\pi')$ = "if X, then Y," then output Y.
3. Otherwise, output some fixed element of $\mathsf{P}^{\mathsf{HALT}}$.

Step (1) ensures that $\mathcal{L}^\star$ polynomially-simulates $\mathcal{L}$. Step (2) ensures $\mathcal{L}^\star$ is polynomially closed under $\mathcal{L}'$ deduction. Because $\mathcal{L}^\star$ makes at most one recursive call to a shorter input length, it is easy to see that $\mathcal{L}^\star$ runs in polynomial-time. □

4.2 Cryptography

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

Definition 4.2 (Computational Indistinguishability)**.** *First, we give a pointwise definition: Let C and D be multi-output circuits that just take as input randomness r, and let $\epsilon \in \mathbb{R}$. We let $C \approx_\epsilon D$ denote that C* and *D* are ϵ-computationally indistinguishable*. This means that for every adversary circuit A of size at most $1/\epsilon$ we have that*

$$\left|\Pr_r[A(C(r)) = 1] - \Pr_r[A(D(r)) = 1]\right| < \epsilon.$$

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

$$\left|\Pr_{x\leftarrow D_\lambda}[A(x) = 1] - \Pr_{x\leftarrow D'_\lambda}[A(x) = 1]\right| < \epsilon(\lambda).$$

Next, we recall the definition of a non-interactive witness indistinguishable proof system (NIWI) [FS90; BOV07]. We note that the definition below differs mildly from the usual definition in that we require that ϵ be efficiently computable.

Definition 4.3 (Non-Interactive Witness Indistinguishable Proof (NIWI))**.** *A* non-interactive witness indistinguishable proof system *is a tuple of uniform polynomial-time algorithms* $(\mathbf{NIWI.Prove}, \mathbf{NIWI.Verify})$ *where* **NIWI.Prove** *is randomized,* **NIWI.Verify** *is deterministic,*[46] *and all of the following hold:*

- Functionality: *For all formulas φ with $\varphi(w) = 1$ and all λ*

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

- Perfect Soundness: $\mathbf{NIWI.Verify}(\varphi, \pi, 1^\lambda) = 0$ *for all π, λ and unsatisfiable φ.*

- Security (Witness Indistinguishability): *There exists a polynomial-time computable function $\epsilon(\lambda) = \lambda^{-\omega(1)}$ such that for all formulas φ with $\varphi(w) = \varphi(w') = 1$, we have that*

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

  We say the NIWI is subexponentially secure *if the above holds for some polynomial-time computable*[47] *$\epsilon = 2^{-\lambda^{\Omega(1)}}$.*

When the underlying NIWI is clear from context, we say *output a NIWI proof of φ with witness w and security parameter 1^λ* to mean output $\mathbf{NIWI.Prove}(\varphi, w, 1^\lambda)$. Similarly, *a NIWI-proof of φ* (on security parameter λ) means a π satisfying $\mathbf{NIWI.Verify}(\varphi, \pi, 1^\lambda) = 1$.

5 Our Relaxation of Zero-Knowledge

We now (re)state our main definition.

[46] Assuming $\mathsf{P} = \mathsf{BPP}$, the verifier can always be made deterministic.

[47] Actually, this requirement is without loss of generality. Any $\epsilon = 2^{-\lambda^{\Omega(1)}}$ is upper bounded by a polynomial-time computable function $\epsilon' = 2^{-\lambda^{\Omega(1)}}$.

Definition 3.6 (Effectively Zero-Knowledge to $\mathcal{L}$)**.** *Let $\mathcal{P}$ be a prover, and let $\mathcal{L}$ be a proof system. We say $\mathcal{P}$ is* effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ *if for some $t = \lambda^{\omega(1)}$ (respectively, $t = 2^{\lambda^{\Omega(1)}}$) and $s = \mathsf{poly}(\lambda)$ we have that for all $\lambda \in \mathbb{N}$*

$$\text{“}\mathcal{P} \text{ has an } s(\lambda)\text{-size } \frac{1}{t(\lambda)}\text{-indistinguishable simulator on } \lambda\text{”}^{48} \tag{3}$$

is $t(\lambda)$-time indistinguishable from true to $\mathcal{L}$.

Remark 5.1. We clarify the precise meaning of (3) on a fixed λ to avoid any ambiguity arising from notation. Let $s^\star, t^\star \in \mathbb{N}$ be the concrete natural numbers with $s^\star = s(\lambda)$ and $t^\star = t(\lambda)$. Let P_λ be the concrete $\mathsf{poly}(\lambda)$-sized circuit that computes $\mathcal{P}.\mathsf{prove}(\cdot, \cdot, 1^\lambda)$ on inputs φ and w of size at most λ. Then (3) refers to the statement that

“there exists an $s^\star$-size circuit Sim such that $Sim(\varphi) \approx_{1/t^\star} P_\lambda(\varphi, w)$ whenever $|\varphi| \leq \lambda$ and $\varphi(w) = 1$.”

In particular, the statement does not include any further information about how to compute s, t or $\mathcal{P}$.

We also make the following definitions, which will be useful for extending our results. We advise the reader to skip these definitions on their first read through. The first definition lets us precisely quantify how non-uniform a simulator is.

Definition 5.2 (Non-Uniformity Quantified Simulator)**.** *Let $\lambda, \frac{1}{\epsilon}, s, \alpha \in \mathbb{N}$. We say $\mathcal{P}$ has an s-size α-non-uniform ϵ-indistinguishable simulator on λ if there is an $M \in \{0,1\}^{\leq \alpha}$ such that*[49] *$Sim_\lambda = U^s(M)$ is a probabilistic circuit satisfying $Sim_\lambda(\varphi) \approx_\epsilon \mathcal{P}.\mathsf{prove}(\varphi, w, 1^\lambda)$ for all φ with $\varphi(w) = 1$ and $|\varphi| \leq \lambda$.*

We can then define effectively zero-knowledge to $\mathcal{L}$ in a way that quantifies non-uniformity.

Definition 5.3 (α-Non-Uniform Effectively Zero-Knowledge to $\mathcal{L}$)**.** *Let $\mathcal{P}$ be a prover, and let $\mathcal{L}$ be a proof system. We say $\mathcal{P}$ is* α-non-uniform effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ *if for some $t = \lambda^{\omega(1)}$ (respectively, $t = 2^{\lambda^{\Omega(1)}}$) and $s = \mathsf{poly}(\lambda)$ and for all $\lambda \in \mathbb{N}$*

$$\text{“}\mathcal{P} \text{ has an } s(\lambda)\text{-size } \alpha(\lambda)\text{-non-uniform } \frac{1}{t(\lambda)}\text{-indistinguishable simulator on } \lambda,\text{”}$$

is $t(\lambda)$-time indistinguishable from true to $\mathcal{L}$.

Finally, we introduce another parameter to our definition.

Definition 5.4 (Effectively Zero-Knowledge to $\mathcal{L}$ with Hardness Ψ)**.** *Let $\mathcal{P}$ be a prover, and let $\mathcal{L}$ be a proof system. Let $\Psi = \{\psi_\lambda\}$ be a sequence of formulas of size at most λ. We say $\mathcal{P}$ is* effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ with hardness Ψ *if for some $\epsilon = \lambda^{-\omega(1)}$ (respectively, $\epsilon = 2^{-\lambda^{\Omega(1)}}$) and $s = \mathsf{poly}(\lambda)$ the following holds for all λ, M, t, and X: if there is an ℓ-length $\mathcal{L}$-proof that*

$$\text{“}\textit{if } \mathcal{P} \textit{ has an } s(\lambda)\textit{-size } \epsilon(\lambda)\textit{-indistinguishable simulator on } \lambda \textit{ and } X \textit{ is true, then } U^t(M) = 1,\text{”}$$

then either $U^t(M) = 1$ or there is a $\mathsf{poly}(t, \ell, \lambda)$-length $\mathcal{L}$-proof that “if X, then ψ_λ is unsatisfiable.”

6 Our Construction

Let $\Psi = \{\psi_\lambda\}$ be a sequence of λ-sized formulas (when we say λ-sized we mean size at most λ). We now state the construction of our prover, which is parameterized by a choice of Ψ and an implicit choice of NIWI that we fix for the remainder of this paper. The prover will be uniform when the sequence Ψ is P-uniform (i.e., there is a deterministic polynomial-time algorithm that, given 1^λ, outputs ψ_λ).

[48] In Remark 5.1, we clarify potential ambiguities arising from notation in (3).

[49] We are essentially saying that Sim_λ has low time-bounded Kolmogorov complexity.

Prover $\mathcal{P}[\Psi]$

$\mathcal{P}[\Psi].\mathsf{prove}$ on input $(\varphi, w, 1^\lambda)$:

1. Reject if $|\varphi| > \lambda$. Also reject if $\varphi(w) = \psi_\lambda(w) = 0$.
2. Output a NIWI proof of "either φ or ψ_λ is satisfiable" with witness w and security parameter λ.

$\mathcal{P}[\Psi].\mathsf{verify}$ on input $(\varphi, \pi, 1^\lambda)$:

1. Accept if π is a valid NIWI proof of "either φ or ψ_λ is satisfiable."

It is easy to see that this prover is perfectly sound if every ψ_λ is unsatisfiable.

Lemma 6.1 (Perfect Soundness of $\mathcal{P}[\Psi]$)**.** *If ψ_λ is unsatisfiable for all λ, then $\mathcal{P}[\Psi]$ is perfectly sound.*

Proof. If $\mathcal{P}[\Psi].\mathsf{verify}(\varphi, \pi, 1^\lambda)$ accepts, then (by the perfect soundness of the NIWI) we have that either φ is satisfiable or ψ_λ is satisfiable. Since ψ_λ is unsatisfiable, this means that φ is satisfiable. □

6.1 Analysis

In our analysis, it will be helpful to isolate the circuit corresponding to our prover on 1^λ inputs. In particular, consider the following probabilistic circuit, which is parameterized by a $\lambda \in \mathbb{N}$ and a single formula ψ of size at most λ.

Probabilistic Circuit $\mathcal{P}[\psi, \lambda]$

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

1. If $\varphi(w) = 0$ and $\psi(w) = 0$, then reject.
2. Output a NIWI proof of "either φ or ψ is satisfiable" with witness w and security parameter λ.

We can naturally extend the definition of a simulator to $\mathcal{P}[\psi, \lambda]$. That is, for $s, \frac{1}{\epsilon} \in \mathbb{N}$, we say that $\mathcal{P}[\psi, \lambda]$ has an *s-size ϵ-indistinguishable simulator* if there exists an s-size circuit Sim_λ such that, for all φ of size at most λ with $\varphi(w) = 1$,

$$Sim_\lambda(\varphi) \approx_\epsilon \mathcal{P}[\psi, \lambda](\varphi, w).$$

We analogously define $\mathcal{P}[\psi, \lambda]$ having an *s-size α-non-uniform ϵ-indistinguishable simulator*.

Note that the definition of "$\mathcal{P}[\Psi]$ has an s-size ϵ-indistinguishable simulator on λ" is exactly that "$\mathcal{P}[\psi_\lambda, \lambda]$ has an s-size ϵ-indistinguishable simulator."

We now prove the key lemma regarding this construction, which builds on the ideas of Kuykendall and Zhandry [KZ20]. The lemma roughly says that either (a strengthening of) $\mathcal{L}$ has short proofs of unsatisfiability for Ψ, or $\mathcal{P}[\Psi]$ has all the "time-bounded" consequences of zero-knowledge that are provable in $\mathcal{L}$. Properties (2) and (3) below will only be relevant for extensions of our main result; we recommend the reader ignore them on their first read through.

Lemma 6.2. *Assume a NIWI (respectively, subexponentially secure NIWI) exists. There exist functions $\epsilon(\lambda) = \lambda^{-\omega(1)}$ (respectively $\epsilon(\lambda) = 2^{-\lambda^{\Omega(1)}}$) and $s(\lambda) = \mathsf{poly}(\lambda)$ such that the following three statements hold for every proof system $\mathcal{L}$, every $\lambda, t \in \mathbb{N}$, every statement X, and every formula ψ of size at most λ.*

1. *If there is an ℓ-length $\mathcal{L}$-proof of the statement*

   "if $\mathcal{P}[\psi, \lambda]$ has an $s(\lambda)$-size $\epsilon(\lambda)$-indistinguishable simulator, then $U^t(M) = 1$,"

   then either:

- $U^t(M) = 1$, *or*
- *in the proof system* $\mathcal{L}_{extended}$ *(which is defined below and depends only on* $\mathcal{L}$ *and the choice of NIWI), there is a* $\mathsf{poly}(\ell, \lambda, t)$*-length proof that "*$\psi$ *is unsatisfiable."*

2. *If there is an* ℓ*-length* $\mathcal{L}$*-proof of the statement*

   "if $\mathcal{P}[\psi, \lambda]$ *has an* $s(\lambda)$*-size* $s(|\psi|)$*-non-uniform* $\epsilon(\lambda)$*-indistinguishable simulator, then* $U^t(M) = 1$*,"*

   then either:

   - $U^t(M) = 1$, *or*
   - *in the proof system* $\mathcal{L}_{extended}$ *there is a* $\mathsf{poly}(\ell, \lambda, t)$*-length proof that "*$\psi$ *is unsatisfiable."*

3. *If there is an* ℓ*-length* $\mathcal{L}$*-proof of the statement*

   "if $\mathcal{P}[\psi, \lambda]$ *has an* $s(\lambda)$*-size* $\epsilon(\lambda)$*-indistinguishable simulator and* X *is true, then* $U^t(M) = 1$*,"*

   then either:

   - $U^t(M) = 1$, *or*
   - *there is a* $\mathsf{poly}(\ell, \lambda, t)$*-length* $\mathcal{L}_{extended}$*-proof that "if* X*, then* ψ *is unsatisfiable."*

Proof. Let $\epsilon(\lambda)$ be the polynomial-time computable function corresponding to the security of the NIWI. Let $s(\lambda) = \mathsf{poly}(\lambda)$ be a sufficiently large polynomial satisfying $s(\lambda) \geq |\mathcal{P}[\psi', \lambda]|$ for every ψ' of size at most λ.

Now fix any proof system $\mathcal{L}$. Let $\mathcal{L}_{extended}$ be the proof system defined as follows.

Proof System $\mathcal{L}_{extended}$

On input $\pi, 1^\lambda, 1^t$, and $\alpha \in [4]$:

1. Output "ψ is unsatisfiable" if $\alpha = 1$ and $|\psi| \leq \lambda$ and
   - $\mathcal{L}(\pi)$ = "if $\mathcal{P}[\psi, \lambda]$ has an $s(\lambda)$-size $\epsilon(\lambda)$-indistinguishable[a] simulator, then $U^t(M) = 1$"[b] and
   - $U^t(M) \neq 1$ (we check this by just running it).
2. Output "ψ is unsatisfiable" if $\alpha = 2$ and $|\psi| \leq \lambda$ and
   - $\mathcal{L}(\pi)$ = "if $\mathcal{P}[\psi, \lambda]$ has $s(\lambda)$-size $s(|\psi|)$-non-uniform $\epsilon(\lambda)$-indistinguishable simulator, then $U^t(M) = 1$."
   - $U^t(M) \neq 1$.
3. Output "if X, then ψ is unsatisfiable" if $\alpha = 3$ and $|\psi| \leq \lambda$ and
   - $\mathcal{L}(\pi)$ = "if $\mathcal{P}[\psi, \lambda]$ has an $s(\lambda)$-size $\epsilon(\lambda)$-indistinguishable simulator and X is true, then $U^t(M) = 1$," and
   - $U^t(M) \neq 1$.
4. Otherwise, output $\mathcal{L}(\pi)$

[a] This is where we need that ϵ is efficiently computable.
[b] We choose a reasonable encoding such that this step is efficient.

By construction, $\mathcal{L}_{extended}$ runs in polynomial time. Moreover, if $\mathcal{L}_{extended}$ is indeed a proof system, the lemma follows immediately by construction of $\mathcal{L}_{extended}$. It remains to show the following claim.

Claim 6.3. $\mathcal{L}_{extended}$ *is a proof system.*

Proof. We need to show that if $\mathcal{L}_{extended}$ outputs a statement, then that statement is true. We divide into cases depending on whether it terminates at step (1), (2), (3), or (4). If it terminates at step (4), then we are done because $\mathcal{L}$ is a proof system.

Next, suppose that it terminates at step (3). By the soundness of $\mathcal{L}$, we know that if "X" is true, then $\mathcal{P}[\psi, \lambda]$ does not have an $s(\lambda)$-size $\epsilon(\lambda)$-indistinguishable simulator. On the other hand, if it was the case that $\psi(w^\star) = 1$ for some $w^\star$, then NIWI security guarantees that the probabilistic circuit

$$Sim_\lambda(\varphi) = \mathcal{P}[\psi, \lambda](\varphi, w^\star)$$

is an $s(\lambda)$-sized $\epsilon(\lambda)$-indistinguishable simulator for $\mathcal{P}[\psi, \lambda]$. Hence, ψ is unsatisfiable if "X" is true. So the statement output at step (3) is true.

The argument for termination at step (1) is similar to the argument for step (3). Finally, suppose it terminates at step (2). The argument is again similar to the one for step (3). However, note that the potential simulator

$$Sim_\lambda(\varphi) = \mathcal{P}[\psi, \lambda](\varphi, w)$$

is determined by a choice of ψ and w and the constant size code of the NIWI. All this can be encoded by strings of length at most $\mathsf{poly}(|\psi|)$. Hence by setting the polynomial s sufficiently large, we get that $U^{s(\lambda)}(M) = Sim_\lambda$ for some M of length at most $s(|\psi|)$. □

□

Motivated by Lemma 6.2, we define a notion of Ψ being hard for $\mathcal{L}$.

Definition 6.4 (Ψ is hard for $\mathcal{L}$)**.** *Let $\Psi = \{\psi_\lambda\}$ be a sequence of λ-sized formulas. Let $\mathcal{L}$ be a proof system. We say Ψ is* hard (respectively, subexponentially hard) for $\mathcal{L}$ *if there is an $\ell = \lambda^{\omega(1)}$ (respectively $\ell = 2^{\lambda^{\Omega(1)}}$) such that for all λ there is no $\ell(\lambda)$-length $\mathcal{L}_{extended}$-proof that "ψ_λ is unsatisfiable." We stress that the previous sentence talks about $\mathcal{L}_{extended}$-proofs.*

Note that this definition depends on the definition of $\mathcal{L}_{extended}$, which depends on our choice of NIWI. Also note that this definition does not require that the formulas ψ_λ actually be unsatisfiable. For example, any sequence of satisfiable formulas is hard for $\mathcal{L}$.

Combining this definition with Lemma 6.2, we get the following theorem. It roughly says that, if Ψ is hard for $\mathcal{L}$, then $\mathcal{P}[\Psi]$ is effectively zero-knowledge to $\mathcal{L}$.

Theorem 6.5. *Assume Ψ is* hard (respectively, subexponentially hard) for $\mathcal{L}$*. Then $\mathcal{P}[\Psi]$ is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$. Furthermore, $\mathcal{P}[\Psi]$ is $\mathsf{poly}(|\psi_\lambda|)$-non-uniform effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$.*

Proof. Let $\epsilon = \lambda^{-\omega(1)}$ (respectively, $\epsilon = 2^{-\lambda^{\Omega(1)}}$) and $s = \mathsf{poly}(\lambda)$ be the functions given by Lemma 6.2. Let $\alpha(\lambda) = s(|\psi_\lambda|)$. Let $t = t(\lambda)$ be a function we set later. Let $\mathcal{P} = \mathcal{P}[\Psi]$.

Suppose there is a $t(\lambda)$-length $\mathcal{L}$-proof that

$$\text{"if } \mathcal{P} \text{ has an } s(\lambda)\text{-size } \epsilon(\lambda)\text{-indistinguishable simulator on } \lambda, \text{ then } U^t(M) = 1\text{"}.$$

Then, by Lemma 6.2, either $U^t(M) = 1$ or there is a $\mathsf{poly}(t(\lambda))$-length $\mathcal{L}_{extended}$-proof that "ψ_λ is unsatisfiable." By assumption, any $\mathcal{L}_{extended}$-proof that "ψ_λ is unsatisfiable" has length at least $\lambda^{\omega(1)} > \mathsf{poly}(t(\lambda))$ (respectively, $2^{\lambda^{\Omega(1)}} > \mathsf{poly}(t(\lambda))$) by setting $t = \lambda^{\omega(1)}$ (respectively $t = 2^{\lambda^{\Omega(1)}}$) sufficiently small. Hence, $U^t(M) = 1$. This shows that $\mathcal{P}[\Psi]$ is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$.

The proof that $\mathcal{P}[\Psi]$ is α-non-uniform effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ is similar. □

We also prove the following proposition, which is useful for an extension of our main result.

Proposition 6.6. *For every $\mathcal{L}$, we have that $\mathcal{P}[\Psi]$ is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}_{extended}$ with hardness Ψ.*

Proof. Let $\epsilon = \lambda^{-\omega(1)}$ (respectively, $\epsilon = 2^{-\lambda^{\Omega(1)}}$), and $s = \lambda^{O(1)}$ be the parameters given by Lemma 6.2. Let $\mathcal{P} = \mathcal{P}[\Psi]$. Suppose there is an ℓ-length $\mathcal{L}$-proof that

"if $\mathcal{P}$ has an $s(\lambda)$-size $\epsilon(\lambda)$-indistinguishable simulator on λ and X is true, then $U^t(M) = 1$".

Then by Lemma 6.2, either $U^t(M) = 1$ or there is a $\mathsf{poly}(\ell, \lambda, t)$-length $\mathcal{L}_{extended}$-proof that "if X, then ψ_λ is unsatisfiable," as desired. □

6.2 Optimal Proof Systems

We begin by recalling the definition of an optimal propositional proof system.

Definition 6.7 (Optimal Propositional Proof System [KP89])**.** *A propositional proof system $\mathcal{L}$ is* optimal (respectively, subexponentially optimal) *if the following holds. For every propositional proof system $\mathcal{L}'$, there exists a $p(\ell) = \mathsf{poly}(\ell)$ (respectively, $p(\ell) = 2^{\ell^{o(1)}}$) such that for all ψ and all ℓ', we have that if there is an ℓ'-length $\mathcal{L}'$-proof that "ψ is unsatisfiable," then there is a $p(\ell')$-length $\mathcal{L}$-proof that "ψ is unsatisfiable."*

We say that $\mathcal{L}$ is an infinitely often optimal (respectively, subexponentially optimal) propositional proof system *if the analogous statement holds but for infinitely many ℓ'.*

It is conjectured that there is no optimal propositional proof system.

Conjecture 6.8 (No Optimal Proof System [KP89])**.** *There is no optimal propositional proof system.*

It is natural to extend this conjecture to the infinitely often setting.

Conjecture 6.9. *There is no infinitely often optimal propositional proof system.*

Using this conjecture, we can get P-uniform Ψ that are hard for an arbitrary $\mathcal{L}$. The proof is a slight modification of a result by Krajíček and Pudlák [KP89].

Theorem 6.10. *Assume there is no infinitely often optimal (respectively, subexponentially optimal) propositional proof system. Then for every proof system $\mathcal{L}$, there is a P-uniform sequence $\Psi = \{\psi_\lambda\}$ of λ-sized unsatisfiable formulas that are hard (respectively, subexponentially hard) for $\mathcal{L}$.*[50]

Proof. We prove the contrapositive. Assume $\mathcal{L}$ is a proof system for which there is no uniform sequence of unsatisfiable formulas that is hard (respectively, subexponentially hard) for $\mathcal{L}$. We will construct an infinitely often optimal (respectively, subexponentially optimal) propositional proof system. Consider the propositional proof system $\mathcal{L}^\star$ defined as follows.

$\mathcal{L}^\star(\pi, \pi',$ circuit $C)$:
1. Output "ψ is unsatisfiable" if $C(\pi') = \psi$ and $\mathcal{L}_{extended}(\pi) =$ "C only outputs unsatisfiable formulas."[a]
2. Otherwise, output the trivially unsatisfiable formula $x_1 \wedge \neg x_1$.

[a] Note that "C only outputs unsatisfiable formulas" is a coNP statement: to witness it is false, provide an x and w with $C(x) = \varphi$ and $\varphi(w) = 1$. Hence, it also encodable as an instance of UNSAT.

We now show $\mathcal{L}^\star$ is an infinitely often optimal (respectively, subexponentially optimal) propositional proof system. Let $\mathcal{L}'$ be an arbitrary propositional proof system, and let C'_λ be the corresponding circuit that computes $\mathcal{L}'$ on inputs of length up to λ. Then let ψ_λ be the formula of size $\mathsf{poly}(\lambda)$ that is unsatisfiable if and only if C'_λ only outputs unsatisfiable formulas.

[50] The definition of hard for $\mathcal{L}$ involves $\mathcal{L}_{extended}$. In turn, the definition of $\mathcal{L}_{extended}$ depends on an underlying NIWI. Thus, implicitly this theorem statement says that for all choices of NIWIs, this statement holds.

Since $\mathcal{L}'$ is sound, $\{\psi_\lambda\}$ is a P-uniform sequence of polynomial-size unsatisfiable formulas. To make these formulas have size at most λ, let ψ'_λ be equal to the ψ_i where $i \in [\lambda]$ is the largest number with $|\psi_i| \leq \lambda$ (if there is no such i, output a trivially unsatisfiable formula). Note that we have $i = \lambda^{\Omega(1)}$ since $|\psi_\lambda| = \mathsf{poly}(\lambda)$.

Then by (the contrapositive) assumption, there are $\mathsf{poly}(\lambda)$-length (respectively, $2^{\lambda^{o(1)}}$-length) $\mathcal{L}_{extended}$-proofs that "ψ'_λ is unsatisfiable" for infinitely many λ. Hence, there are $\mathsf{poly}(\lambda)$-length (respectively, $2^{\lambda^{o(1)}}$-length) $\mathcal{L}_{extended}$-proofs that "ψ_λ is unsatisfiable" for infinitely many λ. Then, by the definition of $\mathcal{L}^\star$, it follows that λ-length proofs in $\mathcal{L}'$ have analogous $\mathsf{poly}(\lambda)$-length (respectively, $2^{\lambda^{o(1)}}$-length) proofs in $\mathcal{L}^\star$ for infinitely many λ. □

6.3 Main Result

We can now prove our main result.

Theorem 3.1 (Main Result)**.** *Assume*

- *NIWIs (respectively, subexponentially-secure NIWIs) exist, and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

Then for every proof system $\mathcal{L}$, there exists a perfectly sound prover $\mathcal{P}$ that is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$.

Proof. Fix a proof system $\mathcal{L}$. By Theorem 6.10, there exists a P-uniform sequence Ψ of λ-sized unsatisfiable formulas that are hard (respectively, subexponentially hard) for $\mathcal{L}$. Set $\mathcal{P} = \mathcal{P}[\Psi]$. By Theorem 6.5, we get that $\mathcal{P}$ is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$. By Lemma 6.1, $\mathcal{P}$ is perfectly sound. □

7 Extensions

7.1 Effectively Zero-Knowledge to Every Proof System?

One might wonder whether there is a perfectly sound prover that is effectively zero-knowledge to *every* proof system. This is unlikely, essentially because of the impossibility results of Goldreich and Oren [GO94].

Proposition 7.1. *Assume* SAT *does not have circuits of size $n^{O(\log^\star n)}$ infinitely often.*[51] *Then no perfectly sound prover $\mathcal{P}$ is effectively zero-knowledge to every proof system.*

Proof. For contradiction, suppose this is not the case for $\mathcal{P}$. Let q be a sufficiently large polynomial. We will show the following two claims.

Claim 7.2. *$\mathcal{P}$ has a $\lambda^{\log^\star \lambda}$-size $\frac{1}{q(\lambda)}$-indistinguishable simulator on λ for infinitely many λ.*

Claim 7.3. *Let $\lambda \in \mathbb{N}$. If $\mathcal{P}$ has a $\lambda^{\log^\star \lambda}$-size $\frac{1}{q(\lambda)}$-indistinguishable simulator Sim_λ on λ, then there is a circuit of size $\lambda^{O(\log^\star \lambda)}$ solving SAT on all formulas of size at most λ.*

The proposition follows immediately from combining the two claims. We now prove the claims.

Proof of Claim 7.2. For contradiction, suppose not. Then there is a constant $\lambda_0 \in \mathbb{N}$ such that, for all $\lambda \geq \lambda_0$, there is no $\lambda^{\log^\star \lambda}$-size $\frac{1}{q(\lambda)}$-indistinguishable simulator for $\mathcal{P}$ on λ. Thus, we can construct a proof system $\mathcal{L}$ with $\mathsf{poly}(\log \frac{s\lambda t}{\epsilon})$-length proofs of all statements of the form

$$\text{"if } \mathcal{P} \text{ has an } s\text{-size } \epsilon\text{-indistinguishable simulator on } \lambda, \text{ then } U^t(M) = 1\text{"}$$

where $\lambda \geq \lambda_0$ and $\epsilon \leq 1/q(\lambda)$ and $s \leq \lambda^{\log^\star \lambda}$ and where M is a fixed string we choose satisfying $U^t(M) = 0$ for all t. Since $\mathcal{P}$ is effectively zero-knowledge to $\mathcal{L}$, it follows that $U^t(M) = 1$ for some t, which is a contradiction. □

[51] Here n denotes the actual input length to SAT, not the number of inputs the formula has.

Proof of Claim 7.3. Consider the following randomized circuit.

> On an input φ of size at most λ:
>
> 1. Accept φ if and only if $Sim_\lambda(\varphi)$ outputs a proof of "φ is satisfiable" that $\mathcal{P}.\mathsf{verify}$ accepts.

By perfect soundness, this circuit rejects all NO instances of SAT. Setting q to be sufficiently large, the security of Sim_λ implies the circuit accepts every YES instance of SAT with probability at least $2/3$. Thus, there is a probabilistic circuit of size $\lambda^{O(\log^\star \lambda)}$ that solves SAT on instances of size λ.

This can be converted into a deterministic circuit of size $\lambda^{O(\log^\star \lambda)}$ using Adleman's trick (i.e., make the failure probability exponentially small by taking the majority of polynomially many independent trials and then use non-uniformity to choose random bits that work for all inputs). □

□

Perhaps surprisingly, one can avoid the above impossibility result by considering *non-uniform* provers. To do so, consider the following definition.

Definition 7.4 (Distribution Hard for All Propositional Proof Systems)**.** *Let $\mathcal{D}_\lambda$ be a polynomial-time samplable distribution on λ-sized formulas. We say that $\mathcal{D}_\lambda$ is a* distribution hard for all propositional proof systems *if for all proof systems $\mathcal{L}$ there is an $\ell(\lambda) = \lambda^{\omega(1)}$ such that*

$$\Pr_{\psi \leftarrow \mathcal{D}_\lambda}[\textit{either } \psi \textit{ is satisfiable or there is an } \ell(\lambda)\textit{-length } \mathcal{L}\textit{-proof that "}\psi \textit{ is unsatisfiable"}] = o(\frac{1}{\lambda^2}).$$

One candidate for such a distribution comes from Rudich's conjectured extension of the natural proofs barrier [Rud97]. Specifically, the candidate distribution is: sample a uniformly random truth table T and output a $\mathsf{poly}(|T|)$-sized formula ψ that is satisfiable if and only if T is computable by an unexpectedly small (e.g. $\frac{|T|}{10 \log |T|}$) circuit.

Next, we show that sampling from a distribution hard for all propositional proof systems leads (with high probability) to a sequence of unsatisfiable formulas hard for every proof system $\mathcal{L}$.

Proposition 7.5. *Assume there is a distribution $\mathcal{D}_\lambda$ hard for all propositional proof systems. Then there is a non-uniform sequence $\Psi = \{\psi_\lambda\}$ of λ-sized unsatisfiable formulas that is hard for every proof system $\mathcal{L}$.*

Proof. We will do this by a probabilistic argument. In particular, we will set $\psi_\lambda \leftarrow \mathcal{D}_\lambda$ and show that, with positive probability, it has the desired properties.

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

By assumption, we have that for every proof system $\mathcal{L}$ and every polynomial ℓ

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

The set consisting of all pairs $(\mathcal{L}, \ell)$ is countable. So, by Proposition 7.9 below (essentially the Borel-Cantelli lemma), we get that

$$\Pr[\text{for every } \mathcal{L} \text{ and every } \ell\text{, only finitely many } E_{\mathcal{L},\ell,\lambda} \text{ occur}] = 1,$$

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

This leads to a non-uniform prover that is effectively zero-knowledge to every proof system.

Theorem 7.6. *Assume NIWIs exist and there is a universally hard distribution $\mathcal{D}_\lambda$ for UNSAT. Then there exists a non-uniform prover that is effectively zero-knowledge to every proof system.*

Proof. By Proposition 7.5, we get a non-uniform sequence $\Psi = \{\psi_\lambda\}$ of λ-sized unsatisfiable formulas hard for every proof system. Our non-uniform prover is $\mathcal{P}[\Psi]$. By Theorem 6.5, we have that $\mathcal{P}[\Psi]$ is effectively zero-knowledge to every proof system. By Lemma 6.1, we have that $\mathcal{P}[\Psi]$ is perfectly sound. □

We note that the non-uniformity in Theorem 7.6 is essentially just a uniformly random string (to sample from $\mathcal{D}_\lambda$). Thus, one could also view Theorem 7.6 as a version of a NIZK (non-interactive zero-knowledge with trusted setup) that even has security against adversaries that non-uniformly depend on the common random string.

It remains to prove the aforementioned consequence of the Borel-Cantelli lemma.

Lemma 7.7 (Borel-Cantelli Lemma)**.** *Let $\{E_n\}_{n\in\mathbb{N}}$ be a collection of events. Assume $\sum_{n\in\mathbb{N}} \Pr[E_n]$ is finite. Then, with probability one, only a finite number of events E_n occur.*

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

Proposition 7.8. *Let $\{E_{m,n}\}_{m,n\in\mathbb{N}}$ be a collection of events. Assume that $\sum_{n\in\mathbb{N}} \sum_{m<g(n)} \Pr[E_{m,n}]$ is finite for some function $g(n) = \omega(1)$. Then*

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

Proof. Let G_n be the event that $E_{m,n}$ occurs for some $m \leq g(n)$. By the Borel-Cantelli lemma, with probability one, only finitely many G_n occur. On the other hand, if for some m, infinitely many $E_{m,n}$ occur, then infinitely many G_n occur (using that $g = \omega(1)$). The proposition follows. □

In particular, one setting of parameters gives the following.

Proposition 7.9. *Let $\{E_{m,n}\}_{m,n\in\mathbb{N}}$ be a collection of events. Assume that for all m, there exists an integer n_m such that*

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

for all $n \geq n_m$. Then

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

Proof. Define $g : \mathbb{N} \to \mathbb{N}$ by $g(n) = \max\{m \leq \sqrt{n} : n > n_{m'} \text{ for all } m' < m\}$. Observe that $g = \omega(1)$. Then we have that

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

is finite. Then the result follows from Proposition 7.8. □

7.2 Falsifiable Properties of Zero-Knowledge

We now prove the weak version of our main result from Section 2.

Theorem 2.1 (Weak Version of Main Result)**.** *Assume*

- $\mathsf{P} = \mathsf{BPP}$,
- *NIWIs exist (respectively, subexponentially secure NIWIs exist), and*
- *there is no infinitely often optimal (respectively, subexponentially optimal) proof system.*

For every falsifiable property of zero-knowledge Π, there exists a perfectly sound prover for which Π holds (respectively, holds with subexponential security).

Proof. Since $\Pi = (G, \Delta)$ is a falsifiable property of zero-knowledge, we have that

$$\mathsf{Estimate}(G, (\mathcal{P}, A, 1^\lambda), 1^\tau) \leq \frac{1}{\tau} + \Delta(\lambda, A, s, \epsilon) \tag{4}$$

whenever $\mathcal{P}$ has an s-size ϵ-indistinguishable simulator on λ.

Since Δ is computable in polynomial time, we can construct a proof system $\mathcal{L}$ such that

"If $\mathcal{P}$ has an s-size ϵ-indistinguishable simulator on λ, then (4) holds"

always has a proof of length $\mathsf{poly}(|\mathcal{P}|, |A|, \log \frac{s\lambda\tau}{\epsilon})$. By Theorem 3.1, there is a perfectly sound prover $\mathcal{P}^\star$ that is effectively zero-knowledge (respectively, subexponentially zero-knowledge) to $\mathcal{L}$ with associated parameters $s^\star = \mathsf{poly}(\lambda)$ and $t^\star = \lambda^{\omega(1)}$ (respectively, $t^\star = 2^{\lambda^{\Omega(1)}}$) and $\epsilon^\star = 1/t^\star$.

Effectively zero-knowledge to $\mathcal{L}$ implies that for some $\tau^\star = \lambda^{\omega(1)}$ (respectively, $\tau^\star = 2^{\lambda^{\Omega(1)}}$)

$$\mathsf{Estimate}(G, (\mathcal{P}^\star, A, 1^\lambda), 1^{\tau^\star}) \leq \frac{1}{\tau^\star} + \Delta(\lambda, A, s^\star, \epsilon^\star).$$

Hence, by the correctness of $\mathsf{Estimate}$,

$$\Pr_G[G(\mathcal{P}^\star, A, 1^\lambda) \text{ outputs 1 in } \tau^\star \text{ time}] \leq \frac{2}{\tau^\star} + \Delta(\lambda, A, s^\star, \epsilon^\star).$$

The theorem follows, by redefining $\tau^\star = \min\{\frac{\tau^\star}{2}, \frac{1}{\epsilon}\}$ and then redefining $\epsilon^\star = \frac{1}{\tau^\star}$. □

We also show that it is unlikely that a single perfectly sound prover has every falsifiable property of zero-knowledge. The proof is essentially Barak, Ong, and Vadhan's attack [BOV07] that "the verifier gains the ability to prove the same statement to others."

Proposition 7.10 (No Universal Prover for Falsifiable Properties)**.** *Assume there is a cryptographic pseudorandom generator PRG that is $n^{-\Omega(\log^\star n)}$-indistinguishable from random. Then for every perfectly sound prover $\mathcal{P}^\star$, there is a falsifiable property of zero-knowledge that does not hold on $\mathcal{P}^\star$.*

Proof. Set $n = \lambda^{\Omega(1)}$ sufficiently small. Consider the following game G.

$G(\mathcal{P}, A, 1^\lambda)$:

1. Let $x \leftarrow \{0,1\}^n$ and $y = PRG(x)$ and let φ_y be the SAT instance corresponding to "y is in the range of PRG on n-length inputs." Set n sufficiently small so that $|\varphi_y| \leq \lambda$.
2. Let $\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi_y, x, 1^\lambda)$. Let $\pi^\star = A(y, \pi)$.
3. Accept if and only if $\mathcal{P}^\star.\mathsf{verify}(\varphi_y, \pi^\star, 1^\lambda) = 1$. We stress that this is the verifier for $\mathcal{P}^\star$, not $\mathcal{P}$.

Now we show that $\Pi = (G, \Delta)$ is a falsifiable property of zero-knowledge for a suitable Δ. Fix a λ. Let $\mathcal{P}$ be a prover with an s-size ϵ-indistinguishable simulator Sim_λ on λ, and let A be an adversary circuit. Consider the circuit A' given by

$$A'(y) = \mathcal{P}^\star.\mathsf{verify}(\varphi_y, A(y, Sim_\lambda(\varphi_y)), 1^\lambda).$$

Because $\mathcal{P}^\star$ is perfectly sound, the security of the PRG implies that

$$\Pr_y[A'(y) = 1] \leq \lambda^{-\Omega(\log^\star \lambda)} \cdot \mathsf{poly}(|A|, \lambda).$$

Then by the security of Sim_λ and the construction of G, we have that

$$\Pr[G(1^\lambda, \mathcal{P}, A) = 1] \leq \left(\lambda^{-\Omega(\log^\star \lambda)} + \epsilon\right) \cdot \mathsf{poly}(|A|, \lambda, s).$$

Thus, $\Pi = (G, \Delta)$ is a falsifiable property of zero-knowledge for some $\Delta = \left(\lambda^{-\Omega(\log^\star \lambda)} + \epsilon\right) \cdot \mathsf{poly}(|A|, \lambda, s)$.

On the other hand, let $A^\star$ be the $\mathsf{poly}(\lambda)$-sized adversary given by $A^\star(x, \pi) = \pi$. By construction, for every λ we have that $G(\mathcal{P}^\star, A^\star, 1^\lambda)$ outputs 1 in polynomial time. Hence, Π does not hold for $\mathcal{P}^\star$.

□

7.3 Ultimate Provers

Let Natural be a collection of falsifiable properties of zero-knowledge. Our results in this subsection will be parameterized by a choice of Natural. For simplicity in this subsection, we only consider security against polynomial-time adversaries.

First, we define what it means for a proof system to prove every falsifiable property of zero-knowledge in Natural.

Definition 7.11. *We say a proof system $\mathcal{L}$* proves all of Natural *if for every $\Pi \in$ Natural, there exists an $\mathcal{L}$-proof (of any length) of "Π is a falsifiable property of zero-knowledge."*

A natural candidate for such a proof system is ZFC. If indeed such a proof system exists, then we get an ultimate prover.

Theorem 7.12. *Assume $\mathsf{P} = \mathsf{BPP}$, NIWIs exist, a proof system $\mathcal{L}$ proves all of* Natural*, and there is a P-uniform sequence Ψ of λ-sized unsatisfiable formulas that is hard for $\mathcal{L}$. Then there exists a perfectly sound prover $\mathcal{P}$ such that every $\Pi \in$ Natural holds on $\mathcal{P}$.*

Proof. By Theorem 6.5 and Lemma 6.1, there is a perfectly sound prover $\mathcal{P}^\star$ that is effectively zero-knowledge to $\mathcal{L}$ with associated parameters $s^\star = \mathsf{poly}(\lambda)$ and $t^\star = \lambda^{\omega(1)}$ and $\epsilon^\star = \frac{1}{t^\star}$.

Fix a $\Pi = (G, \Delta) \in$ Natural. Since $\mathcal{L}$ proves all of Natural and is (without loss of generality) sufficiently strong, there is always a $\mathsf{poly}(|A|, |\mathcal{P}|, \log \frac{s\lambda\tau}{\epsilon})$-length proof that

$$\text{"if } \mathcal{P} \text{ has an } s\text{-size } \epsilon\text{-indistinguishable simulator on } \lambda, \text{ then } \mathsf{Estimate}(G, (\mathcal{P}, A, 1^\lambda), 1^\tau) \leq \frac{1}{\tau} + \Delta(\lambda, A, s, \epsilon)".$$

The proof is then the same as the proof of Theorem 2.1 in Section 7.2. □

As discussed in Section 2.5, we can also consider a different hypothesis to get an ultimate prover. We state this for ZFC for simplicity, but one could consider other proof systems.

Definition 7.13. *We say there is a hard* UNSAT *sequence for* Natural *if all of the following are true.*

- *For some choice of NIWI, ZFC simulates $ZFC_{extended}$.*
- *ZFC proves the correctness of a polynomial-time deterministic* Estimate *algorithm.*
- *There is a P-uniform sequence $\Psi = \{\psi_\lambda\}$ of unsatisfiable formulas with the following property. For every $\Pi \in$ Natural, there exists an $\ell = \lambda^{\omega(1)}$ such that any ZFC-proof of*

  "if Π is a falsifiable property of zero-knowledge, then ψ_λ is unsatisfiable"

  has length at least $\ell(\lambda)$.

We get the following theorem.

Theorem 7.14. *Assume NIWIs exist and there is a hard* UNSAT *sequence for* Natural*. Then there exists a perfectly sound prover $\mathcal{P}$ such that every $\Pi \in$ Natural holds on $\mathcal{P}$.*

Proof. Let $\Psi = \{\psi_\lambda\}$ be as in Definition 7.13. Let $\mathcal{P} = \mathcal{P}[\Psi]$. By Lemma 6.1, $\mathcal{P}$ is perfectly sound. By Proposition 6.6, we have that $\mathcal{P}$ is effectively zero-knowledge to ZFC with hardness Ψ with associated parameters $\epsilon^\star = \lambda^{-\omega(1)}$ and $s^\star = \mathsf{poly}(\lambda)$.

Now fix a $\Pi = (G, \Delta) \in$ Natural, a λ, and an adversary A.

Claim 7.15. *There is a $\mathsf{poly}(|A|, |G|, |\mathcal{P}|, \log \frac{\lambda s\tau}{\epsilon})$-length ZFC-proof that*

"if $\mathcal{P}$ has an s-size ϵ-indistinguishable simulator on λ and Π is a falsifiable property of zero-knowledge, then $\mathsf{Estimate}(G, (\mathcal{P}, A, 1^\lambda), 1^\tau) \leq \frac{1}{\tau} + \Delta(\lambda, A, s, \epsilon)$*"*

Proof. This follows from the definition of a falsifiable property of zero-knowledge and because we assumed the correctness of $\mathsf{Estimate}$ is provable in ZFC. □

Hence, because $\mathcal{P}[\Psi]$ is effectively zero-knowledge to ZFC with hardness Ψ, we get that for some sufficiently small $\tau^\star = \lambda^{\omega(1)}$ that we choose later

$$\mathsf{Estimate}(G, (\mathcal{P}, A, 1^\lambda), 1^{\tau^\star}) \leq \frac{1}{\tau^\star} + \Delta(\lambda, A, s^\star, \epsilon^\star),$$

and thus that

$$\Pr[G(\mathcal{P}, A, 1^\lambda) \text{ outputs one in time } \tau^\star] \leq \frac{2}{\tau^\star} + \Delta(\lambda, A, s^\star, \epsilon^\star) \tag{5}$$

as long as there is no $\mathsf{poly}(|A|, |G|, |\mathcal{P}|, \tau^\star, \log \frac{\lambda s^\star \tau^\star}{\epsilon^\star})$-length $\mathrm{ZFC}_{extended}$-proof of

"if Π is a falsifiable property of zero-knowledge, then ψ_λ is unsatisfiable."

Indeed, by assumption the above statement requires $\lambda^{\omega(1)}$-length ZFC-proofs and hence also $\mathrm{ZFC}_{extended}$-proofs (since we assumed ZFC simulates $\mathrm{ZFC}_{extended}$). So we can choose $\tau^\star = \lambda^{\omega(1)}$ sufficiently small such that (5) indeed holds. Thus, Π holds on $\mathcal{P}$. □

7.4 Low Non-uniformity Simulators

We extend our result to simulators with a small amount of non-uniformity.

Theorem 7.16. *Assume*

- *NIWIs exist, and*
- *there is no infinitely often subexponentially optimal proof system.*

Then for every proof system $\mathcal{L}$, there exists a perfectly sound (uniform) prover $\mathcal{P}$ that is $(\mathsf{poly}\log\lambda)$-non-uniform effectively zero-knowledge to $\mathcal{L}$.

Proof. Fix a proof system $\mathcal{L}$. By Theorem 6.10 (and reindexing over λ appropriately), there exists a P-uniform sequence Ψ of $\mathsf{poly}\log(\lambda)$-size unsatisfiable formulas that are hard for $\mathcal{L}$. Set $\mathcal{P} = \mathcal{P}[\Psi]$. By Theorem 6.5 we get that $\mathcal{P}$ is $(\mathsf{poly}\log\lambda)$-non-uniform effectively zero-knowledge to $\mathcal{L}$. By Lemma 6.1, $\mathcal{P}$ is perfectly sound. □

7.5 Hard Tautology Generators

A hard tautology[52] generator [Kha24] is, roughly, a polynomial-time algorithm that takes as input the code of a propositional proof system and outputs an unsatisfiable formula that is hard to prove unsatisfiable in that proof system.

Definition 7.17 (Efficient Almost Everywhere Hard Tautology Generator)**.** *Let H be a polynomial-time Turing machine. We say H is an* efficient almost everywhere hard tautology generator *if for every propositional proof system $\mathcal{L}$, there is an $\ell(\lambda) = \lambda^{\omega(1)}$ such that $\psi_\lambda = H(\mathcal{L}, 1^\lambda)$[53] is an unsatisfiable formula with no $\ell(\lambda)$-length $\mathcal{L}$-proof of "ψ_λ is unsatisfiable."*

Khaniki [Kha24] conjectures that an efficient hard tautology generator exists and shows that this is implied by (even a weak version of) Pudlák's finite Gödel conjecture [Pud86]. But the definition we use actually differs from the definition used in [Kha24] in the following way. While Definition 7.17 requires that $\ell(\lambda)$ is $\lambda^{\omega(1)}$, Khaniki only requires that $\ell(\lambda) = \lambda^{\omega(1)}$ for infinitely many λ. In other words, we require

[52] Even though we use UNSAT rather than TAUT, we keep "tautology" in the name to match the literature.
[53] Here, H takes as input the underlying code of $\mathcal{L}$.

almost everywhere hardness as opposed to infinitely often hardness. Henceforth, we skip writing "almost everywhere" for brevity.

If efficient hard tautology generators exist, then we can efficiently generate perfectly sound provers that are effectively zero-knowledge to a given proof system. However, we are admittedly less certain that efficient hard tautology generators exist (in full generality), compared to assuming no infinitely often optimal proof system exists.

Theorem 7.18. *Assume NIWIs exist and there is an efficient hard tautology generator H. Then there is a polynomial-time algorithm A such that for every proof system $\mathcal{L}$ we have that $A(\mathcal{L})$ outputs the code of a perfectly sound uniform prover that is effectively zero-knowledge to $\mathcal{L}$.*

Proof. For every proof system $\mathcal{L}$, we can consider the related propositional proof system $\mathcal{L}^{prop}$ given by

$$\mathcal{L}^{prop}(x) = \begin{cases} \varphi, & \text{if } \mathcal{L}(x) = \text{“}\varphi \text{ is unsatisfiable”} \\ x_1 \wedge \neg x_1, & \text{otherwise.} \end{cases}.$$

The algorithm A works as follows. Given $\mathcal{L}$, it constructs $\mathcal{L}_{extended}$ and lets $\psi_\lambda = H(\mathcal{L}^{prop}_{extended}, 1^\lambda)$. It then outputs the code of the prover $\mathcal{P} = \mathcal{P}[\Psi]$. Clearly, A runs in polynomial time. It remains to argue for correctness.

By construction, $\Psi = \{\psi_\lambda\}$ is hard for $\mathcal{L}$, so by Theorem 6.5, we have that $\mathcal{P}$ is effectively zero-knowledge to $\mathcal{L}$. Since each ψ_λ is unsatisfiable, we get that $\mathcal{P}$ is perfectly sound by Lemma 6.1. □

7.6 An Alternative Definition

We formally define the alternative to our main definition that is based on unprovability rather than t-time indistinguishability.

Definition 7.19 (Unprovability-Based Definition of Effectively Zero-Knowledge to $\mathcal{L}$)**.** *Let $\mathcal{P}$ be a prover, and let $\mathcal{L}$ be a proof system. We say $\mathcal{P}$ is* unprovability-based effectively zero-knowledge to $\mathcal{L}$ *if for some $t = \lambda^{\omega(1)}$ and some $s = \mathsf{poly}(\lambda)$, we have that for all $\lambda \in \mathbb{N}$*

$$\text{“}\mathcal{P} \textit{ does not have an } s(\lambda)\textit{-size } \frac{1}{t(\lambda)}\textit{-indistinguishable simulator on } \lambda\text{”}$$

lacks a $t(\lambda)$-length $\mathcal{L}$-proof.

7.7 The Necessity of No Optimal Proof Systems

In this subsection, we give evidence that the non-existence of optimal proof systems is necessary for our results. To do so, the following cryptographic object will be useful.

Definition 7.20 (Bit Commitment Scheme [Blu82])**.** *An S-secure perfectly-binding bit commitment scheme is a randomized polynomial-time algorithm $Commit(b \in \{0,1\}, 1^\lambda)$ with the following two properties:*

- perfectly binding: *the support of $Commit(0,\cdot)$ and $Commit(1,\cdot)$ are disjoint, and*
- computationally hiding: *for all λ, we have $Commit(0,1^\lambda) \approx_{1/S(\lambda)} Commit(1,1^\lambda)$.*

Such schemes exist assuming the existence of injective one-way functions.

Theorem 7.21 ([Blu82; Yao82; GL89])**.** *Assume there is an injective one-way function $f : \{0,1\}^n \to \{0,1\}^{\mathsf{poly}(n)}$ with security $S(n)$. Then there is an $\left(\frac{S(\lambda)}{\mathsf{poly}(\lambda)}\right)^{\Omega(1)}$-secure perfectly-binding bit commitment scheme.*

Assuming the existence of injective one-way functions with slightly superpolynomial security, we show the necessity of the optimal proof system assumption. The high-level idea is this. Recall Barak, Ong, and Vadhan's attack that "the verifier gains the ability to prove the same statement to others." It turns out that one can strengthen this attack by using the guarantees of an optimal proof system. The reason is that an optimal proof system can (almost) prove the soundness of *any* perfectly sound verifier.

Theorem 7.22 (Necessity of No Optimal Proof Systems)**.** *Assume an injective one-way function with security $n^{\Omega(\log^\star n)}$ exists and an infinitely often optimal proof system $\mathcal{L}_{opt}$ exists. Then there is a falsifiable property of zero-knowledge that does not hold on any perfectly sound prover.*

Proof. Let $Commit$ be the commitment scheme guaranteed by Theorem 7.21. Let $\mathcal{L}$ be a proof system we choose later. Set $\lambda'(\lambda) = \lambda^{\Omega(1)}$ sufficiently small. Consider the following game G.

$G(\mathcal{P}, A, 1^\lambda)$:

1. Let $c \leftarrow Commit(1, 1^{\lambda'}; r)$, where r is the internal randomness used. Let φ_c be the SAT instance corresponding to "$c = Commit(1, 1^{\lambda'}; r')$ for some r'." We set $\lambda' = \lambda^{\Omega(1)}$ sufficiently small such that $|\varphi_c| \leq \lambda$.
2. Sample $\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi_c, r, 1^\lambda)$.
3. Accept if $A(c, \pi)$ outputs a $\mathcal{L}$-proof that "φ_c is satisfiable."

We begin by showing this is a falsifiable property of zero-knowledge.

Claim 7.23. *$\Pi = (G, \Delta)$ is a falsifiable property of zero-knowledge for some $\Delta = (\lambda^{-\omega(1)} + \epsilon) \cdot \mathsf{poly}(|A|, s, \lambda)$.*

Proof. Let $\mathcal{P}$ be a prover with an s-size ϵ-indistinguishable simulator Sim_λ on λ, and let A be an adversary. Consider the circuit $A'(c)$ that outputs one if and only if $A(c, Sim_\lambda(\varphi_c))$ outputs a $\mathcal{L}$-proof that "φ_c is satisfiable." Because $\mathcal{L}$ is a proof system (and hence is perfectly sound), the security of the commitment scheme implies that

$$\Pr_{c \leftarrow Commit(1,1^\lambda)}[A'(c) = 1] \leq \lambda^{-\Omega(\log^\star \lambda)} \cdot \mathsf{poly}(|A'|).$$

Then by the security of Sim_λ and the construction of G, we have that

$$\Pr[G(1^\lambda, \mathcal{P}, A) = 1] \leq \left(\lambda^{-\Omega(\log^\star \lambda)} + \epsilon\right) \cdot \mathsf{poly}(|A|, s, \lambda)$$

□

It remains to show that Π does not hold on any perfectly sound prover. Note that for every prover $\mathcal{P}$

"for all $\lambda \in [\ell]$ and all $c \leftarrow Commit(0, 1^\lambda)$ and all $\pi \in \{0,1\}^{\leq \ell}$ we have $\mathcal{P}.\mathsf{verify}(\varphi_c, \pi, 1^\lambda) = 0$ "

can be encoded as an instance $\psi_{\mathcal{P},\ell}$ of UNSAT of size $\mathsf{poly}(\ell)$, where the polynomial can depend on $\mathcal{P}$. We make the following claim.

Claim 7.24. *For every perfectly sound prover $\mathcal{P}$, there are infinitely many ℓ such that "$\psi_{\mathcal{P},\ell}$ is unsatisfiable" has an $\mathcal{L}_{opt}$-proof of length $\mathsf{poly}(\ell)$, where the polynomial can depend on $\mathcal{P}$.*

Proof. Fix a perfectly sound prover $\mathcal{P}$. By perfect soundness, we have that $\psi_{\mathcal{P},\ell}$ is unsatisfiable for all ℓ. Thus, we can construct a trivial propositional proof system with $\mathsf{poly}(\ell)$-length proofs that "$\psi_{\mathcal{P},\ell}$ is unsatisfiable." Then the claim follows from the infinitely often optimality of $\mathcal{L}_{opt}$. □

We now choose $\mathcal{L}$ as follows.

$\mathcal{L}(\mathcal{P}, \varphi, \lambda, \pi, \pi')$:

1. Output "φ is satisfiable" if $\mathcal{P}.\mathsf{verify}(\varphi, \pi, 1^\lambda) = 1$ and $\mathcal{L}_{opt}(\pi') = \psi_{\mathcal{P},\ell}$, where $\ell \geq \max\{\lambda, |\pi|\}$.
2. Output a fixed element of $\mathsf{P}^{\mathsf{HALT}}$ otherwise.

We have constructed $\mathcal{L}$ specifically so that the following claim holds, which finishes the proof.

Claim 7.25. *Π does not hold on any perfectly sound prover.*

Proof. Fix a perfectly sound prover $\mathcal{P}$. Let $\ell = \ell(\lambda) = \mathsf{poly}(\lambda)$ be the length of the π generated in step (2) of G when run on $\mathcal{P}$ and 1^λ. By Claim 7.24, there are infinitely many λ such that there exists a $\mathsf{poly}(\ell)$-length $\mathcal{L}_{opt}$-proof π_ℓ of "$\psi_{\mathcal{P},\ell}$ is unsatisfiable." For such a λ, consider the adversary circuit $A(c, \pi)$ which has π_ℓ as non-uniform advice and outputs $(\mathcal{P}, \varphi_c, \lambda, \pi, \pi_\ell)$. By construction of G and $\mathcal{L}$, we will have that $G(\mathcal{P}, A, 1^\lambda)$ outputs 1 in $\mathsf{poly}(\lambda)$ time. Hence, we have that Π does not hold on $\mathcal{P}$. □

□

7.8 Application to TFNP

We recall the definitions of Search-NP and TFNP [MP91].

Definition 7.26 (Search-NP and TFNP Problems)**.** *A* Search-NP problem *is defined by a polynomial-time Turing machine R that takes as input an "instance" $x \in \{0,1\}^\star$ and a "witness" $w \in \{0,1\}^{\mathsf{poly}(|x|)}$ and outputs a bit. We say an n-input circuit C* computes *R if for all $x \in \{0,1\}^n$ we have that $R(x, C(x)) = 1$ whenever there exists a w with $R(x, w) = 1$. We say R is a* TFNP problem *if for all x there exists a w with $R(x, w) = 1$.*

Every Search-NP problem has a natural TFNP analogue modulo a choice of perfectly sound prover [HNY17]. The idea is that one also includes a proof π that the instance has a witness.

Definition 7.27. *Let R be a* Search-NP *problem. Let $\mathcal{P}$ be a perfectly sound prover. Define the related* TFNP *problem*

$$R_{L,\mathcal{P}}((x, \pi, 1^\lambda), w) = \begin{cases} 1, & \text{if } R(x, w) = 1 \\ 1, & \text{if } \mathcal{P}.\mathsf{verify}(\varphi_x, \pi, 1^\lambda) = 0 \\ 0, & \text{otherwise} \end{cases}$$

where φ_x is the $\mathsf{poly}(|x|)$-sized formula encoding "there exists a w with $R(x, w) = 1$."

It is easy to see that solving $R_{L,\mathcal{P}}$ is only easier (up to polynomial blow-up) than solving R. We show a converse to this.

Corollary 7.28. *Assume*

- $\mathsf{P} = \mathsf{BPP}$,
- *subexponentially-secure NIWIs exist, and*
- *there is no infinitely often subexponentially optimal proof system.*

Let $S : \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable function. If a Search-NP *problem R requires circuits of size $S(n)$, then there exists a perfectly sound prover $\mathcal{P}$ such that solving $R_{L,\mathcal{P}}$ on inputs of the form $(x \in \{0,1\}^n, \pi \in \{0,1\}^{\mathsf{poly}(n)}, 1^{\mathsf{poly}(n)})$ requires circuits of size $\frac{S(n)}{\mathsf{poly}(n)}$.*

Proof. Consider the following game G.

$G(\mathcal{P}, A, 1^\lambda)$:

1. Let n be the number of input bits to the circuit A.
2. For all $x \in \{0,1\}^n$:
   (a) Let $\varphi_x(w)$ be the formula of size $\mathsf{poly}(n)$ that outputs one if and only if $R(x,w) = 1$.
   (b) By brute force, find a w with $\varphi_x(w) = 1$. If none exists, then go to the next x.
   (c) Using $\mathsf{Estimate}$, deterministically compute an estimate v of $\Pr_{\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi_x, w, 1^\lambda)}[R(x, A(x,\pi)) = 1]$ to within additive error .01.
   (d) Output 0 if $v \leq .75$.
3. Output 1.

Observe that G is deterministic and runs in time $2^{\mathsf{poly}(n)} \cdot \mathsf{poly}(|A|, \lambda)$. We now show this is a falsifiable property of zero-knowledge.

Claim 7.29. $\Pi = (G, \Delta)$ *is a falsifiable property of zero-knowledge where*

$$\Delta = \mathbb{1}\left[|A| \geq \min\left\{\frac{S(n)}{\mathsf{poly}(n,s)}, \frac{1}{\epsilon} - \mathsf{poly}(s)\right\}\right].$$

Proof. Fix a λ, an adversary A, and a perfectly sound prover with an s-size ϵ-indistinguishable simulator Sim_λ on λ. Suppose that $G(\mathcal{P}, A, 1^\lambda) = 1$ (recall G is deterministic). Our goal is to show that $|A|$ is large. If $|A| > \frac{1}{\epsilon} - \mathsf{poly}(s)$, then we are done. Otherwise, $|A| \leq \frac{1}{\epsilon} - \mathsf{poly}(s)$, so we have that $A'(x) = A(x, Sim_\lambda(\varphi_x))$ is a randomized circuit that solves R with probability at least $.74 - \epsilon \geq .51$. Then, by Adleman's trick,[54] there is a circuit A'' of size $\mathsf{poly}(n) \cdot |A'|$ that solves R. Hence, we have that

$$S(n) \leq |A''| \leq |A'| \cdot \mathsf{poly}(n) \leq |A| \cdot \mathsf{poly}(n,s),$$

which implies that $|A| \geq S(n)/\mathsf{poly}(n,s)$. □

Hence, by Theorem 2.1, there exists a perfectly sound prover $\mathcal{P}$ on which Π holds with subexponential security. Now let $\lambda = \lambda(n) = \mathsf{poly}(n)$ be a sufficiently large polynomial we choose later. Now fix an adversary circuit A that solves $R_{L,\mathcal{P}}(x \in \{0,1\}^n, \pi \in \{0,1\}^{\mathsf{poly}(\lambda)}, 1^\lambda)$. It follows that $G(\mathcal{P}, A, 1^\lambda) = 1$ in time at most $2^{\mathsf{poly}(n)} \cdot \mathsf{poly}(|A|)$. By setting $\lambda = \mathsf{poly}(n)$ sufficiently large, we conclude that

$$|A| \geq \min\left\{\frac{S(n)}{\mathsf{poly}(n)}, \frac{1}{2^{-2n}}\right\} \geq S(n)/\mathsf{poly}(n)$$

since Π holds on $\mathcal{P}$ with subexponential security and since $S(n) \leq 2^n \cdot \mathsf{poly}(n)$ (by the trivial circuit upper bound for computing R). □

7.9 Application to NP ∩ coNP

Recall a language L is in NP if and only if there is a polynomial-time Turing machine R such that $x \in L$ if and only if $R(x,w) = 1$ for some w of length at most $\mathsf{poly}(|x|)$. Furthermore, L is in UP if and only if there exists such a R with the additional property that for all x we have that $|\{w : R(x,w) = 1\}| \leq 1$.

As done in [GIK+23], for any UP language, one can construct a related $\mathsf{NP} \cap \mathsf{coNP}$ language modulo a choice of a perfectly sound prover.

[54] Repeat A independently polynomial many times until the failure probability becomes small enough that one can non-uniformly fix a setting of random coins that work for all inputs.

Definition 7.30. *Let $L \in \mathsf{UP}$. Let $\mathcal{P}$ be a perfectly sound prover. Define the related $\mathsf{NP} \cap \mathsf{coNP}$ language $L_{\mathcal{P}}$ whose membership function is given by*

$$L_{\mathcal{P}}(x, i, \pi, 1^\lambda) = \begin{cases} 0, & \text{if } \mathcal{P}.\mathsf{verify}(\varphi_x, \pi, 1^\lambda) = 0 \\ w_i, & \text{otherwise, where } w_i \text{ is the } i\text{'th bit of the unique witness for } x \end{cases}$$

where φ_x is the $\mathsf{poly}(|x|)$-sized formula encoding "there exists a w with $R(x, w) = 1$."

We show that $L_{\mathcal{P}}$ is roughly as hard as L for some choice of $\mathcal{P}$.

Corollary 7.31. *Assume*

- $\mathsf{P} = \mathsf{BPP}$,
- *subexponentially secure NIWIs exist, and*
- *there is no infinitely often subexponentially optimal proof system.*

Let $S : \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable function. If $L \in \mathsf{UP}$ requires circuits of size $S(n)$, then there exists a perfectly sound prover such that solving $L_{\mathcal{P}}$ on inputs of the form $(x \in \{0,1\}^n, i \in [\mathsf{poly}(n)], \pi \in \{0,1\}^{\mathsf{poly}(n)}, 1^{\mathsf{poly}(n)})$ requires circuits of size $\frac{S(n)}{\mathsf{poly}(n)}$.

Proof. The proof is similar to the proof of Corollary 7.28, except on the following game.

$G(\mathcal{P}, A, 1^\lambda)$:

1. Let n be the number of inputs to the circuit A. For all $x \in \{0,1\}^n$:
   (a) Let $\varphi_x(w)$ be the formula of size $\mathsf{poly}(n)$ that is satisfiable if $x \in L$.
   (b) By brute force, find a w with $\varphi_x(w) = 1$. If none exists, then go to the next x.
   (c) For all $i \in [|w|]$:
      i. Using $\mathsf{Estimate}$, deterministically compute an estimate v of
      $$\Pr_{\pi \leftarrow \mathcal{P}.\mathsf{prove}(\varphi_x, w, 1^\lambda)}[R(x, A(x, i, \pi)) = w_i]$$
      to within additive error .01.
      ii. Output 0 if $v \leq .75$.
2. Output 1.

□

Acknowledgments

We are especially grateful to Alex Lombardi, Roei Tell, and Neekon Vafa for their insights, questions, and valuable feedback. We thank Pavel Pudlák and Erfan Khaniki for their patient and detailed answers to questions about their works [Pud86; Kha24]. We also thank Yael Kalai, Jiatu Li, Igor Oliveira, Pavel Pudlák, Vinod Vaikuntanathan, Ryan Williams, and anonymous FOCS and STOC reviewers for helpful discussions and comments on this paper. This work was supported by NSF CCF-2420092 and an NSF graduate research fellowship.

References

[Bey07] Olaf Beyersdorff. "Classes of representable disjoint NP-pairs". In: *Theor. Comput. Sci.* 377.1-3 (2007), pp. 93–109. DOI: 10.1016/J.TCS.2007.02.005. URL: https://doi.org/10.1016/j.tcs.2007.02.005 (cit. on p. 7).

[Bey10] Olaf Beyersdorff. "The Deduction Theorem for Strong Propositional Proof Systems". In: *Theory Comput. Syst.* 47.1 (2010), pp. 162–178. DOI: 10.1007/S00224-008-9146-6. URL: https://doi.org/10.1007/s00224-008-9146-6 (cit. on p. 7).

[BF03] Dan Boneh and Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *SIAM J. Comput.* 32.3 (2003), pp. 586–615. DOI: 10.1137/S0097539701398521. URL: https://doi.org/10.1137/S0097539701398521 (cit. on p. 8).

[BFFM00] Harry Buhrman, Stephen A. Fenner, Lance Fortnow, and Dieter van Melkebeek. "Optimal Proof Systems and Sparse Sets". In: *STACS 2000*. Vol. 1770. Lecture Notes in Computer Science. Springer, 2000, pp. 407–418. DOI: 10.1007/3-540-46541-3_34. URL: https://doi.org/10.1007/3-540-46541-3%5C_34 (cit. on p. 7).

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *STOC 1988*. ACM, 1988, pp. 103–112. DOI: 10.1145/62212.62222. URL: https://doi.org/10.1145/62212.62222 (cit. on pp. 3, 5).

[BG98] Shai Ben-David and Anna Gringauze. "On the Existence of Propositional Proof Systems and Oracle-relativized Propositional Logic". In: *Electron. Colloquium Comput. Complex.* TR98-021 (1998). ECCC: TR98-021. URL: https://eccc.weizmann.ac.il/eccc-reports/1998/TR98-021/index.html (cit. on p. 7).

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (im)possibility of obfuscating programs". In: *J. ACM* 59.2 (2012), 6:1–6:48. DOI: 10.1145/2160158.2160159. URL: https://doi.org/10.1145/2160158.2160159 (cit. on pp. 6, 7, 24, 48).

[BKM09] Olaf Beyersdorff, Johannes Köbler, and Jochen Messner. "Nondeterministic functions and the existence of optimal proof systems". In: *Theor. Comput. Sci.* 410.38-40 (2009), pp. 3839–3855. DOI: 10.1016/J.TCS.2009.05.021. URL: https://doi.org/10.1016/j.tcs.2009.05.021 (cit. on p. 7).

[BKM11] Olaf Beyersdorff, Johannes Köbler, and Sebastian Müller. "Proof systems that take advice". In: *Inf. Comput.* 209.3 (2011), pp. 320–332. DOI: 10.1016/J.IC.2010.11.006. URL: https://doi.org/10.1016/j.ic.2010.11.006 (cit. on p. 7).

[BL18] Nir Bitansky and Huijia Lin. "One-Message Zero Knowledge and Non-malleable Commitments". In: *TCC 2018*. Vol. 11239. Lecture Notes in Computer Science. Springer, 2018, pp. 209–234. DOI: 10.1007/978-3-030-03807-6_8. URL: https://doi.org/10.1007/978-3-030-03807-6%5C_8 (cit. on p. 10).

[Blu82] Manuel Blum. "Coin Flipping by Phone". In: *Proceedings of the 24th IEEE Computer Conference (CompCon)*. IEEE. 1982, pp. 133–137 (cit. on p. 38).

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. "Derandomization in Cryptography". In: *SIAM J. Comput.* 37.2 (2007), pp. 380–400. DOI: 10.1137/050641958. URL: https://doi.org/10.1137/050641958 (cit. on pp. 3, 5, 7, 8, 10, 15, 22, 26, 35).

[BP04] Boaz Barak and Rafael Pass. "On the Possibility of One-Message Weak Zero-Knowledge". In: *TCC 2004*. Vol. 2951. Lecture Notes in Computer Science. Springer, 2004, pp. 121–132. DOI: 10.1007/978-3-540-24638-1_7. URL: https://doi.org/10.1007/978-3-540-24638-1%5C_7 (cit. on pp. 10, 24).

[BP15] Nir Bitansky and Omer Paneth. "ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation". In: *TCC 2015*. Vol. 9015. Lecture Notes in Computer Science. Springer, 2015, pp. 401–427. DOI: 10.1007/978-3-662-46497-7_16. URL: https://doi.org/10.1007/978-3-662-46497-7%5C_16 (cit. on pp. 7, 8, 22).

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. "On the Cryptographic Hardness of Finding a Nash Equilibrium". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. IEEE Computer Society, 2015, pp. 1480–1498. DOI: 10.1109/FOCS.2015.94. URL: https://doi.org/10.1109/FOCS.2015.94 (cit. on p. 10).

[BR93] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*. ACM, 1993, pp. 62–73. DOI: 10.1145/168588.168596. URL: https://doi.org/10.1145/168588.168596 (cit. on p. 9).

[BS09] Olaf Beyersdorff and Zenon Sadowski. "Characterizing the Existence of Optimal Proof Systems and Complete Sets for Promise Classes". In: *CSR 2009*. Vol. 5675. Lecture Notes in Computer Science. Springer, 2009, pp. 47–58. DOI: 10.1007/978-3-642-03351-3_7. URL: https://doi.org/10.1007/978-3-642-03351-3%5C_7 (cit. on p. 7).

[BS11] Olaf Beyersdorff and Zenon Sadowski. "Do there exist complete sets for promise classes?" In: *Math. Log. Q.* 57.6 (2011), pp. 535–550. DOI: 10.1002/MALQ.201010021. URL: https://doi.org/10.1002/malq.201010021 (cit. on p. 7).

[BSMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. "Noninteractive Zero-Knowledge". In: *SIAM J. Comput.* 20.6 (1991), pp. 1084–1118. DOI: 10.1137/0220068. URL: https://doi.org/10.1137/0220068 (cit. on pp. 3, 5).

[CF10] Yijia Chen and Jörg Flum. "On Slicewise Monotone Parameterized Problems and Optimal Proof Systems for TAUT". In: *CSL 2010*. Vol. 6247. Lecture Notes in Computer Science. Springer, 2010, pp. 200–214. DOI: 10.1007/978-3-642-15205-4_18. URL: https://doi.org/10.1007/978-3-642-15205-4%5C_18 (cit. on p. 7).

[CFM14] Yijia Chen, Jörg Flum, and Moritz Müller. "Hard Instances of Algorithms and Proof Systems". In: *ACM Trans. Comput. Theory* 6.2 (2014), 7:1–7:25. DOI: 10.1145/2601336. URL: https://doi.org/10.1145/2601336 (cit. on p. 7).

[CR79] Stephen A. Cook and Robert A. Reckhow. "The Relative Efficiency of Propositional Proof Systems". In: *J. Symb. Log.* 44.1 (1979), pp. 36–50. DOI: 10.2307/2273702. URL: https://doi.org/10.2307/2273702 (cit. on pp. 4, 16, 20).

[CT23] Lijie Chen and Roei Tell. "Guest Column: New ways of studying the BPP = P conjecture". In: *SIGACT News* 54.2 (2023), pp. 44–69. DOI: 10.1145/3604943.3604950. URL: https://doi.org/10.1145/3604943.3604950 (cit. on p. 8).

[DG20] Titus Dose and Christian Glaßer. "NP-Completeness, Proof Systems, and Disjoint NP-Pairs". In: *STACS 2020*. Vol. 154. LIPIcs. 2020, 9:1–9:18. DOI: 10.4230/LIPICS.STACS.2020.9. URL: https://doi.org/10.4230/LIPIcs.STACS.2020.9 (cit. on p. 7).

[DN07] Cynthia Dwork and Moni Naor. "Zaps and Their Applications". In: *SIAM J. Comput.* 36.6 (2007), pp. 1513–1543. DOI: 10.1137/S0097539703426817. URL: https://doi.org/10.1137/S0097539703426817 (cit. on pp. 8, 22).

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)". In: *FOCS 1990*. IEEE Computer Society, 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549. URL: https://doi.org/10.1109/FSCS.1990.89549 (cit. on pp. 7, 8, 22).

[FS90] Uriel Feige and Adi Shamir. "Witness Indistinguishable and Witness Hiding Protocols". In: *STOC 1990*. ACM, 1990, pp. 416–426. DOI: 10.1145/100216.100272. URL: https://doi.org/10.1145/100216.100272 (cit. on pp. 3, 6–8, 12, 22, 23, 26).

[GIK+23] Riddhi Ghosal, Yuval Ishai, Alexis Korb, Eyal Kushilevitz, Paul Lou, and Amit Sahai. "Hard Languages in NP ∩ coNP and NIZK Proofs from Unstructured Hardness". In: *STOC 2023*. ACM, 2023, pp. 1243–1256. DOI: 10.1145/3564246.3585119. URL: https://doi.org/10.1145/3564246.3585119 (cit. on pp. 6, 41).

[GK16] Shafi Goldwasser and Yael Tauman Kalai. "Cryptographic Assumptions: A Position Paper". In: *TCC 2016*. Vol. 9562. Lecture Notes in Computer Science. Springer, 2016, pp. 505–522. DOI: 10.1007/978-3-662-49096-9_21. URL: https://doi.org/10.1007/978-3-662-49096-9%5C_21 (cit. on p. 24).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for All One-Way Functions". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*. ACM, 1989, pp. 25–32 (cit. on p. 38).

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM J. Comput.* 18.1 (1989), pp. 186–208. DOI: 10.1137/0218012. URL: https://doi.org/10.1137/0218012 (cit. on pp. 3, 7, 11).

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

[GO94] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *J. Cryptol.* 7.1 (1994), pp. 1–32. DOI: 10.1007/BF00195207. URL: https://doi.org/10.1007/BF00195207 (cit. on pp. 3, 4, 8, 32).

[Göd31] Kurt Gödel. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. New York, NY, USA: Basic Books, 1931 (cit. on pp. 7, 21).

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. ISBN: 0-521-79172-3. DOI: 10.1017/CBO9780511546891. URL: http://www.wisdom.weizmann.ac.il/%5C%7Eoded/foc-vol1.html (cit. on p. 25).

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004. ISBN: 0-521-83084-2. DOI: 10.1017/CBO9780511721656. URL: http://www.wisdom.weizmann.ac.il/%5C%7Eoded/foc-vol2.html (cit. on p. 25).

[Gol11] Oded Goldreich. "In a World of P=BPP". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Vol. 6650. Lecture Notes in Computer Science. Springer, 2011, pp. 191–232. DOI: 10.1007/978-3-642-22670-0_20. URL: https://doi.org/10.1007/978-3-642-22670-0%5C_20 (cit. on p. 8).

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "New Techniques for Noninteractive Zero-Knowledge". In: *J. ACM* 59.3 (2012), 11:1–11:35. DOI: 10.1145/2220357.2220358. URL: https://doi.org/10.1145/2220357.2220358 (cit. on pp. 8, 22).

[GSSZ04] Christian Glaßer, Alan L. Selman, Samik Sengupta, and Liyu Zhang. "Disjoint NP-Pairs". In: *SIAM J. Comput.* 33.6 (2004), pp. 1369–1416. DOI: 10.1137/S0097539703425848. URL: https://doi.org/10.1137/S0097539703425848 (cit. on p. 7).

[HIMS12] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. "On Optimal Heuristic Randomized Semidecision Procedures, with Applications to Proof Complexity and Cryptography". In: *Theory Comput. Syst.* 51.2 (2012), pp. 179–195. DOI: 10.1007/S00224-011-9354-3. URL: https://doi.org/10.1007/s00224-011-9354-3 (cit. on p. 7).

[HKKS20] Pavel Hubácek, Chethan Kamath, Karel Král, and Veronika Slívová. "On Average-Case Hardness in TFNP from One-Way Functions". In: *TCC 2020*. Vol. 12552. Lecture Notes in Computer Science. Springer, 2020, pp. 614–638. DOI: 10.1007/978-3-030-64381-2_22. URL: https://doi.org/10.1007/978-3-030-64381-2%5C_22 (cit. on p. 10).

[HNY17] Pavel Hubácek, Moni Naor, and Eylon Yogev. "The Journey from NP to TFNP Hardness". In: *ITCS 2017*. Vol. 67. LIPIcs. 2017, 60:1–60:21. DOI: 10.4230/LIPICS.ITCS.2017.60. URL: https://doi.org/10.4230/LIPIcs.ITCS.2017.60 (cit. on pp. 5, 10, 40).

[Ila25] Rahul Ilango. *Proof Complexity, the Difficulty of Analyzing Computation, and Chaitin for Relative Circuit Complexity*. Manuscript in preperation. 2025 (cit. on p. 6).

[IW97] Russell Impagliazzo and Avi Wigderson. "$P = BPP$ if E Requires Exponential Circuits: Derandomizing the XOR Lemma". In: *STOC 1997*. ACM, 1997, pp. 220–229. DOI: 10.1145/258533.258590. URL: https://doi.org/10.1145/258533.258590 (cit. on p. 8).

[JJ22] Abhishek Jain and Zhengzhong Jin. "Indistinguishability Obfuscation via Mathematical Proofs of Equivalence". In: *FOCS 2022*. IEEE, 2022, pp. 1023–1034. DOI: 10.1109/FOCS54457.2022.00100. URL: https://doi.org/10.1109/FOCS54457.2022.00100 (cit. on pp. 9, 10).

[JKLM25] Zhengzhong Jin, Yael Tauman Kalai, Alex Lombardi, and Surya Mathialagan. "Universal SNARGs for NP from Proofs of Correctness". In: *STOC 2025*. ACM, 2025 (cit. on pp. 9, 10).

[JKLV24] Zhengzhong Jin, Yael Kalai, Alex Lombardi, and Vinod Vaikuntanathan. "SNARGs under LWE via Propositional Proofs". In: *STOC 2024*. ACM, 2024, pp. 1750–1757. DOI: 10.1145/3618260.3649770. URL: https://doi.org/10.1145/3618260.3649770 (cit. on pp. 9, 10).

[Kha22] Erfan Khaniki. "Nisan-Wigderson Generators in Proof Complexity: New Lower Bounds". In: *CCC 2022*. Vol. 234. LIPIcs. 2022, 17:1–17:15. DOI: 10.4230/LIPICS.CCC.2022.17. URL: https://doi.org/10.4230/LIPIcs.CCC.2022.17 (cit. on p. 7).

[Kha24] Erfan Khaniki. "Jump Operators, Interactive Proofs and Proof Complexity Generators". In: *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*. 2024, pp. 573–593. DOI: 10.1109/FOCS61266.2024.00044 (cit. on pp. 7, 20, 37, 42).

[KL10] A. V. Kontorovich and J. C. Lagarias. "Stochastic models for the 3x + 1 problem and generalizations". In: (2010), pp. 131–188 (cit. on p. 24).

[KM98] Johannes Köbler and Jochen Messner. "Complete Problems for Promise Classes by Optimal Proof Systems for Test Sets". In: *Conference on Computational Complexity, 1998*. IEEE Computer Society, 1998, pp. 132–140. DOI: 10.1109/CCC.1998.694599. URL: https://doi.org/10.1109/CCC.1998.694599 (cit. on p. 7).

[KMT03] Johannes Köbler, Jochen Messner, and Jacobo Torán. "Optimal proof systems imply complete sets for promise classes". In: *Inf. Comput.* 184.1 (2003), pp. 71–92. DOI: 10.1016/S0890-5401(03)00058-0. URL: https://doi.org/10.1016/S0890-5401(03)00058-0 (cit. on p. 7).

[KP89] Jan Krajíček and Pavel Pudlák. "Propositional Proof Systems, the Consistency of First Order Theories and the Complexity of Computations". In: *J. Symb. Log.* 54.3 (1989), pp. 1063–1079. DOI: 10.2307/2274765. URL: https://doi.org/10.2307/2274765 (cit. on pp. 7, 20, 21, 31).

[Kra14] Jan Krajíček. "On the computational complexity of finding hard tautologies". In: *Bulletin of the London Mathematical Society* 46.1 (2014), pp. 111–125. DOI: https://doi.org/10.1112/blms/bdt071. URL: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/blms/bdt071 (cit. on p. 7).

[KZ20] Benjamin Kuykendall and Mark Zhandry. "Towards Non-interactive Witness Hiding". In: *TCC 2020*. Vol. 12550. Lecture Notes in Computer Science. Springer, 2020, pp. 627–656. DOI: 10.1007/978-3-030-64375-1_22. URL: https://doi.org/10.1007/978-3-030-64375-1%5C_22 (cit. on pp. 3, 5, 8, 10, 12, 23, 24, 28).

[LW92] J. C. Lagarias and A. Weiss. "The 3x + 1 Problem: Two Stochastic Models". In: *The Annals of Applied Probability* 2.1 (1992), pp. 229–261. ISSN: 10505164, 21688737. (Visited on 07/13/2025) (cit. on p. 24).

[MDS25] Yaohua Ma, Chenxin Dai, and Elaine Shi. "Quasi-Linear Indistinguishability Obfuscation via Mathematical Proofs of Equivalence and Applications". In: *IACR Cryptol. ePrint Arch.* (2025), p. 307. URL: https://eprint.iacr.org/2025/307 (cit. on pp. 9, 10).

[Meß99] Jochen Meßner. "On Optimal Algorithms and Optimal Proof Systems". In: *STACS 99*. Vol. 1563. Lecture Notes in Computer Science. Springer, 1999, pp. 541–550. DOI: 10.1007/3-540-49116-3_51. URL: https://doi.org/10.1007/3-540-49116-3%5C_51 (cit. on p. 7).

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. "On Total Functions, Existence Theorems and Computational Complexity". In: *Theor. Comput. Sci.* 81.2 (1991), pp. 317–324. DOI: 10.1016/0304-3975(91)90200-L. URL: https://doi.org/10.1016/0304-3975(91)90200-L (cit. on pp. 5, 40).

[MT98] Jochen Meßner and Jacobo Torán. "Optimal Proof Systems for Propositional Logic and Complete Sets". In: *STACS 98*. Vol. 1373. Lecture Notes in Computer Science. Springer, 1998, pp. 477–487. DOI: 10.1007/BFB0028583. URL: https://doi.org/10.1007/BFb0028583 (cit. on p. 7).

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges". In: *CRYPTO 2003*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 96–109. DOI: 10.1007/978-3-540-45146-4_6. URL: https://doi.org/10.1007/978-3-540-45146-4%5C_6 (cit. on pp. 3, 6, 11).

[Pap94] Christos H. Papadimitriou. "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence". In: *J. Comput. Syst. Sci.* 48.3 (1994), pp. 498–532. DOI: 10.1016/S0022-0000(05)80063-7. URL: https://doi.org/10.1016/S0022-0000(05)80063-7 (cit. on p. 10).

[Pas03a] Rafael Pass. "On Deniability in the Common Reference String and Random Oracle Model". In: *CRYPTO 2003*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 316–337. DOI: 10.1007/978-3-540-45146-4_19. URL: https://doi.org/10.1007/978-3-540-45146-4%5C_19 (cit. on p. 5).

[Pas03b] Rafael Pass. "Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition". In: *EUROCRYPT 2003*. Vol. 2656. Lecture Notes in Computer Science. Springer, 2003, pp. 160–176. DOI: 10.1007/3-540-39200-9_10. URL: https://doi.org/10.1007/3-540-39200-9%5C_10 (cit. on p. 10).

[PS19] Ján Pich and Rahul Santhanam. "Why are Proof Complexity Lower Bounds Hard?" In: *FOCS 2019*. IEEE Computer Society, 2019, pp. 1305–1324. DOI: 10.1109/FOCS.2019.00080. URL: https://doi.org/10.1109/FOCS.2019.00080 (cit. on p. 7).

[Pud06] Pavel Pudlák. "Gödel and computations: a 100th anniversary retrospective". In: *SIGACT News* 37.4 (2006), pp. 13–21. DOI: 10.1145/1189056.1189058. URL: https://doi.org/10.1145/1189056.1189058 (cit. on p. 7).

[Pud13] Pavel Pudlák. *Logical Foundations of Mathematics and Computational Complexity - A Gentle Introduction*. Springer monographs in mathematics. Springer, 2013. ISBN: 978-3-319-00118-0. DOI: 10.1007/978-3-319-00119-7. URL: https://doi.org/10.1007/978-3-319-00119-7 (cit. on pp. 7, 21, 22).

[Pud17] Pavel Pudlák. "Incompleteness in the finite Domain". In: *Bull. Symb. Log.* 23.4 (2017), pp. 405–441. DOI: 10.1017/BSL.2017.32. URL: https://doi.org/10.1017/bsl.2017.32 (cit. on p. 21).

[Pud86] Pavel Pudlák. "On the length of proofs of finitistic consistency statements in first order theories". In: *Logic Colloquium '84*. Vol. 120. Studies in Logic and the Foundations of Mathematics. Elsevier, 1986, pp. 165–196. DOI: https://doi.org/10.1016/S0049-237X(08)70462-2 (cit. on pp. 7, 16, 21, 22, 37, 42).

[Rab05] Michael O. Rabin. "How To Exchange Secrets with Oblivious Transfer". In: *IACR Cryptol. ePrint Arch.* (2005), p. 187. URL: http://eprint.iacr.org/2005/187 (cit. on p. 14).

[Rog06] Phillip Rogaway. "Formalizing Human Ignorance: Collision-Resistant Hashing without the Keys". In: *IACR Cryptol. ePrint Arch.* (2006), p. 281. URL: http://eprint.iacr.org/2006/281 (cit. on p. 23).

[Rud97] Steven Rudich. "Super-bits, Demi-bits, and NP/qpoly-natural Proofs". In: *RANDOM'97*. Vol. 1269. Lecture Notes in Computer Science. Springer, 1997, pp. 85–93. DOI: 10.1007/3-540-63248-4_8. URL: https://doi.org/10.1007/3-540-63248-4%5C_8 (cit. on p. 33).

[Sad02] Zenon Sadowski. "On an optimal propositional proof system and the structure of easy subsets of TAUT". In: *Theor. Comput. Sci.* 288.1 (2002), pp. 181–193. DOI: 10.1016/S0304-3975(01)00155-4. URL: https://doi.org/10.1016/S0304-3975(01)00155-4 (cit. on p. 7).

[Sad07] Zenon Sadowski. "Optimal Proof Systems, Optimal Acceptors and Recursive Presentability". In: *Fundam. Informaticae* 79.1-2 (2007), pp. 169–185. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi79-1-2-08 (cit. on p. 7).

[Sad08] Zenon Sadowski. "Optimal Proof Systems and Complete Languages". In: *Electron. Colloquium Comput. Complex.* TR08-107 (2008). ECCC: TR08-107. URL: https://eccc.weizmann.ac.il/eccc-reports/2008/TR08-107/index.html (cit. on p. 7).

[Tur37] Alan M. Turing. "On computable numbers, with an application to the Entscheidungsproblem". In: *Proc. London Math. Soc.* s2-42.1 (1937), pp. 230–265. DOI: 10.1112/PLMS/S2-42.1.230. URL: https://doi.org/10.1112/plms/s2-42.1.230 (cit. on p. 21).

[Wik24] Wikipedia contributors. *Proof complexity — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Proof_complexity. [Online; accessed 12-April-2025]. 2024. URL: https://en.wikipedia.org/wiki/Proof_complexity (cit. on p. 6).

[Yao82] Andrew C. Yao. "Theory and Applications of Trapdoor Functions". In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1982, pp. 80–91 (cit. on p. 38).

A "Universal" Non-Interactive Witness Hiding?

We conjecture there is no "universal" non-interactive witness hiding prover.

Conjecture A.1 (No Universal Witness Hiding Conjecture)**.** *For every (uniform) perfectly sound prover $\mathcal{P}$, there exists a hard* Search-SAT *distribution $\mathcal{D}$ for which $\mathcal{P}$ is not witness hiding.*

Our intuition comes from obfuscation and follows previous impossibility results [BGI+12]. It seems likely we are not the first to consider the following argument, but we could not find it in the literature.

Imagine a distribution $\mathcal{D}$ that outputs tuples of the form $(\varphi = \varphi_1 \vee \varphi_2, w)$ where

- w is a satisfying assignment to φ_1,
- given just φ_1, it is hard to find a satisfying assignment to φ_1,
- φ_2 is a trivially unsatisfiable formula, but its description encodes an obfuscated Turing machine M with the following behavior: given a proof π of "φ is satisfiable" that $\mathcal{P}$.verify accepts, it outputs w.

Because of the last property, $\mathcal{P}$ cannot be witness hiding for $\mathcal{D}$. If M is "sufficiently obfuscated," one could hope that $\mathcal{D}$ is a hard Search-SAT distribution.

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