On the Impossibility of SNARGs with Short CRS

(*or: Revisiting Gentry-Wichs Barrier in the Non-adaptive Setting*)

Liyan Chen[1] and Zhengzhong Jin[2]

[1]MIT
[2]Northeastern University

June 2, 2026

Abstract

We study the inherent barriers to constructing *non-adaptively* sound succinct non-interactive arguments (SNARGs) for NP with a CRS whose length is sublinear in the witness length. Our results cover both the standard SNARGs and SNARGs with an additional updatable feature (i.e. incrementally verifiable computation for NP).

- For updatable SNARGs, we show a black-box separation from falsifiable assumptions for *uniform* polynomial-time reductions, assuming sub-exponential hardness of learning with error.
- For general SNARGs, we show a black-box separation from falsifiable assumptions for *non-uniform* polynomial-time reductions that only non-adaptively make an instance-size-independent number of queries to the adversary, assuming the existence of sub-exponentially secure super-bit generators. We observe that all known SNARG constructions from polynomial hardness of standard assumptions have 1-query soundness reductions. Thus, our result complements existing constructions.

Previously, the seminal work [Gentry-Wichs, STOC'11] showed a black-box separation of SNARGs from falsifiable assumptions in the *adaptive* soundness setting. We explore whether any barriers exist in the non-adaptive setting. To obtain our result, we derive a simulation lemma for unbounded polynomial-length auxiliary inputs assuming super-bit generators.

Contents

1 Introduction **1**
- 1.1 Our Results . . . 2
- 1.2 More Related Works . . . 4

2 Technical Overview **5**
- 2.1 Recap of Gentry-Wichs Barrier . . . 5
- 2.2 Impossibility for Updatable SNARGs . . . 5
- 2.3 Impossibility for General SNARGs . . . 11
- 2.4 One-query Nature of Existing SNARG Reductions . . . 12
  - 2.4.1 Sahai-Waters SNARGs . . . 13
  - 2.4.2 Hash-and-BARG SNARGs . . . 13

3 Preliminary **14**
- 3.1 Leakage Simulation Lemma . . . 15
- 3.2 Incrementally Verifiable Computation (IVC) for NP . . . 15
- 3.3 Succinct Non-interactive Arguments (SNARGs) . . . 16
- 3.4 Falsifiable Cryptographic Assumptions . . . 17
- 3.5 Black-Box Reductions . . . 17

4 Black-box Reduction Lower-bound for IVC **18**
- 4.1 OR-Homomorphic Encoding . . . 18
- 4.2 A Simulatable Adversary . . . 20
- 4.3 Proof of Theorem 4.3 . . . 21

5 A Lower-bound of Black-box Non-uniform Reductions for SNARG **26**
- 5.1 Cryptography Against Nondeterministic Adversary . . . 27
- 5.2 A Simulatable Adversary Fooling Query-bounded Distinguishers . . . 30
- 5.3 Proof of Theorem 5.9 . . . 31

1 Introduction

Succinct non-interactive arguments (SNARGs) allow a prover to convince a verifier of the truth of a statement through a short, single-round message. The standard setting for SNARGs is the *common reference string* (*CRS*) model, where a trusted setup algorithm generates a CRS and shares it with both the prover and the verifier. SNARGs have a wide range of applications, including delegation of computation [GKR15], verifying distributed computation [Val08; CT10], private information retrieval [BDKP22], zero-knowledge [Kiy22], blockchain [BCG+14]. Beyond cryptography, it has been used to establish the cryptographic hardness of PPAD [CHK+19; KPY20; JKKZ21; KLV23].

Following the seminal work of Micali [Mic00], which constructed SNARGs in the random oracle model, a long line of research [KRR13; KRR14; BHK17; BKKSW18; KPY19; CJJ21; KVZ21; CJJ22; HJKS22; WW22; KLVW23; CGJJZ23] has aimed to construct SNARGs from well-studied, falsifiable hardness assumptions [Nao03]. Notably, it is known how to build SNARGs for NP under well-studied cryptographic assumptions such as indistinguishability obfuscation [SW21; WW24b; WZ24; WW24a] or witness encryption with polynomial-size Extended Frege proof of correctness [JKLM25]. However, all prior constructions of SNARGs for NP from well-studied assumptions have a long CRS whose length grows with the NP witness size. This leads to the following question:

Is there any fundamental barrier to constructing SNARGs for NP *from standard assumptions, with a short CRS whose length is sublinear in the witness size?*

A short CRS is a natural requirement of succinctness in SNARGs. In practice, SNARGs typically require a distributed "ceremony" protocol to generate the CRS. A shorter CRS can reduce the cost of this setup phase and also improve the efficiency of the prover and verifier.

Gentry-Wichs Barrier. Previously, the seminal work by Gentry and Wichs [GW11] showed that SNARGs for NP cannot be constructed from any falsifiable assumptions, assuming a black-box soundness reduction and the SNARG is adaptively sound. As observed in [CGKS23], if the reduction preserves the security parameter of the underlying falsifiable assumption, the Gentry–Wichs impossibility result also applies in the long CRS setting. However, Gentry-Wichs impossibility result crucially relies on *adaptive soundness*, where the cheating prover may choose the instance after observing the CRS.

This Work: Non-adaptive Soundness. In this work, we investigate whether any inherent barrier still exists in the *non-adaptive* soundness setting. In this setting, the CRS is generated independently of the instance selected by the adversary. This is a weaker notion than adaptive soundness, as it is implied by the latter. Nevertheless, non-adaptive soundness suffices for many applications in which the instance is chosen independently of the CRS. Moreover, non-adaptive soundness can be generically upgraded to adaptive soundness using complexity leveraging, although this comes at the cost of increasing the proof length with the instance size.

This leads to the central question of this work:

Is there an inherent barrier to constructing SNARGs for NP *with a short CRS that satisfy non-adaptive soundness from falsifiable assumptions?*

Proving black-box separations in the non-adaptive setting faces several critical challenges, which stem from the following distinctions between non-adaptive and adaptive soundness reductions. In the non-adaptive setting, since the adversary chooses the instance before the CRS is generated, the reduction can run the adversary multiple times with different CRS on the same instance. This reduction strategy resembles *rewinding*, which is typically not possible in the adaptive setting. Moreover, in the non-adaptive setting, the reduction can exploit non-uniform advice that arbitrarily depends on the instance, as shown in [JKLV24], whereas in the adaptive setting, this is impossible since the instances are chosen adaptively. Without these two challenges, Gentry-Wichs [GW11] can extend directly to the non-adaptive setting for 1-query, uniform reductions.

Our work investigates the inherent barriers for both the standard SNARG notion and SNARGs with an additional feature—updatability.

Updatable SNARGs (i.e. IVC for NP). Updatable SNARGs are defined in the following setting. Consider a nondeterministic computation $\mathsf{st}_0 \to \mathsf{st}_1 \to \mathsf{st}_2 \to \ldots \to \mathsf{st}_T$, where in the i-th step, one runs a nondeterministic machine with some witness w_i on the current state st_i and produces the next state st_{i+1}. We say a SNARG is *updatable*, if given the initial state st_0, a state st_i, a proof π_i that st_i is reachable via i-step nondeterministic computation, and the i-th witness w_i, the prover can efficiently transform π_i into a proof π_{i+1} that the state st_{i+1} is reachable via a $(i+1)$-step computation, without recomputing everything from scratch.

This notion was initially introduced for proving deterministic computation and is known as *incrementally verifiable computation* (*IVC*) [Val08]. Recently, IVC has been implicitly extended to support nondeterministic computation in concretely efficient IVC schemes [KST22; KS22; KS23; KS24], and the recent work [DJJ+25] constructed IVC for NP with CRS length growing with the single-step witness from standard assumptions.

In the broader context, the updatable feature of SNARGs has been extensively studied for verifying distributed computation. Indeed, it can be viewed as a special case of proof-carrying data [CT10] and recursive composition [BCCT13]. These techniques are particularly attractive for concretely efficient SNARG constructions (e.g. [BGH19; BCMS20]), as their prover preserves the time and space complexity of the underlying computation, and is a common "bootstrap" technique to reduce the CRS size [BCCT13]. The impossibility we obtain in the updatable setting also implies the impossibility of those more general settings.

In this work, we focus on updatable SNARGs whose CRS length and proof length are both succinct, smaller than the intermediate-state size or the length of single-step witnesses. We consider non-adaptive soundness for updatable SNARGs, where the initial state and the final state of the computation must be chosen independently of the CRS.

1.1 Our Results

We consider both standard SNARGs and SNARGs with an additional, updatable feature. We begin with the updatable SNARG setting (IVC for NP):

Theorem 1.1 (Impossibility for Updatable SNARGs, Informal Version of Theorem 4.5). Assuming sub-exponential hardness of Learning with Errors, there does not exist an updatable SNARG for NP (i.e. IVC for NP) that satisfies non-adaptive soundness, has a CRS whose length is independent of the length of the instance and witness, and has a polynomial-time black-box *uniform* soundness reduction to falsifiable assumptions. ♢

Remark 1 (CRS and Proof Lengths v.s. Instance Length). Similar to Gentry-Wichs, our impossibility naturally extends to the setting where CRS length and proof length depends on the instance length, by assuming stronger assumptions. In both this work and Gentry-Wichs, the requirement is that $2^{O(|\mathsf{crs}|+|\pi|)}$ needs to be less than the hardness of the language, i.e. an adversary running in time $2^{O(|\mathsf{crs}|+|\pi|)}$ should not distinguish yes instances from no instances. Therefore, if the language has exponential hardness, then we rule out $|\mathsf{crs}|, |\pi| = o(n)$ (n is instance size); if the language has sub-exponential hardness, then we rule out $|\mathsf{crs}|, |\pi| = O(n^c)$ for some $c \in (0, 1)$. ♢

Falsifiable assumptions [Nao03] are assumptions that can be formulated as an interactive game between a challenger and an adversary. In this framework, the challenger generates a challenge, and the adversary attempts to solve it. The assumption is considered falsifiable, if the challenger runs in polynomial-time—so any successful adversary serves as a piece of evidence against the assumption. All standard cryptographic assumptions—such as factoring or Learning With Errors—are captured by falsifiable assumptions.

Non-Uniform Reduction. Our Theorem 1.1 focuses on uniform soundness reductions. We then turn to the question of whether any inherent barrier remains when allowing *non-uniform reductions*. In this setting, the reduction algorithm is allowed to take non-uniform advice—an arbitrary string that may depend on the instance, but independent of the adversary. Indeed, recent work [JKLV24; JKLM25] has shown that non-adaptively sound SNARGs can be achieved by using non-uniform reductions for some subclasses of NP from standard assumptions. In the setting of non-uniform reductions, we consider general SNARGs and obtain the following theorem:

Theorem 1.2 (Impossibility for *General* SNARGs, Informal Version of Theorem 5.10). Assuming the existence of a sub-exponentially secure super-bit generator, there does not exist a SNARG for NP that satisfies non-adaptive soundness, has a polynomial-time black-box *non-uniform* soundness reduction to falsifiable assumptions that only makes non-adaptive query to the adversary, whose CRS size, proof size, and reduction query complexity are independent of the length of the instance and witness. ♢

Our result relies on the existence of super-bit generators. Rudich introduced the super-bit generator as a natural extension of pseudorandom generators secure against NP adversaries, in the context of the natural proofs barrier [RR97]. The super-bit generator can be viewed as an average-case analog of the widely believed complexity conjecture $\mathsf{coNP} \not\subseteq \mathsf{NP}$. In his original work, Rudich proposed a candidate construction based on the subset-sum problem. Recently, [CL24] proposed another candidate of super-bit generators based on lattice assumptions.

On Query-bounded Reductions. Our Theorem 1.2 considers reduction algorithms with a small number of non-adaptive queries to the adversary. While such soundness reduction may appear limited at first glance, we observe that all known SNARG constructions from polynomial-time standard assumptions, such as [CJJ21; CJJ22; BBKLP23; JKLV24], in fact only rely on *1-query* soundness reductions.

Hence, we view our non-adaptive-query-bounded impossibility result as capturing the limitations of the soundness reduction techniques used in the literature. Namely, our result suggests that, in order to achieve SNARGs for NP with a short CRS and proof from polynomial hardness of falsifiable assumption—even with non-adaptive soundness—one must rely on either a soundness reduction with large query complexity or adaptive queries, or non-black-box reduction.

We defer more detailed discussion of 1-query nature of existing reductions to Section 2.4.

On Polynomial-time Reductions and Prior Works of SNARGs for NP. Both our results Theorem 1.1 and Theorem 1.2 consider reductions runs in polynomial time. The reader may wonder whether a reduction with a much larger runtime could circumvent our impossibility result. We next explain the main barrier to such an approach.

Indeed, in the prior works of SNARGs for NP from indistinguishability obfuscation (iO) or witness encryption [1] [SW21; WW24b; WZ24; WW24a; JKLM25], when iO and witness encryption are instantiated from falsifiable assumptions [JLS21; JLS22; RVV24], the resulting constructions can be interpreted as having a reduction that runs in time exponential in the NP witness length, and thereby circumventing Gentry-Wichs. However, the security of these SNARGs inherently rely on complexity leveraging, which forces the security parameter of the underlying cryptographic assumptions to grow at least with the NP witness length. As a result, if the CRS contains even a single instance of the underlying assumption (e.g. a LWE sample), its size is already large. To take advantage of a large reduction time, one might need to develop new reduction techniques that avoid including even a single instance of the underlying assumption in the CRS.

More Comparison with Gentry-Wichs. The seminal work of Gentry and Wichs [GW11] proved a black-box separation in the adaptive soundness setting, whereas our work focuses on the weaker, non-adaptive setting. Gentry-Wichs considers general SNARGs and allows reductions that can make multiple queries to the adversary. In contrast, our Theorem 1.1 considers updatable SNARGs under uniform reductions, while Theorem 1.2 considers general SNARGs with non-adaptive-query-bounded non-uniform reductions. Additionally, the original Gentry-Wichs impossibility result assumes a short CRS that is a fixed polynomial in the security parameter to handle non-security parameter preserving reductions. But as observed in [CGKS23], Gentry-Wichs also applies in the long CRS setting when the CRS can grow with the NP instance length and witness length. All of our results, by contrast, assume that the CRS length is short.

1.2 More Related Works

Updatable SNARGs (IVC) for P. Although constructing updatable SNARGs for NP remains a hard task, it is well-understood on how to construct updatable SNARGs for P (IVC) [DGKV22; PP22], from standard cryptogrpahic assumptions including learning with errors (LWE), a combination of quadratic residuosity (QR) and decisional Diffie-Hellman (DDH), or the DLIN assumption on bilinear groups.

Other Impossibilities for Updatable SNARGs. [HAN23] considered updatable SNARGs for NP in the random oracle model. They proved impossibility for updatable SNARGs with either 1) zero-knowledge property that involves reprogramming the random oracle; 2) knowledge soundness (and an additional structural assumption).

[1] The witness encryption requires a polynomial-size Extended Frege proof of correctness.

2 Technical Overview

Before diving into the technical details of this work, we briefly recall the Gentry–Wichs impossibility result [GW11], which showed that adaptive soundness of SNARGs cannot be based on falsifiable assumptions via black-box reductions.

2.1 Recap of Gentry-Wichs Barrier

A black-box soundness reduction is an algorithm $\mathcal{R}$ that uses a hypothetical adversary $\mathcal{P}$ as an oracle to break the underlying falsifiable assumption. As a black-box reduction, the algorithm $\mathcal{R}^{\mathcal{P}}$ should break the assumption as long as $\mathcal{P}$ breaks the soundness of the SNARG, regardless of whether $\mathcal{P}$ is efficient.

At a high level, Gentry-Wichs rules out such a black-box reduction for adaptively sound SNARGs via the following "meta-reduction" strategy. They construct a pair of adversaries $\mathcal{P}$ and $\overline{\mathcal{P}}$ where $\mathcal{P}$ is a polynomial-time algorithm that always chooses instances that are in the language and follows the honest prover algorithm, whereas $\overline{\mathcal{P}}$ runs in unbounded time and chooses instances that are not in the language. Moreover, for any CRS, the outputs of the two adversaries $\mathcal{P}$ and $\overline{\mathcal{P}}$ are indistinguishable against any polynomial-time algorithm.

To rule out black-box reductions, they start with the "honest" prover $\mathcal{P}$ that always makes the verifier accept. Then they switch $\mathcal{P}$ to $\overline{\mathcal{P}}$ and argue that $\overline{\mathcal{P}}$ breaks the soundness. This is because $\mathcal{P}$ and $\overline{\mathcal{P}}$ are indistinguishable and hence, $\overline{\mathcal{P}}$ also makes the verifier accept, but chooses instances that are not in the language. Further, the correctness of the reduction implies that $\mathcal{R}^{\overline{\mathcal{P}}}$ breaks the underlying falsifiable assumption. Finally, they switch $\mathcal{R}^{\overline{\mathcal{P}}}$ back to $\mathcal{R}^{\mathcal{P}}$ and argue that $\mathcal{R}^{\mathcal{P}}$ still breaks the underlying assumption, again due to the indistinguishability of $\mathcal{P}$ and $\overline{\mathcal{P}}$. This reaches a contradiction because we find a polynomial-time attack $\mathcal{R}^{\mathcal{P}}$ to break the underlying falsifiable assumption.

Leakage Lemma. At the heart of the above meta-reduction proof strategy is the construction of $\overline{\mathcal{P}}$ whose output is indistinguishable from the honest prover algorithm $\mathcal{P}$. To construct this $\overline{\mathcal{P}}$, Gentry-Wichs proves the following *leakage lemma*. Let $\mathcal{D}, \overline{\mathcal{D}}$ be a pair of "hard" distributions, where $\mathcal{D}$ is over $\mathcal{L}$ and $\overline{\mathcal{D}}$ is over $\overline{\mathcal{L}} = \{0,1\}^* \setminus \mathcal{L}$, and $\mathcal{D}$ and $\overline{\mathcal{D}}$ are indistinguishable against circuits of size T. For any "small leakage" π whose size is much smaller than $\log T$, and π is allow to be arbitrarily correlated to $x \leftarrow \mathcal{D}$, there exists a "simulated" leakage $\overline{\pi}$ correlated with $\bar{x} \leftarrow \overline{\mathcal{D}}$, such that the joint distributions $\{(x, \pi) : x \leftarrow \mathcal{D}\}$ and $\{(\bar{x}, \bar{\pi}) : \bar{x} \leftarrow \overline{\mathcal{D}}\}$ are indistinguishable against polynomial-size circuits.

To apply this lemma, Gentry-Wichs takes the language $\mathcal{L}$ as the image of a pseudorandom generator (PRG) and $\mathcal{D}$ as the distribution of a random PRG image, and $\overline{\mathcal{D}}$ is taken as the uniform distribution over $\overline{\mathcal{D}}$. By the leakage lemma, there exists a simulated proof $\overline{\pi}$ for any "false" instance $\bar{x}$ sampled from $\overline{\mathcal{D}}$. Then the outputs of the honest $\mathcal{P}$ and a $\overline{\mathcal{P}}$ that outputs $\overline{\pi}$ are computationally indistinguishable as desired. This completes the overview of Gentry-Wichs impossibility for adaptively sound SNARGs.

2.2 Impossibility for Updatable SNARGs

Challenge: Ruling Out "Rewinding" Reductions. The proof strategy in Gentry-Wichs works for adaptive soundness. However, it is unclear how to extend the idea to the non-adaptive sound-

ness setting. In this setting, after the adversary chooses the instance, the reduction algorithm can send different CRS queries to the adversary and obtain multiple SNARG proofs for the *same* instance. The reduction algorithm could potentially extract an entire witness from the adversary and then prove the soundness by a proof of contradiction. This resembles the rewinding technique in multi-round interactive proof soundness proofs. [2] To extend Gentry-Wichs to the non-adaptive soundness setting, we need to rule out such a rewinding proof strategy.

Blueprint of Our Proof. Our intuition to rule out the above rewinding reduction strategy is that, for a general NP language, if an instance has multiple witnesses, then the rewinding strategy seems unlikely to work——if an honest prover chooses different witnesses based on different CRS each time, then it's unclear how to "stitch" the information from different SNARG proofs to form a consistent NP witness in the reduction.

Following this intuition, our blueprint for proving impossibility in the non-adaptive soundness setting is as follows. We first choose an NP language $\mathcal{L}$ where each instance in $\mathcal{L}$ has multiple NP witnesses. Then we construct a polynomial-time algorithm $\mathcal{P}$ for an instance $x \in \mathcal{L}$ sampled from some distribution: Given any CRS queried by the reduction, $\mathcal{P}$ randomly chooses a witness of x and uses the honest prover algorithm to generate an SNARG proof as its output. Next, we use a series of hybrid arguments to "gradually" switch from $\mathcal{P}$ to an unbounded-time adversary $\overline{\mathcal{P}}$ for an instance $\bar{x} \notin \mathcal{L}$, such that any polynomial-time oracle-algorithm D cannot distinguish $D^{\mathcal{P}}$ and $D^{\overline{\mathcal{P}}}$. The number of hybrids is the same as the number of NP witnesses for x. In each hybrid, we "puncture" one witness from x, and replace the honestly generated SNARG proof of this witness with a simulated proof using the leakage lemma.

Specifically, in the i-th hybrid, the instance is switched from the initial x to an instance $x^{(i)}$ where the first i witnesses have already been removed in $x^{(i)}$. The unbounded-time adversary $\mathcal{P}^{(i)}$ in this hybrid behaves almost the same as $\mathcal{P}$, except that whenever the witness it samples falls in the first i witnesses, the adversary $\mathcal{P}^{(i)}$ answers the query from the reduction using a simulated proof via the leakage lemma. In this way, we can switch from an honest prover $\mathcal{P}$ for some $x \in \mathcal{L}$ to some unbounded prover $\overline{\mathcal{P}}$ for some $\bar{x} \notin \mathcal{L}$.

Next, we describe how to instantiate the above blueprint.

Towards Constructing NP Languages with Multiple Witnesses. Our initial attempt is to take a PRG G and let the language $\mathcal{L}$ be

$$\mathcal{L} = \{\mathbf{x} = (x_1, \ldots, x_m) \mid \exists i \in [m] : x_i \in \text{Image}(G)\}\,.$$

The NP witness is an index i and the PRG seed for x_i. If we sample all $x_1, \ldots, x_m$ from the PRG image, then we have m PRG seeds as the witnesses. However, this language does not meet our needs directly since the number of CRS queries made by the reduction may depend on the length of the instance or the NP verification circuit size. If the reduction learn the value m and makes more than m queries, then our intuition in the above blueprint does not hold because there will be many collisions in the random sampling of the witnesses, and one witness may be used multiple times to generate the proof. Now the reduction can still potentially extract an entire witness.

[2] Indeed, for language in the complexity class UP, it seems plausible to use such a rewinding strategy to recover the entire witness from a falsifiable assumption. However, how to build SNARK (or SNARG) for UP from standard polynomial-time falsifiable assumptions remains an interesting open question, despite the recent construction from evasive LWE [MPV24].

To fix this issue, we want to construct an NP language where the number of witnesses can be an arbitrarily large polynomial of the instance length (and more generally, the NP verification circuit size). To achieve this, our next idea is to apply a Merkle hash to further compress the PRG images. Namely, the language now looks like the following:

$$\mathcal{L} = \left\{ \mathbf{x} \;\middle|\; \begin{array}{c} \text{There exists an index } i\text{, a PRG image } x_i, \\ \text{and a local opening for the } i\text{-th leaf with respect to the Merkle hash root } \mathbf{x}. \end{array} \right\}.$$

Recall that, the local opening of a Merkle hash tree consists of the root-to-leaf path for a particular leaf in the Merkle tree. For this language, if we sample $x_1, \ldots, x_m$ from the PRG image and let $\mathbf{x} = \mathsf{MerkleHash}(x_1, \ldots, x_m)$, then we have m witnesses, and the verification circuit size of this NP relation is essentially independent of the number of witnesses m.

However, this construction is still problematic: the complement of $\mathcal{L}$ may be empty! Since the hash function is compressing, it is likely that every $\mathbf{x}$ has some preimage $(x_1, \ldots, x_m)$ where one of them is in the image of the PRG. To avoid such a situation, we need to guarantee that, if all of $x_1, \ldots, x_m$ are non-images, then the resulting hash $\mathsf{MerkleHash}(x_1, \ldots, x_m)$ must fall outside of $\mathcal{L}$.

Our Hard NP Language from FHE. We achieve this property using fully-homomorphic encryption (FHE). Specifically, we take $x_1, \ldots, x_m$ as FHE ciphertexts where each of them encodes a bit, and replace the PRG image with the FHE ciphertexts of 1's. Then we replace the hash function in the Merkle hash tree with the homomorphic evaluation of the OR function. Now the polynomial-time prover $\mathcal{P}$ in our blueprint uses an instance $\mathbf{x}$ obtained by homomorphically evaluating the OR function over $x_1, \ldots, x_m$ where each x_i is an FHE encryption of 1. The language $\mathcal{L}$ now looks like the following[3]:

$$\mathcal{L} = \left\{ (\mathsf{pk}, \mathbf{x}) \;\middle|\; \begin{array}{c} \text{There exists a root-to-leaf path of homomorphic-OR Merkle Tree,} \\ \text{and there exists some random coins to certify that the leaf is an FHE encryption of } 1. \end{array} \right\}$$

Clearly, $\mathcal{L}$ only consists of encryption of 1's, since if one of the leaf nodes is an encryption of 1, then the resulting $\mathbf{x}$ must also be an encryption of 1 due to the correctness of homomorphic evaluation along the root-to-leaf path. Looking ahead, when instantiating our blueprint, we will use m hybrid arguments to switch $x_1, \ldots, x_m$ one-by-one from encryption of 1's to encryption of 0's. In the end, the instance $\mathbf{x}$ is switched to an instance $\overline{\mathbf{x}}$ that is obtained by homomorphically evaluating OR over a bunch of encryptions of 0's, which also results in an encryption of 0. Then the last instance in the hybrid $(\mathsf{pk}, \overline{\mathbf{x}})$ for the unbounded-time prover $\overline{\mathcal{P}}$ is not in the language $\mathcal{L}$. This meets what we need.

Leakage Simulation for SNARG Proofs. As mentioned above, we use hybrid arguments to switch $x_1, \ldots, x_m$ from encryption of 1's to encryption of 0's. Note that the distinguisher we want to fool needs to run the reduction algorithm internally, which can issue CRS queries to learn many SNARG proofs. In the honest case where all of $x_1, \ldots, x_m$ are encryption of 1's, for every x_i, the honest proof that uses random coins of x_i can be generated using the rest of $x_1, \ldots, x_m$. During the hybrid arguments, in the j-th hybrid, we want to switch x_j from "1" to "0", and its prefix $x_1, \ldots, x_{j-1}$ are already switched to encryption of 0's. The distinguisher for the leakage simulation of x_j needs to

[3] In the technical part, we abstract the usage of FHE as a primitive called OR-homomorphic encoding. This helps our later discussion.

take all $x_1, \ldots, x_m$ as input[4], and it internally simulates the proofs for $x_1, \ldots, x_{j-1}$ and then runs the reduction algorithm.

Since we need an explicit leakage simulator in the distinguisher, we rely on the simulators in [VZ13; CCL18]. They guarantee that, given a pair of "hard" distributions $\mathcal{D}, \overline{\mathcal{D}}$, for any leakage of length $|\pi|$ jointly distributed with $\mathcal{D}$, there exists an algorithm that computes the corresponding leakage for input sampled from $\overline{\mathcal{D}}$. This leakage simulator runs in time $2^{\Theta(|\pi|)} \cdot |D|$ where $|D|$ is the size of the distinguisher.

For now, let's assume we have a fixed CRS for simplicity, so we only need to simulate proofs $\pi_1, \ldots, \pi_m$ for $x_1, \ldots, x_m$ respectively. Applying the above explicit leakage simulator, a naïve idea is the following: let S_j be the simulator that generates $\bar{\pi}_j$ against distinguishers that also runs $S_1, \ldots, S_{j-1}$ to generate $\bar{\pi}_1, \ldots, \bar{\pi}_{j-1}$. The problem is, the simulator S_j for $\bar{\pi}_j$ now runs in time $2^{|\pi|} \cdot \Omega(|S_{j-1}|)$, since the distinguisher for S_j needs to run S_{j-1} internally. In the end, the last simulator S_m runs in time $2^{\Omega(m|\pi|)}$, which is insufficient for us, since $m|\pi|$ can be larger than the hardness of the language.

Ideally, we want each of $S_1, \ldots, S_m$ runs in time $2^{O(|\pi|)}$. To achieve so, the potential distinguisher should not simulate proofs by itself, but take those simulated proofs directly as input. This brings the concern of *what input* should be taken by the simulators. The leakage simulators needs to take the same input as potential distinguishers to run them internally, that is, simulators need to take previously simulated proofs, for example, S_j needs to take outputs of $S_1, \ldots, S_{j-1}$. Unfortunately, this creates a circularity issue: recall that, every S_i needs to take all later proofs $\pi_{i+1}, \ldots, \pi_m$ to answer reduction algorithm's queries, but now to generate $\bar{\pi}_j$ (for $j > i$), we needs to run S_i first. The simulators depends on the outputs of each other, and we don't even know where to start.

Updatability to the Rescue. We rely on the updatability of SNARGs to get around the current issue. Updatable SNARGs for NP is defined with respect to NP machines, where in each computation step it reads an NP advice and modifies its internal state. Given the machine description, its internal state, a proof for the computation up to this state, and the advice taken by the next computation step, updatable SNARGs allows to generate the proof for the next internal state.

Our hard NP language can be written as an NP machine as follows:

- The internal state contains a Merkle tree node and public parameter (FHE public key).
- In the first computation step, the machine reads (from the NP advice) a leaf node x_i and random coins to certify that x_i is an encryption of 1.
- In the ensuing computation steps, the machine reads (from the NP advice) the root-to-leaf path from x_i from bottom to top. Each time it reads the sibling of the current node, and computes the parent of these two nodes to update its state.
- The final internal state would contain the root node (i.e. instance $\mathtt{x}$), and pk.

The updatability of SNARGs gives the following property: for any index i, there exists a polynomial-time algorithm F_i that, given as input the i-th SNARG proof π_i for the leaf node x_i (corresponding to the first step of computation) together with all leaf nodes $x_1, \ldots, x_m$, outputs a SNARG proof for

[4]In fact, the distinguisher also needs to know the witness for $x_{j+1}, \ldots, x_m$ or their honestly generated proofs to answer the queries from the reduction. This can be achieved by adding them to the input of the simulator and the distinguisher.

the root node (i.e. final state). Concretely, F_i constructs the root-to-leaf path for the i-th leaf using $x_1, \ldots, x_m$, and then applies the update algorithm to update π_i to an SNARG for the root node.

Given this additional property, we apply leakage simulators as follows. For now, let's assume the CRS is fixed (We will deal with this problem later). For leaf nodes $x_1, \ldots, x_m$, our plan is to simulate first-step proofs $\pi_1, \ldots, \pi_m$ for each of them. Fully SNARG proofs can be computed by applying F_i to π_i and $x_1, \ldots, x_m$. We stress that π_i can be honestly computed merely from pk and random coins of x_i without other leaf nodes.

To illustrate our idea, consider the case where all but $\bar{x}_1$ (we use this notation to emphasize $\bar{x}_1$ encrypts 0) are encryption of 1's, and we want to simulate proof $\bar{\pi}_1$ for $\bar{x}_1$. The goal of simulation is to fool the reduction algorithm, which takes $(\mathsf{pk}, x_1, \pi_1, \ldots, x_m, \pi_m)$ (or $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, x_m, \pi_m)$) and performs some (efficient) computation.

Our core observation is it suffices to consider distinguishers of the following form: an efficient algorithm that only takes $(\mathsf{pk}, x_1, \pi_1)$ as input. The reason is we can think of distinguisher that samples $x_2, \ldots, x_m$ and computes $\pi_2, \ldots, \pi_m$ *by itself*, and runs the reduction algorithm. We can do this because we assume π_1 only depends on pk, x_1, and in particular conditioned on pk, (x_1, π_1) is independent of (x_j, π_j) (for $j \geq 2$). Therefore, even conditioned on $(\mathsf{pk}, x_1, \pi_1)$, the rest input $(x_2, \pi_2, \ldots, x_m, \pi_m)$ can be efficiently sampled[5]. Given the input format of disitnguisher, it suffices to design a leakage simulator algorithm that only reads (pk, x_1) and simulates the proof as $\tilde{\pi}_1 = \mathsf{Sim}(\mathsf{pk}, x_1)$, such that $(\mathsf{pk}, x_1, \pi_1)$ and $(\mathsf{pk}, x_1, \tilde{\pi}_1 = \mathsf{Sim}(\mathsf{pk}, x_1))$ are indistinguishable. Then by semantic security of FHE we know $(\mathsf{pk}, x_1, \pi_1)$ is indistinguishable from $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1 = \mathsf{Sim}(\mathsf{pk}, \bar{x}_1))$. Such leakage simulation algorithm is known and runs in time $2^{O(|\pi_1|)}$.

In later hybrids where $\bar{x}_1, \ldots, \bar{x}_i$ are encryption of 0's, we consider distinguishers that read $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_{i-1}, \bar{\pi}_{i-1}, \bar{x}_i, \bar{\pi}_i)$ (or $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_{i-1}, \bar{\pi}_{i-1}, x_i, \pi_i)$) and sample $x_{i+1}, \ldots, x_m$ by itself. Similarly, it suffices to have a leakage simulator that reads $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, x_i)$ and simulates $\tilde{\pi}_i \leftarrow \mathsf{Sim}(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, x_i)$. Since $(\bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_{i-1}, \bar{\pi}_{i-1})$ can be generated in $2^{O(|\pi|)}$ time from pk, by semantic security of FHE,

$$(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, x_i) \approx_c (\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_i),$$

and by leakage simulation

$$(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, x_i, \pi_i) \approx_c (\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_i, \bar{\pi}_i \leftarrow \mathsf{Sim}(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_i)).$$

Continuing this argument allows us to change $(\mathsf{pk}, x_1, \pi_1, \ldots, x_m, \pi_m)$ into $(\mathsf{pk}, \bar{x}_1, \bar{\pi}_1, \ldots, \bar{x}_m, \bar{\pi}_m)$ in an indistinguishable way.

Dealing with CRS. Previously we assume the CRS is fixed. In general, the reduction algorithm makes various CRS queries, and our leakage simulator needs to simulate proofs for all possible CRS, i.e. $\{\pi_i^{\mathsf{crs}}\}_{\mathsf{crs}}$ for each x_i. This leads to a problem: we need to simulate $2^{|\mathsf{crs}|}$-many SNARG proofs for every x_i, but we need to simulate them in time $2^{O(|\mathsf{crs}|)}$ as otherwise we cannot argue indistinguishability from FHE security. Recall that to simulate a general leakge of length ℓ, it takes $2^{\Omega(\ell)}$ time. Unfortunately, in this case ℓ is at least $2^{|\mathsf{crs}|}$.

The reason that $\{\pi_i^{\mathsf{crs}}\}_{\mathsf{crs}}$ can be simulated in much shorter time than a general leakage, is our NP language has many witnesses. The intuition is, the reduction algorithm only uses at most one

[5]This does not hold in the general case, where π_1 depends on all of $x_1, \ldots, x_m$ and an efficient conditional sampling is not possible.

proof in $\{\pi_i^{\mathsf{crs}}\}_{\mathsf{crs}}$, since on each crs query, $\mathcal{P}$ can randomly pick x_i and generates the proof from π_i^{crs}. As long as m is much larger than the number of queries made by the reduction algorithm, with high probability $\mathcal{P}$ "uses" every x_i at most once. In our construction, we take number of leaves m significantly larger than the running time of reduction algorithm, and design $\mathcal{P}$ to randomly select a leaf x_i on each crs query, computes the SNARG proof from π_i^{crs} and the root-to-leaf path to x_i.

How can at most one of $\{\pi_i^{\mathsf{crs}}\}$ is used by helpful? Consider any intermediate hybrid step that we want to switch x_i to $\bar{x}_i$. The naive (and wrong) idea is to simualte $\{\pi_i^{\mathsf{crs}}\}_{\mathsf{crs}}$ such that

$$(\mathsf{pk}, x_1, \{\pi_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \ldots, x_i, \{\pi_i^{\mathsf{crs}}\}) \approx_c (\mathsf{pk}, x_1, \{\pi_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \ldots, x_i, \{\tilde{\pi}_i^{\mathsf{crs}}\} \leftarrow \mathsf{Sim}).$$

Such leakage simulation cannot be achieved in affordable time. The above argument shows the indistinguishability guarantee above is *too strong*: it suffices to fool distinguishers that only access at most one of $\{\pi_i^{\mathsf{crs}}\}$. In particular, by complexity leveraging over all possible crs, it suffices to have for every possible crs,

$$(\mathsf{pk}, x_1, \{\pi_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \ldots, x_i, \pi_i^{\mathsf{crs}}) \approx_c (\mathsf{pk}, x_1, \{\pi_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \ldots, x_i, \tilde{\pi}_i^{\mathsf{crs}} \leftarrow \mathsf{Sim}).$$

Now we only need leakage simulators that read $(\mathsf{pk}, x_1, \{\pi_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \ldots, x_i)$ and simulate π_1^{crs}, for every possible crs. The running time of any such leakage simulator is at most $2^{O(|\mathsf{crs}|+|\pi|)}$, since the length of leakage is now only $|\pi|$.

In our final construction, the $2^{|\mathsf{crs}|}$-many first-step proofs $\{\bar{\pi}_i^{\mathsf{crs}}\}_{\mathsf{crs}}$ for x_i is simulated *separately*, with total simulation time $2^{O(|\mathsf{crs}|+|\pi|)}$. This is the reason our result also needs short crs, since we need FHE security against algorithms running in time $2^{O(|\mathsf{crs}|+|\pi|)}$.

Summary of Our Full Proof. We build the following polynomial-time prover $\mathcal{P}$'s strategy:

1. Take all leaves as encryption of 1, and compute the Merkle root as the instance.
2. Given any CRS, $\mathcal{P}$ randomly selects a leaf node i and honestly computes π_i that leaf i encodes 1. It then updates π_i into a full proof for the Merkle root.

We perform a series of hybrids that switches every leaf node into an encryption of 0, in which $\overline{\mathcal{P}}$ needs to generate simulated proofs for the leaves.

Let us start with the first hybrid: in this hybrid, we need to change the first leaf node x_1 into an encryption of 0, as well as change its leaf-level proof π_1 into $\bar{\pi}_1$. We can assume the first leaf node only provides one leaf-level proofs throughout the reduction algorithm, and enumerate the corresponding CRS of this leaf-level proof. Then, the entire computation of the reduction algorithm can be interpreted as an efficient algorithm that takes $(\mathsf{pk}, x_1, \pi_1)$ as input: it samples remaining leaf nodes as encryptions of 1; on any crs query, $\mathcal{P}$ uses π_1 if x_1 is the selected leaf node, or honestly generates the leaf-level proof otherwise. Therefore, to switch the first leaf from an encryption of 1 x_1 into an encryption of 0 $\bar{x}_1$, and its honest proof π_1 into a fake proof $\bar{\pi}_1$, we only need to ensure that $\{\mathsf{pk}, x_1, \pi_1\} \approx_c \{\mathsf{pk}, \bar{x}_1, \bar{\pi}_1\}$. The simulator reads (pk, x_1) and outputs a simulated proof $\tilde{\pi}_1$ in time $2^{O(|\pi|)}$.

Similar ideas are applied to later steps of hybrids. Again, we enumerate the relevant CRS for the *current* leaf node. Taking the second hybrid as example, we need indistinguishability of

$$(\mathsf{pk}, \bar{x}_1, \{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}, x_2, \pi_2) \approx_c (\mathsf{pk}, \bar{x}_1, \{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \bar{x}_2, \bar{\pi}_2).$$

We construct a leakage simulator that reads $\mathsf{pk}, \bar{x}_1, \{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}, x_2$ and simulates π_2. To prove indistinguishability, it suffices to have

$$(\mathsf{pk}, \bar{x}_1, \{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}, x_2) \approx_c (\mathsf{pk}, \bar{x}_1, \{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}, \bar{x}_2).$$

These two are indistinguishable since they can be efficiently generated from (pk, x_2) and $(\mathsf{pk}, \bar{x}_2)$ respectively: $\bar{x}_1$ can be efficiently sampled from pk, and $\{\bar{\pi}_1^{\mathsf{crs}}\}_{\mathsf{crs}}$ can be generated from $(\mathsf{pk}, \bar{x}_1)$ in $2^{O(|\mathsf{crs}|+|\pi|)}$-time.

For more details, please see Section 4.

2.3 Impossibility for General SNARGs

In this section, we focus on general non-adaptively sound SNARGs, without assuming any additional feature of the prover's algorithm. We outline the techniques in our Theorem 1.2, which rules out non-adaptive-query-bounded non-uniform black-box reductions in this setting.

Non-uniform Reduction and Non-deterministic Computation. In a non-uniform reduction, the reduction algorithm is additionally given a polynomially long advice that arbitrarily depends on the instance. Since the (number-bounded) queries to the adversary can always be simulated if the language is "hard" enough, as shown in Gentry-Wichs' leakage lemma, we can ignore the queries for the moment. Under this view, the reduction algorithm is essentially a nondeterministic algorithm, where the non-uniform advice can be viewed as the NP witness.

To rule out non-uniform reductions, we need to choose a hard language $\mathcal{L}$ that can fool this nondeterministic reduction algorithm. A natural candidate is the super-bit generators introduced in [Rud97] as the analogue of PRG against nondeterministic distinguishers. The security of super-bit generators guarantees that, any *nondeterministic* algorithm accepts a random string with no noticeably higher probability than accepting a random super-bit generator image. Intuitively, this assumption can be viewed as saying "uniformly random strings" do not have NP proofs "on average". Note that in the nondeterministic setting, indistinguishability is *directional*. While we believe the existence of super-bit generator, it is easy to construct a nondeterministic algorithm that accepts a random super-bit generator image with higher probability than a random string: the algorithm just checks if the input is in the image of the super-bit generator. Because of this reason, in the rest of this section we adopt the notation "$X \to Y$ are indistinguishable" (which means X cannot be accepted with noticeably higher probability than Y) rather than "X, Y are indistinguishable" to stress the order between distributions X, Y.

We can let the hard language $\mathcal{L}$ be the set of the images of a super-bit generator. However, the security guarantee of the super-bit generator cannot be directly applied in our setting, since so far we have ignored the queries to the adversary. To take the queries into account, we first prove the following leakage simulation lemma for unbounded leakage.

Lemma 2.1 (Simulation Lemma for Unbounded Advice, Informal). For two distributions X, Y, if $X \to Y$ are indistinguishable against nondeterministic polynomial-time distinguishers, then for any polynomial-length advice Y' of Y, there exists a "simulated advice" X' possibly correlated with X, such that X', Y' are indistinguishable against (deterministic) polynomial-time distinguishers.$\diamondsuit$

We prove this lemma using von Neumann's min-max theorem [Neu28]. In our setting, $X = \mathcal{U}_{\ell(n)}$ is the uniform distribution over strings of the generator output length and, with high probability,

falls outside of $\mathcal{L}$. We take $Y = g(\mathcal{U}_n)$ to be the distribution over images of the super-bit generator, which are YES instances of $\mathcal{L}$. The non-uniform advice is first attached to the random/NO instances X, and the simulation lemma provides simulated advice for the image/YES instances Y.

Once we have the simulated advice X', we can simulate the prover's response for the queries as follows. Since we are considering non-adaptive-query-bounded reductions, we can enumerate the list of CRSes being used and fix them. Then, by the leakage lemma of Gentry-Wichs, we can simulate the short SNARG proofs. This implies a black-box barrier for query-bounded non-uniform reductions in a similar way to Gentry-Wichs.

Nondeterministic Indistinguishability vs Unbounded Leakage Simulation. We indeed prove a stronger lemma than Lemma 2.1, which shows an equivalence between simulating unbounded advice and indistinguishability against nondeterministic distinguishers. This establishes a strong relationship between algorithms with unbounded auxiliary information and nondeterministic computation, which may be of independent interest. We refer to Section 5 for more details of the proof.

Difficulties in Reaching Impossibility for Multi-query Non-uniform Reductions. The reader may wonder whether we can extend the previous impossibility of multi-query reduction for updatable SNARGs into the non-uniform reduction setting. Unfortunately, the NP language we construct in the previous impossibility result cannot be hard for nondeterministic algorithms (in any direction): it is in $\mathsf{NP} \cap \mathsf{coNP}$. In fact, the language we consider is closely related to the notion of OR-compression [BDFH08]: we say an NP language $\mathcal{L}$ supports OR-compression, if there is an efficient algorithm that on input instance $x_1, \ldots, x_k$ outputs a string with length sublinear in k which statistically encodes whether one of $x_1, \ldots, x_k$ is in $\mathcal{L}$. It has been shown that any language supporting OR-compression is in $\mathsf{NP} \cap \mathsf{coNP}$ [FS08; Dru12; Del16].

2.4 One-query Nature of Existing SNARG Reductions

We briefly sketch why the security reductions in many recent SNARG constructions are essentially one-query.

These security reductions have the following pattern: first, switch the distribution $\mathcal{D}_{\mathsf{crs}}$ of CRS to some special distribution $\mathcal{D}'_{\mathsf{crs}}$ while preserving the probability to generate an accepting SNARG proof, following an indistinguishability hybrid argument. More formally speaking, assume there are ℓ hybrids $\mathcal{D}_{\mathsf{crs}} = \mathcal{D}^0_{\mathsf{crs}}, \mathcal{D}^1_{\mathsf{crs}}, \ldots, \mathcal{D}^\ell_{\mathsf{crs}} = \mathcal{D}'_{\mathsf{crs}}$, then

$$\Pr_{\mathcal{D}^i_{\mathsf{crs}}}[\mathcal{P} \text{ wins}] \leq \Pr_{\mathcal{D}^{i+1}_{\mathsf{crs}}}[\mathcal{P} \text{ wins}] + \mathsf{negl}(\lambda), \quad \forall 0 \leq i < \ell.$$

Second, show that under the special CRS distribution, generating an accepting SNARG proof is hard, in the sense that it implies solving a hard search problem.

Suppose a many-query security proof consists of a family of single-query checks $\{\mathsf{Chk}_r\}_{r \in R}$, where each check is either a search check or a one-sided hybrid check, and suppose any successful SNARG adversary makes the sum of the corresponding check advantages noticeable. Then a reduction can sample $r \leftarrow R$, make only the query needed for Chk_r, and inherit the average advantage. For hybrid checks, the one-sided formulation is essential: it ensures that the relevant advantages add rather than cancel.

In the following, we examine several popular SNARG constructions, including hash-and-BARG SNARGs (to name a few, [CJJ22; BBKLP23; JKLV24]) and Sahai-Waters SNARGs [SW21].

2.4.1 Sahai-Waters SNARGs

Here we consider the non-adaptively sound SNARG based on iO from [SW21]. In this construction, the crs contains two obfuscated circuits: a prover circuit and a verifier circuit. The security reduction switches the crs into a punctured crs: for the punctured crs, the adversary must break a OWF in order to succeed. The indistinguishability of standard crs and punctured crs is implied by security of iO. Therefore, the security reduction contains two parts:

- Reduction to OWF. Breaking OWF is a search problem and can be easily batched.
- Reduction to iO indistinguishability. We observe that the security reduction only needs one-side indistinguishability: we only need the adversary's success probability to not drop significantly after switching crs distribution. So this part can also be batched.

2.4.2 Hash-and-BARG SNARGs

We take the hash-and-BARG SNARG for P [CJJ22] as example.

The construction relies on somewhere extractable batch arguments (seBARG), which can be regarded as a falsifiable assumption. seBARG allows the prover to succinctly commit the entire history of a polynomial-time computation (composed of configurations for each step), using a CRS that is generated with respect to a hidden set S of time steps. With an extraction key generated together with the CRS, the configuration for time steps in S can be efficiently extracted from the commitment. The security of seBARG guarantees that:

No-signaling Extraction. For any two sets S_1, S_2 and any efficient prover, the extracted configurations for time steps in $S_1 \cap S_2$ are indistinguishable, for CRS generated w.r.t. S_1 vs. CRS generated w.r.t. S_2.

Extraction Correctness. Whenever the proof is accepted, for any two consecutive time steps in S, the extracted configurations are consistent: the first configuration correctly transitions to the next configuration, and the configuration for the first step is correct. [6]

The security proof of SNARG for P proceeds as follows: let E_i be the event that seBARG proof is accepted, but the extracted configuration for time step i is incorrect. Then, taking $S = \{1, 2\}$, we know $\Pr[E_1]$ is negligible, so by extraction correctness $\Pr[E_2]$ is also negligible. Then we switch to $S = \{2, 3\}$, and $\Pr[E_2]$ should still be negligible by no-signaling extraction. Therefore, we know $\Pr[E_3]$ is also negligible by extraction correctness. After repeating this argument for many times, we conclude that taking $S = \{T - 1, T\}$, $\Pr[E_T]$ is very small, which bounds the success probability of a cheating prover.

Superficially, the query complexity of this security reduction equals to the number of time steps T. However, the entire security reduction only makes non-adaptive queries, and only checks the following two properties for received proofs:

- For each SNARG proof: whether it breaks extraction correctness.
- For each two consecutive proofs for $S = \{i - 1, i\}$ and $S = \{i, i + 1\}$: whether no-signaling extraction holds, or more exactly, whether $\Pr[E_i]$ is noticeably increased.

[6] [CJJ22] additionally uses collision-resistant hash functions to argue this property. We omit that detail here for simplicity.

The natural idea to make this security reduction 1-query is to sample a check. The 1-query reduction samples a random check among the above two types of checks. If the sampled check is of the first type, the reduction algorithm makes that query and perform the check. Otherwise, the reduction algorithm plays the prediction game for whether CRS is generated w.r.t. $S = \{i-1, i\}$ or $S = \{i, i+1\}$: Given a CRS, the reduction algorithm queries this CRS and makes the prediction by checking whether E_i holds.

We note that the trick of "sample a check" in general does not work for indistinguishability-based security, since the prediction advantage in different checks may cancel out after taking average. (For example, in one check the prediction is correct w.p. $3/4$, while in the other check the prediction is correct w.p. $1/4$, they both breaks indistinguishability individually but their average is a trivial $1/2$.). In our case of SNARG security reduction, the second checks are indistinguishability-based. However, we only need indistinguishability *for one direction*, i.e. $\Pr[E_i]$ does not *increase* a lot.

Writing $p_i^+ = \Pr[E_i \mid S = \{i, i+1\}]$ and $p_i^- = \Pr[E_i \mid S = \{i-1, i\}]$, the prediction advantage of the i-th hybrid check is $\frac{1}{2}(p_i^+ - p_i^-)$. Summing these advantages gives a telescoping expression because, except the extraction correctness is broken (which means the adversary breaks the search game), the event E_{i+1} under $S = \{i, i+1\}$ implies E_i under $S = \{i, i+1\}$, as incorrectness of step $i+1$ implies incorrectness of step i. For the endpoints, define $p_T^+ = \Pr[E_T \mid S = \{T-1, T\}]$ and $p_1^- = \Pr[E_1 \mid S = \{1, 2\}]$. With these endpoint conventions, the telescoping gives us

$$\sum_i \frac{p_i^+ - p_i^-}{2} \geq \Pr[E_T | S = \{T-1, T\}] - \Pr[E_1 | S = \{1, 2\}] - \Pr[\text{search game is broken}] - \mathsf{negl}(\lambda).$$

Hence the intermediate terms cancel, leaving only the final extraction event and the probability that a first-type check was violated. Since the proof only needs the one-sided guarantee that p_i^+ is not much larger than p_i^-, negative hybrid advantages are never used to offset positive ones; sampling a random check therefore preserves a noticeable average advantage.

The reduction algorithm of other known hash-and-BARG SNARG constructions (e.g. [BBKLP23; JKLV24]) can also be shown to be inherently 1-query for similar reasons. One particular example is the universal SNARG of [JKLM25] which also relies on witness encryption (or more generally, laconic argument of NP). Since security of witness encryption is non-falsifiable, whether the security reduction of [JKLM25] is 1-query depends on the reduction for the witness encryption. With that in mind, following the above idea, the security reduction of [JKLM25] is a 1-query reduction to security of witness encryption or other falsifiable assumptions.

3 Preliminary

Notations. We denote the set of all positive integers up to n as $[n] = \{1, \ldots, n\}$. Throughout this paper, unless specified, all polynomials we consider are positive polynomials. For any finite set S, $x \leftarrow S$ denotes a uniformly random element x from the set S. Similarly, for any distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes an element x drawn from the distribution $\mathcal{D}$.

For any two strings $x, y \in \{0,1\}^*$ of the same length, we denote their Hamming distance as $\Delta(x, y)$. For any language $\mathcal{L}$ and string x, we write $\Delta(x, \mathcal{L}) = \min_{y \in \mathcal{L} \cap \{0,1\}^{|x|}} \Delta(x, y)$.

For any two distributions $\mathcal{D}_1, \mathcal{D}_2$, we denote their statistical distance as $\Delta(\mathcal{D}_1, \mathcal{D}_2) := \frac{1}{2} \sum_x |\Pr_{\mathcal{D}_1}[x] - \Pr_{\mathcal{D}_2}[x]|$.

Definition 3.1. For any two distributions $\mathcal{D}_1, \mathcal{D}_2$, we say they are (S, ε)-indistinguishable, if for any size-S distinguisher $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\mathcal{D}_1) = 1] - \Pr[\mathcal{A}(\mathcal{D}_2) = 1]| \leq \varepsilon.$$

We say $\mathcal{D}_1, \mathcal{D}_2$ are S-indistinguishable if they are $(S, 1/S)$-indistinguishable. ♢

We say two distributions are (plain) computationally indistinguishable if for every $s(n) = \mathsf{poly}(n)$ there is some $\varepsilon(n) = \mathsf{negl}(n)$ such that the distributions are $(s(n), \varepsilon(n))$-indistinguishable. We say two distributions are sub-exponentially indistinguishable if for some $0 < c < 1$, the distributions are $(2^{n^c}, 2^{-n^c})$-indistinguishable.

3.1 Leakage Simulation Lemma

We use the leakage simulation lemma [JP14; VZ13; Skó16; CCL18].

Theorem 3.2 (Leakage Simulation Lemma, Theorem 2 of [CCL18]). Let X be a distribution over $\{0,1\}^n$, and Z be a distribution over $\{0,1\}^{\ell=\ell(n)}$ jointly distributed with X. Let $\mathcal{F}$ be a collection of deterministic distinguishers $f : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}$. Then for all $\varepsilon > 0$, there exists a (randomized) simulator circuit $h : \{0,1\}^n \to \{0,1\}^\ell$ such that

1. h has complexity $\tilde{O}(2^\ell \varepsilon^{-2})$ relative to $\mathcal{F}$, i.e. h can be computed by an oracle-aided circuit of size $\tilde{O}(2^\ell \varepsilon^{-2})$ with oracle gates that are functions in $\mathcal{F}$.
2. (X, Z) and $(X, h(X))$ are indistinguishable by $\mathcal{F}$, i.e. for every $f \in \mathcal{F}$,

$$\left| \Pr_{(x,z)\leftarrow(X,Z)}[f(x,z) = 1] - \Pr_{x\leftarrow X}[f(x,h(x)) = 1] \right| \leq \varepsilon.$$ ♢

By taking $\mathcal{F}$ as Boolean circuits with size at most s, we obtain the following result:

Corollary 3.3. Let X be distributions over $\{0,1\}^n$, and Z be a distribution over $\{0,1\}^{\ell=\ell(n)}$ jointly distributed with X. For all $s = s(n), \varepsilon > 0$, there exists a (randomized) simulator circuit $h : \{0,1\}^n \to \{0,1\}^\ell$ of size $s' = \tilde{O}(s \cdot 2^\ell \varepsilon^{-2})$ such that (X, Z) and $(X, h(X))$ are (s, ε)-indistinguishable.♢

If we are considering two indistinguishable distributions X, Y instead, then leakage simulation lemma implies that a short leakage for distribution X can be simulated even for distribution Y, by considering hybrids $(X, Z) \approx_c (X, h(X)) \approx_c (Y, h(Y))$.

Corollary 3.4. Let X, Y be two distributions over $\{0,1\}^n$ that are (T, ε')-indistinguishable, and Z be a distribution over $\{0,1\}^{\ell=\ell(n)}$ jointly distributed with X. For all $s = s(n), \varepsilon > 0$, as long as $T \geq 2s'$ where $s' = \tilde{O}(s \cdot 2^\ell \varepsilon^{-2})$, there exists a (randomized) simulator circuit $h : \{0,1\}^n \to \{0,1\}^\ell$ of size s' such that (X, Z) and $(Y, h(Y))$ are $(s, \varepsilon + \varepsilon')$-indistinguishable. ♢

3.2 Incrementally Verifiable Computation (IVC) for NP

Syntax. An incrementally verifiable computation (IVC) scheme for NP consists of the following algorithms:

$\mathcal{G}(1^\lambda, C) \to (\mathsf{crs}_\mathcal{P}, \mathsf{crs}_\mathcal{V})$**.** The poly-time key generation algorithm takes as input the security parameter 1^λ, and a circuit C. It outputs common reference string $\mathsf{crs}_\mathcal{P}, \mathsf{crs}_\mathcal{V}$.

$\mathcal{P}(\mathsf{crs}_{\mathcal{P}}, C, (i, s_i), w_i, \pi_i) \to \pi_{i+1}$**.** The poly-time prover algorithm takes as input the prover common reference string $\mathsf{crs}_{\mathcal{P}}$, the circuit C, iteration counter i, state s_i, non-deterministic advice w_i and a proof π_i for s_i. It outputs a new proof π_{i+1} for $s_{i+1} = C(s_i, w_i)$.

$\mathcal{V}(\mathsf{crs}_{\mathcal{V}}, C, (i, s_i), \pi_i) \to 0/1$. The poly-time verifier algorithm takes as input $\mathsf{crs}_{\mathcal{V}}$, circuit C, iteration counter i, state s_i, and the proof π_i. It outputs a bit.

Definition 3.5. An incrementally verifiable computation scheme is required to satisfy the following properties:

Completeness. For any $\lambda \in \mathbb{N}$, circuit C such that $|C| \leq 2^\lambda$, $t \leq 2^\lambda$ and $(w_0, \ldots, w_{t-1})$, we have

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathsf{crs}_{\mathcal{V}}, C, (t, s_t), \pi_t) = 1 \\ : \\ \end{array} \begin{array}{c} (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}}) \leftarrow \mathcal{G}(1^\lambda, C) \\ s_0 = \emptyset, \pi_0 = \emptyset \\ \forall i \in [t], s_i = C(s_{i-1}, w_{i-1}) \\ \forall i \in [t], \pi_i = \mathcal{P}(\mathsf{crs}_{\mathcal{P}}, C, (i-1, s_{i-1}), w_{i-1}, \pi_{i-1}) \end{array}\right] = 1.$$

Non-adaptive Soundness. For any circuit family $C = \{C_\lambda\}$, $t = t(\lambda)$ and $s_t = \{s_{t,\lambda}\}$ where $|C_\lambda|, t(\lambda), |s_{t,\lambda}|$ are polynomial in λ, and any poly-size adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for any $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathsf{crs}_{\mathcal{V}}, C, (t, s_t), \pi_t) = 1 \\ \wedge \forall (w_0, \ldots, w_{t-1}), s_t \neq C^{(t)}(w_0, \ldots, w_{t-1}) \end{array} : \begin{array}{c} (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}}) \leftarrow \mathcal{G}(1^\lambda, C) \\ \pi_t = \mathcal{A}(\mathsf{crs}_{\mathcal{P}}) \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where $C^{(t)}(w_0, \ldots, w_{t-1})$ is defined inductively as:

- If $t = 0$, then $C^{(0)} = \emptyset$.
- Otherwise, $C^{(t)}(w_0, \ldots, w_{t-1}) = C(C^{(t-1)}(w_0, \ldots, w_{t-2}), w_{t-1})$.

Succinctness. We consider the fully-succinct setting where the size of IVC proof π and $\mathsf{crs} = (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}})$ is $\mathsf{poly}(\lambda)$, independent of the number of iterations as well as instance size. We also consider the sublinear setting where $|\mathsf{crs}|, |\pi| \leq \ell^c$ for some $c \in (0, 1)$ and ℓ is the instance length $|i| + |s_i|$. ♢

3.3 Succinct Non-interactive Arguments (SNARGs)

In this section, we define SNARG for NP relations.

Syntax. A SNARG for an NP relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, with associated language $\mathsf{L}_{\mathcal{R}} = \{x : \exists w \ (x, w) \in \mathcal{R}\}$, consists of the following polynomial time algorithms:

$\mathcal{G}(1^\lambda) \to (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}})$**.** This is a probabilistic algorithm that takes as input a security parameter 1^λ. It outputs a common reference string crs.

$\mathcal{P}(\mathsf{crs}_{\mathcal{P}}, x, w) \to \pi$**.** This is a deterministic algorithm that takes as input a crs, an instance x and corresponding witness w. It outputs a proof π.

$\mathcal{V}(\mathsf{crs}_{\mathcal{V}}, x, \pi) \to \{0,1\}$. This is a deterministic algorithm that takes as input a crs, an instance x and a proof π. It outputs a bit (1 to accept, 0 to reject).

In this paper, we consider SNARGs with succinct proof and crs.

Definition 3.6 (SNARG). A SNARG for an NP relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, with associated language $\mathsf{L}_{\mathcal{R}}$, is required to satisfy the following properties:

Completeness. For any $\lambda, n, m \in \mathbb{N}$ such that $n, m \leq 2^\lambda$ and any $x \in \{0,1\}^n, w \in \{0,1\}^m$ such that $(x, w) \in \mathcal{R}$ we have that

$$\Pr\left[\mathcal{V}(\mathsf{crs}_{\mathcal{V}}, x, \pi) = 1 \ : \ \begin{array}{c} (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}}) \leftarrow \mathcal{G}(1^\lambda) \\ \pi = \mathcal{P}(\mathsf{crs}_{\mathcal{P}}, x, w) \end{array} \right] = 1.$$

Succinctness. The size of SNARG proof π and $\mathsf{crs} = (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}})$ is $\mathsf{poly}(\lambda)$.

Non-adaptive Soundness. For any poly-size adversary $\mathcal{A}$, any polynomial $n = n(\lambda), m = m(\lambda)$ and sequence of $\left\{x = x(\lambda) \in \{0,1\}^{n(\lambda)} \backslash \mathsf{L}_{\mathcal{R}}\right\}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathcal{V}(\mathsf{crs}_{\mathcal{V}}, x, \pi) = 1 \ : \ \begin{array}{c} (\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}}) \leftarrow \mathcal{G}(1^\lambda) \\ \pi \leftarrow \mathcal{A}(\mathsf{crs}_{\mathcal{P}}) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

♢

3.4 Falsifiable Cryptographic Assumptions

Our notion of falsifiable cryptographic assumptions follows from [GW11].

Definition 3.7. A falsifiable cryptographic assumption consists of an efficient interactive challenger $\mathcal{C}$ and a constant $c \in [0,1)$. On security parameter λ, the challenger $\mathcal{C}(1^\lambda)$ interacts with a machine $\mathcal{A}(1^\lambda)$ and may output a special symbol win. If this occurs, we say that $\mathcal{A}(1^\lambda)$ wins $\mathcal{C}(1^\lambda)$. The assumption associated with the tuple $(\mathcal{C}, c)$ states that for any efficient $\mathcal{A}$, we have $\Pr[\mathcal{A}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \leq c + \mathsf{negl}(\lambda)$, where the probability is over the random coins of $\mathcal{C}$ and $\mathcal{A}$.

For any constant $\delta > 0$, the δ-exponential version of the assumption associated with $(\mathcal{C}, c)$ states that for every $\mathcal{A}$ of size $2^{O(\lambda^\delta)}$ we have $\Pr[\mathcal{A}(1^\lambda) \text{ wins } \mathcal{C}(1^\lambda)] \leq c + 2^{-\Omega(\lambda^\delta)}$.

We say that a (possibly inefficient) machine $\mathcal{A}$ breaks such assumption if its probability of winning exceeds that of the assumption. ♢

3.5 Black-Box Reductions

We only discuss black-box reductions showing the non-adaptive soundness of some IVC (and more generally, SNARG) system based on some falsifiable assumption $(\mathcal{C}, c)$. To unify the terminology, we use "statement" to represent the what a prover is proving: for IVC the statement is a triple (C, t, s_t), while for SNARG the statement is an instance x.

Definition 3.8 (Black-box reduction, uniform version). A *uniform* black-box reduction showing the non-adaptive soundness of a SNARG (or IVC) scheme based on a falsifiable assumption $(\mathcal{C}, c)$ is an efficient oracle-access machine $\mathcal{R}^{(\cdot)}(\cdot)$ such that, for every (possibly inefficient) adversary $\tilde{\mathcal{P}}$ and family of statements ($\{x_\lambda\}$ for SNARG, or additionally $x_\lambda = (C_\lambda, t_\lambda, s_{t,\lambda})$ for IVC) that breaks the SNARG (or IVC) non-adaptive soundness, the machine $\mathcal{R}^{\tilde{\mathcal{P}}}(1^\lambda, x_\lambda)$ breaks the assumption. ♢

For our IVC lower bound, we take one additional assumption for the black-box reduction: the winning probability of the machine $\mathcal{R}^{\tilde{\mathcal{P}}}(1^\lambda, x_\lambda)$ should only depends on $\lambda, |x_\lambda|$ and the success probabilty of $\tilde{\mathcal{P}}$, but is independent of $\tilde{\mathcal{P}}$ otherwise. We say such reductions have implementation-oblivious winning probability. Concretely, this captures reductions whose analysis treats the prover only through its externally visible success probability.

Definition 3.9. We say a black-box reduction $\mathcal{R}^{(\cdot)}(\cdot)$ has implementation-oblivious winning probability, if the winning probability of $\mathcal{R}^{\tilde{\mathcal{P}}}(1^\lambda, x_\lambda)$ only depends on $\lambda, |x_\lambda|$ and the success probability of $\tilde{\mathcal{P}}$, but is independent of $\tilde{\mathcal{P}}$ otherwise. ♢

In a recent paper Jin, Kalai, Lombardi, and Vaikuntanathan [JKLV24] gives a construction of non-adaptive SNARG whose reduction algorithm takes the instance x together with an advice z_x that cannot be efficiently computed from x (in their case, a propositional proof for x not being in the language). To take their reduction into account we introduce a more general notion of non-uniform black-box reduction.

Definition 3.10 (Black-box reduction, non-uniform version). A *non-uniform* black-box reduction showing the non-adaptive soundness of a SNARG (or IVC) scheme based on a falsifiable assumption $(\mathcal{C}, c)$ is an efficient oracle-access machine $\mathcal{R}^{(\cdot)}(\cdot)$ and a computationally unbounded instance-to-advice mapping F with advice length bounded by a fixed polynomial in λ and $|x_\lambda|$ such that, for every (possibly inefficient) adversary $\tilde{\mathcal{P}}$ and family of statements ($\{x_\lambda\}$ for SNARG, or additionally $x_\lambda = (C_\lambda, t_\lambda, s_{t,\lambda})$ for IVC) that breaks the SNARG (or IVC) non-adaptive soundness, if $z_\lambda \leftarrow F(x_\lambda)$, then the machine $\mathcal{R}^{\tilde{\mathcal{P}}}(1^\lambda, x_\lambda, z_\lambda)$ breaks the assumption, where the probability of success is over the draw of z_λ and the usual experiment randomness. ♢

4 Black-box Reduction Lower-bound for IVC

In this section, we prove a lower bound for black-box reductions for IVC. We first abstract the primitive we use to build the OR-homomorphic structure in the language in Section 4.1, state our impossibility result in Section 4.2, and prove the result in Section 4.3.

4.1 OR-Homomorphic Encoding

We introduce OR-homomorphic encoding to abstract our usage of FHE to build the language, as mentioned in Section 2. For this primitive, with a public key, we can generate encoding of 0's and 1's. Additionally, generating ct as an encoding of 1 also gives a signature that certifies ct encodes 1. This encoding supports OR-homomorphic operation.

Syntax. An OR-homomorphic encoding consists of the following algorithms:

$\mathcal{G}(1^\lambda) \to \mathsf{pk}$**.** This poly-time key generation algorithm take security parameter 1^λ as input, and outputs a public key pk. We write length of pk as ℓ_{pk}.

$\mathsf{Enc}(\mathsf{pk}, b) \to \mathsf{ct}[, \sigma]$**.** This algorithm takes as input the public key pk and a bit $b \in \{0,1\}$, and outputs a ciphertext $\mathsf{ct} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ encoding b, and a signature $\sigma \in \{0,1\}^{\mathsf{poly}(\lambda)}$ certifying that ct encodes 1, in the case $b = 1$. We write ℓ_{ct} as the length of ct.

$\mathsf{Verify}(\mathsf{pk}, \mathsf{ct}, \sigma) \to 0/1$. This poly-time verification algorithm takes as input pk, a ciphertext ct and a signature σ. It outputs a bit indicating whether ct is certified as an encoding of 1.

$\mathsf{HomOR}(\mathsf{pk}, \mathsf{ct}_a, \mathsf{ct}_b) \to \mathsf{ct}_c$**.** This poly-time algorithm takes as input pk and two ciphertexts $\mathsf{ct}_a, \mathsf{ct}_b$. It computes OR homomorphically and outputs a ciphertext ct_c.

Moreover, we require every ciphertext to be uniquely associated with one bit $0/1$. We introduce an *inefficient* decoding algorithm as the notation for the underlying bit of a ciphertext.

$\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}) \to b$**.** This *inefficient* algorithm takes as input the public key pk and a ciphertext ct. It outputs a bit $b \in \{0, 1\}$ that is encoded by ct.

Definition 4.1. An OR-homomorphic encoding scheme is required to satisfy the following properties:

Compactness. There exists a polynomial $p(\lambda)$ such that the size of any ciphertext is bounded by $p(\lambda)$, even if it is generated from homomorphic operations.

Encoding Correctness. For any $\lambda \in \mathbb{N}$, $b \in \{0, 1\}$,

$$\Pr\left[\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}) = b \ : \ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, b) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

OR-Homomorphic Correctness. For any $\lambda \in \mathbb{N}$ and $\mathsf{pk} \leftarrow \mathcal{G}(1^\lambda)$, for any two ciphertexts $\mathsf{ct}_a, \mathsf{ct}_b$ where $\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_a) = a, \mathsf{Dec}(\mathsf{pk}, \mathsf{ct}_b) = b$, we have

$$\mathsf{Dec}(\mathsf{pk}, \mathsf{HomOR}(\mathsf{pk}, \mathsf{ct}_a, \mathsf{ct}_b)) = a \vee b.$$

Similar to leveled FHE, we only require OR-homomorphism to hold for bounded-depth operation from freshly-generated encodings. In this work, we consider depth $O(\log \lambda)$.

Signature Correctness. For any $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathsf{Verify}(\mathsf{pk}, \mathsf{ct}, \sigma) = 1 \ : \ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^\lambda) \\ (\mathsf{ct}, \sigma) \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \end{array} \right] = 1.$$

Statistical Signature Security. For any $\lambda \in \mathbb{N}$ and $\mathsf{pk} \leftarrow \mathcal{G}(1^\lambda)$, for any ct where $\mathsf{Dec}(\mathsf{pk}, \mathsf{ct}) = 0$, there is no σ such that $\mathsf{Verify}(\mathsf{pk}, \mathsf{ct}, \sigma) = 1$.

S**-Security.** The following two distributions are $(S, 1/S)$-indistinguishable:

$$\left\{ (\mathsf{pk}, \mathsf{ct}_0) \middle| \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^\lambda) \\ \mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, 0) \end{array} \right\} \approx_c \left\{ (\mathsf{pk}, \mathsf{ct}_1) \middle| \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^\lambda) \\ \mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \end{array} \right\}.$$

In later applications, it is more convenient to describe S as a function of the IVC instance size, i.e. $\ell = \ell_{\mathsf{pk}} + \ell_{\mathsf{ct}} + O(\log \lambda)$. $\diamondsuit$

It is not hard to see that FHE directly implies OR-homomorphic encoding with corresponding security. Assuming FHE with 2^{ℓ^c} security (based on sub-exponential security of LWE), the corresponding OR-homomorphic encoding also has 2^{ℓ^c}-security.

Lemma 4.2. Assuming sub-exponential hardness of LWE, for some $0 < c < 1$ there is an OR-homomorphic encoding scheme with 2^{ℓ^c}-security. ♢

Remark 2. We state the primitive we use as OR-homomorphic encoding instead of FHE, to propose the following question: it is possible to define a version of OR-homomorphic encoding (or another useful primitive) that is plausibly secure against non-deterministic computation? If this is possible, then the barrier for updatable SNARG may be generalized for non-uniform reductions.

Unfortunately, the current OR-homomorphic encoding is impossible to be secure against non-deterministic algorithms, by previous study on OR-compression [FS08; Dru12; Del16]. See the technical overview section for more discussion. ♢

4.2 A Simulatable Adversary

To reach the impossibility result, we construct an inefficient but simulatable adversary in a similar flavor to [GW11]: the adversary 1) breaks the IVC, and 2) can be simulated by an efficient algorithm, such that all computationally bounded algorithms, including the reduction algorithm[7], cannot distinguish.

Parameters In Theorem Statement. We define the parameter notations in our results.

- λ: the security parameter of the IVC scheme as well as the underlying falsifiable assumptions.
- λ': the security parameter of the OR-homomorphic encoding. Looking ahead, λ' is polynomial in λ.
- ℓ: the size of the IVC statement (i, s_i). Looking ahead, $\ell = \ell_{\mathsf{pk}}(\lambda') + \ell_{\mathsf{ct}}(\lambda') + O(\log \lambda)$, since the iteration will only have $\mathsf{poly}(\lambda)$ steps, and s_i will be composed of a public key and a ciphertext of the OR-homomorphic encoding scheme as well as an integer less than $\mathsf{poly}(\lambda)$. While the full IVC statement also contains a description of the iteration circuit C, in this work $|C|$ will always be a fixed polynomial in ℓ. Therefore we use ℓ to represent IVC statement size for simplicity.
- L: an upper bound on the size of crs and π of the IVC scheme. L is a function of security parameter λ and statement size ℓ.
- T: the running time of the IVC security reduction algorithm. Same as L, T is a function of λ and ℓ.

Theorem 4.3. Assume the existence of OR-homomorphic encoding (Definition 4.1) with $2^{\ell^c}-$security for some $c \in (0, 1)$. Let Π be any IVC scheme with $|\mathsf{crs}| + |\pi| = O(\ell^{c'})$ $(c' < c)$. Then, there exists a polynomial setting for $\lambda' = \mathsf{poly}(\lambda)$ (λ' is the security parameter of OR-homomorhpic encoding) such that for every polynomial distinguisher time bound $T = T(\lambda, \ell)$[8] and every polynomial $p = p(\lambda)$, there exist an unbounded adversary $\tilde{\mathcal{P}}$ and an efficient simulator $\mathcal{S}$, both depending on T and p, such that the following two properties hold.

[7] To be rigorous, we should also include the falsifiable assumption security game.

[8] Note that our result naturally extends to quasi-polynomial distinguishers, and therefore quasi-poly-time reduction algorithms.

- $\tilde{\mathcal{P}}$ is a computationally unbounded Π-adversary. On input 1^λ it outputs (C, t, s_t) and a (computationally unbounded) machine $\tilde{\mathcal{P}}'$, such that C, t, s_t and $\tilde{\mathcal{P}}'$ breaks the non-adaptive soundness of Π with probability $1/2$.
- $\tilde{\mathcal{P}}$ is poly-time simulatable against uniform reduction algorithms running in time at most T. That is, for every oracle-aided distinguisher $\mathcal{D}^{(\cdot)}(\cdot)$ running in time at most T, the simulator $\mathcal{S}$ on input 1^λ outputs (C, t, s_t) and an *efficient* machine $\mathcal{S}'$, such that

$$\left| \Pr_{C,t,s_t,\tilde{\mathcal{P}}' \leftarrow \tilde{\mathcal{P}}(1^\lambda)} [\mathcal{D}^{\tilde{\mathcal{P}}'}(1^\lambda, C, t, s_t) = 1] - \Pr_{C,t,s_t,\mathcal{S}' \leftarrow \mathcal{S}(1^\lambda)} [\mathcal{D}^{\mathcal{S}'}(1^\lambda, C, t, s_t) = 1] \right| \leq \frac{1}{p(\lambda)}. \qquad \diamondsuit$$

Take distinguisher D as the black-box reduction algorithm composed with the falsifiable assumption security game, i.e. D simulates the falsifiable game with the reduction algorithm as the adversary. By Theorem 4.3, efficient uniform black-box reduction is impossible for IVC.

Theorem 4.4. Assume the existence of OR-homomorphic encoding (Definition 4.1) with 2^{ℓ^c}−security for some $c \in (0, 1)$. Then any IVC scheme with $|\mathsf{crs}| + |\pi| = O(\ell^{c'})$ $(c' < c)$ does not have a polynomial time uniform black-box security reduction from falsifiable assumptions with implementation-oblivious winning probability (unless the assumptions are false). $\diamondsuit$

Proof. Fix statement size ℓ by taking λ' as specified by Theorem 4.3. By the definition of implementation-oblivious winning probability, there exists a polynomial $q = q(\lambda)$ such that for any $\tilde{\mathcal{P}}$ that breaks IVC non-adaptive soundness for a statement of size n with probability $1/2$, $\mathcal{R}^{\tilde{\mathcal{P}}}(1^\lambda, x_\lambda)$ breaks the assumption with winning advantage $1/q(\lambda)$.

Take $p(\lambda) = 2q(\lambda)$ in Theorem 4.3, and T as the combined time complexity of $\mathcal{R}$ and the falsifiable assumption security game, on security parameter λ and statement size ℓ. With $\tilde{\mathcal{P}}$ and $\mathcal{S}$ as specified in Theorem 4.3:

- $\mathcal{R}$ supplied with $\tilde{\mathcal{P}}$ breaks the assumption with winning advantage $1/q$.
- By indistinguishability of $\tilde{\mathcal{P}}$ and $\mathcal{S}$, $\mathcal{R}$ supplied with $\mathcal{S}$ also breaks the assumption with winning advantage $1/2q$. $\square$

Basing OR-homomorphic encoding on sub-exponential LWE gives the following theorem.

Theorem 4.5. Assume LWE with sub-exponential security. Then for some $c > 0$, any IVC for NP scheme with $|\mathsf{crs}| + |\pi| = O(\ell^{c'})$ $(c' < c)$ has no polynomial-time uniform black-box security reduction from falsifiable assumptions whose winning probability depends on the adversary only through its success probability and statement size (unless the assumptions are false). $\diamondsuit$

4.3 Proof of Theorem 4.3

Construction of iteration function. We first construct the iteration circuit C.

Consider a Merkle tree where each node is associated with a ciphertext from the OR-homomorphic encoding. We let each non-leaf node in the Merkle tree implement the OR-homomorphism: its ciphertext is computed by homomorphically evaluating the OR of two ciphertexts of its children. Therefore, the root node has a ciphertext that encodes the OR of all bits encoded by the leaves.

The iteration function imitates the leaf-to-root computation in the Merkle tree, akin to verifying a local opening to a leaf node in a bottom-to-top approach. In the first step, it reads the value of

the leaf node (from nondeterminism). Then, in every subsequent step, it computes the father of the current node by reading the sibling node from nondeterminism (as well as the order of two siblings). The iteration ends by computing the root. We further require that the leaf node provided in the first step must encode 1, by additionally reading a signature certifying the leaf encodes 1 from nondeterminism. By correctness of OR-homomorphism, the root calculated in the last step must encode 1 if the whole computation is done honestly. Conversely, a root that encodes 0 is guaranteed to be a false statement.

We give the formal description below. The construction uses a OR-homomorphic encoding scheme as an ingredient, and is parameterized by the number of leaves $m = \mathsf{poly}(\lambda)$, and the security parameter λ' of the OR-homomorphic encoding scheme. In the presentation below, the iteration function implicitly keeps an iteration counter in the state.

- In the first iteration, the witness w is parsed as $(\mathsf{pk}, i, \mathsf{ct}, \sigma)$, where pk is a public key, i is the index of the leaf node, ct is a ciphertext and σ is the corresponding signature. If $\mathsf{Verify}(\mathsf{pk}, \mathsf{ct}, \sigma) = 0$, then the iteration aborts by entering some "fail state"; otherwise it outputs $(\mathsf{pk}, i, \mathsf{ct})$.
- In the following $\log m$ iterations, the witness w is parsed as $\mathsf{ct}_{\mathsf{sibling}}$ and the current state is parsed as $(\mathsf{pk}, i, \mathsf{ct})$. The iteration function computes $\mathsf{ct}_{\mathsf{father}}$ as $\mathsf{HomOR}(\mathsf{ct}, \mathsf{ct}_{\mathsf{sibling}})$ or $\mathsf{HomOR}(\mathsf{ct}_{\mathsf{sibling}}, \mathsf{ct})$ depending the order of $\mathsf{ct}, \mathsf{ct}_{\mathsf{sibling}}$ (which is decided by leaf index i and the current iteration counter). The new state is $(\mathsf{pk}, i, \mathsf{ct}_{\mathsf{father}})$.
- Specially, in the last step the iteration function deletes the leaf index i from which the computation begins. So the final state has the form $(\mathsf{pk}, \mathsf{ct}_{\mathsf{root}})$.

By this construction, the size of IVC statement is bounded by $\ell_{\mathsf{pk}} + \ell_{\mathsf{ct}} + O(\log \lambda)$.

Parameter settings. We recall or define the following parameters:

- We use $L = L(\lambda, \ell) = \mathsf{poly}(\lambda) \cdot O(\ell^{c'})$ to bound the $|\mathsf{crs}|, |\pi|$ in the IVC protocol Π.
- We take security parameter λ' of the OR-homomorphic encoding large enough (but still polynomial in λ) such that the OR-homomorphic encoding scheme is $2^{\omega(L)}$-secure. This is possible since the OR-homomorphic encoding has 2^{ℓ^c}-security.
- Recall polynomial $T = T(\lambda, \ell)$ bounds the query complexity of the distinguisher $\mathcal{D}$.
- We take the number of leaves $m = \mathsf{poly}(\lambda)$ large enough that $2T^2/m \leq 1/(2p(\lambda))$. We assume m is a power of 2 for simplicity, and $m2^{-L} \leq 1/(2p(\lambda))$.
- $T^*(\lambda) = 2^{O(L)}$ is a time parameter that would be specified later.

Construction of the adversary and simulator. We show the construction of adversary $\tilde{\mathcal{P}}$ and simulator $\mathcal{S}$. For ease of exposition, we first show the construction of $\mathcal{S}$, then incrementally modify it to $\tilde{\mathcal{P}}$ through a series of hybrids.

In the construction below, we consider $\tilde{\mathcal{P}}', \mathcal{S}'$ as randomized machines. It will be obvious by our construction that one can always convert them into deterministic machines by embedding in a PRF key to generate the randomness (since without loss of generality one can assume the distinguisher makes every oracle query at most once).[9]

[9] We note that the PRF needs sub-exponential security.

Algorithm of Simulator $\mathcal{S}$

1. Sample $\mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'})$ and $(\mathsf{ct}_i, \sigma_i) \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)$ for $1 \le i \le m$.
2. Compute rt as the Merkle tree root from leaves $\{\mathsf{ct}_i\}_{i\in[m]}$, by applying $\mathsf{HomOR}(\mathsf{pk}, \cdot, \cdot)$ at every non-leaf node.
3. Output C described above, $t = 1 + \log m$, $s_t = (\mathsf{pk}, \mathsf{rt})$, and the efficient $\mathcal{S}'$ with $(\mathsf{pk}, \{\mathsf{ct}_i, \sigma_i\}_{i\in[m]})$ hard-coded in, that performs the following computation on input $\mathsf{crs}_{\mathcal{P}}$:
   (a) Sample $i \leftarrow [m]$. Compute $\pi_i^{\mathsf{crs}_{\mathcal{P}}} = \Pi.\mathcal{P}(\mathsf{crs}_{\mathcal{P}}, C, (0, \emptyset), (\mathsf{pk}, i, \mathsf{ct}_i, \sigma_i), \emptyset)$.
   (b) Start with $\pi_1 = \pi_i^{\mathsf{crs}_{\mathcal{P}}}$, compute the final proof π_t honestly through the remaining $\log m$ ancestor computations using the siblings of ancestors of leaf i as the witnesses.
   (c) Output π_t.

Figure 1: Algorithm of the efficient simulator $\mathcal{S}$.

For $\tilde{\mathcal{P}}$, all leaves of the Merkle tree ct_i is sampled from $\mathsf{Enc}(\mathsf{pk}, 0)$ instead of $\mathsf{Enc}(\mathsf{pk}, 1)$. This guarantee that the Merkle tree root encodes 0, and the statement $\tilde{\mathcal{P}}$ is proving is false. The problem is $\tilde{\mathcal{P}}$ needs to generate *accepting* proofs.

For the simulator, every proof is generated by taking some $\pi_i^{\mathsf{crs}_{\mathcal{P}}}$ stating the leaf i encodes 1, then honestly upgrading it into a proof for the IVC statement using other leaves of the Merkle tree. $\tilde{\mathcal{P}}'$ generates proofs by the same steps, except it starts with a "fake" proof $\tilde{\pi}_i^{\mathsf{crs}_{\mathcal{P}}}$, as the leaf node i actually encodes 0. Such "fake" proofs are generated by the leakage simulation lemma (see Corollary 3.3).

In the following, we introduce a series of algorithms that gradually changes from $\mathcal{S}$ to $\tilde{\mathcal{P}}$ by sampling a prefix of leaves from $\mathsf{Enc}(\mathsf{pk}, 0)$ and the rest from $\mathsf{Enc}(\mathsf{pk}, 1)$. Also, for simplicity we denote sampling $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)$ and computing $\pi_i^{\mathsf{crs}_{\mathcal{P}}}$ honestly as $(\mathsf{ct}_i, \pi_i^{\mathsf{crs}_{\mathcal{P}}}) \leftarrow \mathsf{HonestStep1}(\mathsf{pk}, i, \mathsf{crs}_{\mathcal{P}})$.

The detailed description of the algorithms of $\mathcal{H}_j$ for $0 \le j \le m$ is shown in Figure 2. We note that $\mathcal{H}_0$ is equivalent to $\mathcal{S}$, and we take $\tilde{\mathcal{P}}$ as $\mathcal{H}_m$. Clarification for functions used in Figure 2: set $\tilde{\mathcal{S}}_0(\mathsf{pk})$ to output the empty list. For $i \ge 1$, the sampler $\tilde{\mathcal{S}}_i(\mathsf{pk})$ outputs

$$\left\{\tilde{\mathsf{ct}}_k, \{\tilde{\pi}_k^a\}_{a\in\{0,1\}^L}\right\}_{k\le i},$$

where each $\tilde{\mathsf{ct}}_k$ is sampled as an encoding of 0, and each $\tilde{\pi}_k^a$ is a simulated first-step proof for CRS string a. The circuit $\mathcal{F}_i^a$ takes as input pk, the previously sampled prefix $\left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^{a'}\right\}_{k<i, a'\in\{0,1\}^L}$, and the new ciphertext $\tilde{\mathsf{ct}}_i$, and outputs the simulated proof $\tilde{\pi}_i^a$.

Analysis. To prove theorem 4.3, we first show the validity of our construction of $\mathcal{H}_j$ i.e. Lemma 4.6, and that the distinguisher $\mathcal{D}$ cannot distinguish any two consecutive $\mathcal{H}_i, \mathcal{H}_{i+1}$ (see Lemma 4.7).

Recall that key generation for the OR-homomorphic encoding uses security parameter λ', while the resulting IVC statement length satisfies $\ell = \ell_{\mathsf{pk}}(\lambda') + \ell_{\mathsf{ct}}(\lambda') + O(\log \lambda)$.

Lemma 4.6. For every $i \in [m]$, the following two properties holds:

Algorithm of machine $\mathcal{H}_j$ for $0 \le j \le m$

1. Sample $\mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'})$. From $i = 1$ to j, we sample $\tilde{\mathsf{ct}}_i$ from $\mathsf{Enc}(\mathsf{pk}, 0)$ together with their first-step proofs $\tilde{\pi}_i^{\mathsf{crs}_\mathcal{P}}$. The sampling process is defined inductively from $i = 1$ to j. For every i, assume the sampling procedure for $\left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^c\right\}_{k \le i, c \in \{0,1\}^L}$ is $\tilde{\mathcal{S}}_i(\mathsf{pk})$, we sample $\tilde{\mathsf{ct}}_i$ and $\{\tilde{\pi}_i^c\}_{c \in \{0,1\}^L}$ by the following way:

   (a) Sample $\tilde{\mathsf{ct}}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 0)$.

   (b) For every possible $\mathsf{crs}_\mathcal{P} \in \{0,1\}^L$, we sample $\tilde{\pi}_i^{\mathsf{crs}_\mathcal{P}}$ from a $2^{O(L)}$-size circuit $\mathcal{F}_i^{\mathsf{crs}_\mathcal{P}}$ which takes input $\left(\mathsf{pk}, \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a \in \{0,1\}^L}, \tilde{\mathsf{ct}}_i\right)$, such that

$$\left\{\begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a \in \{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ (\mathsf{ct}_i, \pi_i^{\mathsf{crs}_\mathcal{P}}) \leftarrow \mathsf{HonestStep1}(\mathsf{pk}, i, \mathsf{crs}_\mathcal{P}) \end{array}\right\}, \left\{\begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a \in \{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ \tilde{\mathsf{ct}}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 0), \tilde{\pi}_i^{\mathsf{crs}_\mathcal{P}} \leftarrow \mathcal{F}_i^{\mathsf{crs}_\mathcal{P}} \end{array}\right\}$$

   are $(T^*, 2^{-2L})$-indistinguishable. Recall $T^* = 2^{O(L)}$ is a fixed time parameter that would be specified later.

   To show the existence of such $\mathcal{F}_i^{\mathsf{crs}_\mathcal{P}}$, we will apply Corollary 3.4 by considering distinguishers of size T^*, leakage length L and advantage $\varepsilon = 2^{-2L-1}$. Note that to apply Corollary 3.4, we need to show that the two distributions ignoring $\pi_i^{\mathsf{crs}_\mathcal{P}}$ (or $\tilde{\pi}_i^{\mathsf{crs}_\mathcal{P}}$) are indistinguishable (see Lemma 4.6).

   For $j < i \le m$, sample $(\mathsf{ct}_i, \sigma_i) \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)$.

2. Compute rt as the Merkle tree root from leaves $\left\{\tilde{\mathsf{ct}}_i\right\}_{i \le j}, \{\mathsf{ct}_i\}_{i > j}$ by applying $\mathsf{HomOR}(\mathsf{pk}, \cdot, \cdot)$ at every non-leaf node.

3. Output C described above, $t = 1 + \log m$, $s_t = (\mathsf{pk}, \mathsf{rt})$, and the $\mathcal{H}'_j$ hard-coded with

$$\left(\mathsf{pk}, \left\{\tilde{\mathsf{ct}}_i, \tilde{\pi}_i^a\right\}_{1 \le i \le j, a \in \{0,1\}^L}, \{\mathsf{ct}_i, \sigma_i\}_{j+1 \le i \le m}\right),$$

   that performs the following computation on input $\mathsf{crs}_\mathcal{P}$:

   (a) Sample $i \leftarrow [m]$. If $i \le j$, then take out $\pi_1 = \tilde{\pi}_i^{\mathsf{crs}_\mathcal{P}}$; otherwise, honestly compute $\pi_1 = \pi_i^{\mathsf{crs}_\mathcal{P}} = \Pi.\mathcal{P}(\mathsf{crs}_\mathcal{P}, C, (0, \emptyset), (\mathsf{pk}, i, \mathsf{ct}_i, \sigma_i), \emptyset)$.

   (b) Start with π_1, compute the final proof π_t honestly through the remaining $\log m$ ancestor computations using the siblings of ancestors of leaf i as the witnesses.

   (c) Output π_t.

Figure 2: Algorithm of hybrid machines $\mathcal{H}_j$ for $0 \le j \le m$. $\mathcal{H}_0$ is equivalent to $\mathcal{S}$, and $\tilde{\mathcal{P}}$ is defined as $\mathcal{H}_m$.

- For every valid c, the running time of $\mathcal{F}_i^c$ defined in the description of $\mathcal{H}_j$ $(j \ge i)$ is $2^{O(L)}$ that only depends on T^* and L.

- The two distributions below are $(2^{\omega(L)}, 2^{-\omega(L)})$-indistinguishable.

$$\left\{ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a\in\{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ \mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \end{array} \right\} \approx \left\{ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a\in\{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ \tilde{\mathsf{ct}}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 0) \end{array} \right\}.$$

♢

Proof. We prove the lemma by induction. For $i = 1$, the two distributions in the second property are $2^{\omega(L)}$-indistinguishable by the setting of parameters. Therefore, by Corollary 3.4, the running time of $\mathcal{F}_i^c$ is $2^{O(L)}$, which only depends on T^* and L.

Now assume that the properties hold for $i' = 1, \ldots, i-1$. To prove the two properties for i, we first examine the second property. The two distributions can be regarded as obtained by computing $\tilde{\mathcal{S}}_{i-1}(\mathsf{pk})$ from $\{\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0/1)\}$. We observe that computing $\tilde{\mathcal{S}}_{i-1}(\mathsf{pk})$ is relatively fast: By the efficiency of $\mathcal{F}_{i'}^c$ for $i' < i$, the running time of $\tilde{\mathcal{S}}_{i-1}(\mathsf{pk})$ is $i \cdot (2^L \cdot 2^{O(L)} + \mathsf{poly}(\lambda)) = 2^{O(L)}$. Therefore, from $2^{\omega(L)}$-security we know the two distributions are still $2^{\omega(L)}$-indistinguishable. Given the second property, the first property (i.e. the running time of $\mathcal{F}_i^c$) follows from leakage simulation i.e. Corollary 3.4. □

Now we set the exact value of T^*. We take it as the maximum running time over $0 \le i \le m$ of simulating the whole execution of the distinguisher $\mathcal{D}$, after $\mathsf{pk}, \left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k<i, a\in\{0,1\}^L}$ are generated. In particular, this includes sampling $\{\mathsf{ct}_k\}_{k>i}$, computing Merkle tree root from $\{\mathsf{ct}_k\}$, running $\mathcal{D}$ and answer all crs queries of $\mathcal{D}$ using preprocessed first-step proofs, with total running time $2^{O(L)}$.

With all hybrid algorithms well-defined, we prove that for every $1 \le i \le m$, time-bounded distinguisher $\mathcal{D}$ cannot tell $\mathcal{H}_{i-1}$ from $\mathcal{H}_i$. By hybrid argument, this implies that it cannot distinguish $\mathcal{H}_0$ and $\mathcal{H}_m$, i.e. $\mathcal{S}$ and $\tilde{\mathcal{P}}$.

Lemma 4.7. For any distinguisher $\mathcal{D}^{(\cdot)}(\cdot)$ that runs in time $T = T(n)$, and any $1 \le i \le m$,

$$\left| \Pr_{C,t,s_t,\mathcal{H}'_{i-1}\leftarrow\mathcal{H}_{i-1}(1^\lambda)}[\mathcal{D}^{\mathcal{H}'_{i-1}}(1^\lambda, C, t, s_t) = 1] - \Pr_{C,t,s_t,\mathcal{H}'_i\leftarrow\mathcal{H}_i(1^\lambda)}[\mathcal{D}^{\mathcal{H}'_i}(1^\lambda, C, t, s_t) = 1] \right| \le \frac{2T^2}{m^2} + 2^{-L}.$$

♢

Proof. By our construction of $\mathcal{H}_i$, the full execution of running $\mathcal{H}_i$ and then $\mathcal{D}^{\mathcal{H}'_i}(1^\lambda, C, t, s_t)$ can be expressed by the following steps:

- Generate $\mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'})$, $\left\{\tilde{\mathsf{ct}}_k, \tilde{\pi}_k^a\right\}_{k\le i, a\in\{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_i(\mathsf{pk})$, and $\{\mathsf{ct}_k \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)\}_{k>i}$.
- Simulate the execution of $\mathcal{D}$. Whenever $\mathcal{D}^{\mathcal{H}'_i}(1^\lambda, C, t, s_t)$ issues a query to its oracle with input $\mathsf{crs}_{\mathcal{P}}$, an random leaf $k \in [m]$ is allocated to this query. The query answer is efficiently computed from $\pi_k^{\mathsf{crs}_{\mathcal{P}}}$ (as an honest proof) or $\tilde{\pi}_k^{\mathsf{crs}_{\mathcal{P}}}$ (as a "fake" proof), and all leaves.

Our first observations is, with high probability, at most one query would be allocated with the i-th leaf.

Claim 4.8. For any $0 \le j \le m$ and $i \in [m]$,

$$\Pr_{C,t,s_t\mathcal{H}'_j\leftarrow\mathcal{H}_j(1^\lambda)}[\mathcal{D}^{\mathcal{H}'_j}(1^\lambda, C, t, s_t) \text{ queries at leaf } i \text{ at least twice}] \le \frac{T^2}{m^2}.$$

♢

Proof. During the execution only T queries are made, while each query has probability $\frac{1}{m}$ to be allocated with leaf i. By union bound, for at least two queries to be allocated at leaf i, the probability is at most $\frac{T^2}{m^2}$. □

Then we can focus on the case that at most one query is allocated with leaf i.

In the next step we enumerate all possible queried CRS strings that touch leaf i. For every $\rho \in \{0,1\}^L$, we have

$$\left| \Pr_{C,t,s_t,\mathcal{H}'_{i-1} \leftarrow \mathcal{H}_{i-1}(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\mathcal{H}'_{i-1}}(1^\lambda, C, t, s_t) = 1 \\ \wedge \text{only queries } \rho \text{ to leaf } i \end{array} \right] - \Pr_{C,t,s_t,\mathcal{H}'_i \leftarrow \mathcal{H}_i(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\mathcal{H}'_i}(1^\lambda, C, t, s_t) = 1 \\ \wedge \text{only queries } \rho \text{ to leaf } i \end{array} \right] \right| \leq 2^{-2L}.$$

This is because if $\mathcal{D}$ distinguishes these two distributions, by generating $\{\mathsf{ct}_k \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)\}_{k>i}$ and simulating $\mathcal{D}^{\mathcal{H}'_i}$, we have a distinguisher to the following two distributions:

$$\left\{ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \{\tilde{\mathsf{ct}}_k, \tilde{\pi}^a_k\}_{k<i, a\in\{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ (\mathsf{ct}_i, \pi^\rho_i) \leftarrow \mathsf{HonestStep1}(\mathsf{pk}, i, \rho) \end{array} \right\}, \left\{ \begin{array}{c} \mathsf{pk} \leftarrow \mathcal{G}(1^{\lambda'}) \\ \{\tilde{\mathsf{ct}}_k, \tilde{\pi}^a_k\}_{k<i, a\in\{0,1\}^L} \leftarrow \tilde{\mathcal{S}}_{i-1}(\mathsf{pk}) \\ \tilde{\mathsf{ct}}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 0), \tilde{\pi}^\rho_i \leftarrow \mathcal{F}^\rho_i \end{array} \right\},$$

which would contradict with our construction and especially Corollary 3.4.

Take the summation over all $\rho \in \{0,1\}^L$, we conclude that

$$\left| \Pr_{C,t,s_t,\mathcal{H}'_{i-1} \leftarrow \mathcal{H}_{i-1}(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\mathcal{H}'_{i-1}}(1^\lambda, C, t, s_t) = 1 \\ \wedge \text{queries leaf } i \text{ at most once} \end{array} \right] - \Pr_{C,t,s_t,\mathcal{H}'_i \leftarrow \mathcal{H}_i(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\mathcal{H}'_i}(1^\lambda, C, t, s_t) = 1 \\ \wedge \text{queries leaf } i \text{ at most once} \end{array} \right] \right| \leq 2^{-L}.$$

The lemma follows by applying claim 4.8. □

Proof (Theorem 4.3). A hybrid argument with Lemma 4.7 implies that $\tilde{\mathcal{P}}$ can be efficiently simulated by $\mathcal{S}$, with distinguishing advantage bounded by $m(2T^2/m^2 + 2^{-L}) = 2T^2/m + m2^{-L} \leq 1/p(\lambda)$.

Take $\mathcal{D}^{(\cdot)}(\cdot)$ in Lemma 4.7 as the IVC verifier, i.e. the verifier induces a valid oracle distinguisher as follows: sample $(\mathsf{crs}_\mathcal{P}, \mathsf{crs}_\mathcal{V}) \leftarrow \Pi.\mathcal{G}(1^\lambda, C)$, query the oracle on $\mathsf{crs}_\mathcal{P}$ to obtain π_t, and output $\Pi.\mathcal{V}(\mathsf{crs}_\mathcal{V}, C, (t, s_t), \pi_t)$. In the $\mathcal{S}$ experiment, every oracle answer is an honest SNARG proof, so completeness makes this distinguisher output 1 with probability 1. Lemma 4.7 and the hybrid bound therefore imply that the same verifier accepts proofs from $\tilde{\mathcal{P}}$ with probability at least $1 - 2T^2/m - m2^{-L} \geq 1 - 1/p(\lambda)$. Since $\tilde{\mathcal{P}}$'s root encodes 0, the statement is false by the argument above, so this is a non-adaptive soundness attack. □

5 A Lower-bound of Black-box Non-uniform Reductions for SNARG

In this section, we prove a lower bound for non-uniform black-box reductions for non-adaptive SNARGs. Because non-uniform reductions involve nondeterministic advice, we first develop the required background on cryptography against nondeterministic adversaries in Section 5.1. We then state and prove the lower bound in Sections 5.2 and 5.3.

5.1 Cryptography Against Nondeterministic Adversary

To extend the scope of adversaries from polynomial-time algorithms to nondeterministic algorithms, we first clarify the type of nondeterminism considered in this work. We consider MA-type adversaries: that is, the adversary has access to random bits, but the nondeterministic advice is required to be independent of the randomness.

Definition 5.1 (MA-type algorithm). We say an algorithm is of MA-type, if its execution after taking input x can be described as the following two steps:

1. Take a nondeterministic advice z that can arbitrarily depend on x.
2. Perform (randomized) computation on (x, z).

The probability for an algorithm to accept an input x is the maximum accept probability over all possible nondeterministic advice z. ♢

We need the notion of super-bit generator first introduced by [Rud97], which strengthens the security of pseudorandom generator to nondeterministic adversary.

Definition 5.2 (Super-bit generator). We say efficient algorithm $\{g_n : \{0,1\}^n \to \{0,1\}^{l(n)}\}_n$ is a super-bit generator, if it has the following nondeterministic hardness: for every nondeterministic circuit D of polynomial size $s = s(n)$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $n \in \mathbb{N}$,

$$\Pr_{y \leftarrow \{0,1\}^{l(n)}}[D(y) = 1] - \Pr_{x \leftarrow \{0,1\}^n}[D(g_n(x)) = 1] \leq \mathsf{negl}(n).$$

We say g has $(s(n), \varepsilon(n))$ super-bit security if $\mathcal{U}_{\ell(n)}, g_n(\mathcal{U}_n)$ are $(s(n), \varepsilon(n))$-ND-IND in the one-sided sense. ♢

We stress that it is crucial the security definition of super-bit generator does not take absolute value on the left side, contrary to common indistinguishability-based definitions. The reason is a nondeterministic adversary can guess the preimage of the input, and therefore accepts $g_n(\mathcal{U}_n)$ with significantly higher probability than $\mathcal{U}_{l(n)}$.

[Rud97] conjectured the existence of super-bit generator with sub-exponential security and proposed a candidate construction based on subset-sum problem. The lattice-based assumption proposed by [CL24] also implies super-bit generator.

We abstract out the security notion of super-bit generator as ND-indistinguishability.

Definition 5.3 (ND-indistinguishability, or ND-IND). For any two distributions (X, Y), we say they are (s, ε)-ND-indistinguishable if for every nondeterministic circuit D of size s,

$$\Pr_{x \leftarrow X}[D(x) = 1] - \Pr_{y \leftarrow Y}[D(y) = 1] \leq \varepsilon.$$

♢

Finally, we introduce an alternative (but equivalent) security notion for ND-algorithms, that says any auxiliary information for one distribution can be translated into auxiliary information for the other distribution. Similarly, the order of these two distributions cannot be reversed.

Definition 5.4. We say two distributions X, Y are weakly (S, ε)-AI-IND, if for any joint distribution $X' = (x, z)$ where the marginal distribution of x is X and size-S randomized distinguisher D, there exists a joint distribution $Y' = (y, \tilde{z})$ where the marginal distribution of y is Y, such that

$$\Pr_{(x,z) \leftarrow X'}[D(x, z) = 1] - \Pr_{(y,\tilde{z}) \leftarrow Y'}[D(y, \tilde{z}) = 1] \leq \varepsilon.$$

♢

Lemma 5.5. For any two distributions $\mathcal{D}_1, \mathcal{D}_2$, the following two conditions are equivalent:

- $\mathcal{D}_1, \mathcal{D}_2$ are (S, ε)-ND-IND;
- $\mathcal{D}_1, \mathcal{D}_2$ are weakly (S, ε)-AI-IND. ♢

Proof. Denote $\mathsf{Size}(S)$ as the set of (randomized) circuits of size S, and F as the set of functions that maps an instance to an advice. Also, for any set T, let $\mathcal{D}_T$ be family of distributions over T.

Note that every distribution over F induces a mapping from instance to a distribution over advice. We write $z \sim \mathcal{D}(x)$ as sampling z from the distribution mapped from instance x, where $\mathcal{D}$ is a distribution over F.

We have

$$\begin{aligned}
&\max_{C\in\mathsf{Size}(S)} \left(\max_{f_z\in F} \Pr_{x\leftarrow\mathcal{D}_1, z=f_z(x)}[C(x,z)=1] - \max_{f_{\tilde{z}}\in F} \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}=f_{\tilde{z}}(x)}[C(x,\tilde{z})=1] \right) \\
=& \max_{C\in\mathsf{Size}(S)} \left(\max_{\mathcal{D}_z\in\mathcal{D}_F} \Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C(x,z)=1] - \max_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F} \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C(x,\tilde{z})=1] \right) \\
=& \max_{C\in\mathsf{Size}(S)} \max_{\mathcal{D}_z\in\mathcal{D}_F} \min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F} \left(\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C(x,\tilde{z})=1] \right) \\
=& \max_{\mathcal{D}_z\in\mathcal{D}_F} \max_{C\in\mathsf{Size}(S)} \min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F} \left(\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C(x,\tilde{z})=1] \right).
\end{aligned}$$

The first row is the definition of maximum distinguish advantage by "ND-IND", and the last row is the definition of maximum distinguish advantage by "weakly AI-IND". □

The current definition of weak AI-IND is somewhat inconvenient because the simulated auxiliary-input distribution may depend on the distinguisher. The next lemma shows that, with only a polynomial loss in parameters, one can choose the simulated auxiliary-input distribution first and have it work for all size-S distinguishers.

Definition 5.6. We say two distributions $\mathcal{D}_1, \mathcal{D}_2$ are (S, ε)-AI-IND, if for any joint distribution $\mathcal{D}'_1 = (x, z)$ where the marginal distribution of x is $\mathcal{D}_1$, there exists a joint distribution $\mathcal{D}'_2 = (x, \tilde{z})$ where the marginal distribution of x is $\mathcal{D}_2$, such that for any size-S distinguisher D,

$$\left| \Pr_{(x,z)\leftarrow\mathcal{D}'_1}[D(x,z)=1] - \Pr_{(x,\tilde{z})\leftarrow\mathcal{D}'_2}[D(x,\tilde{z})=1] \right| \le \varepsilon.$$ ♢

Lemma 5.7. For any two family of distributions $\mathcal{D}_1 = \{\mathcal{D}_1^\lambda\}, \mathcal{D}_2 = \{\mathcal{D}_2^\lambda\}$ and functions $S = S(\lambda) \ge \lambda$, there is a fixed polynomial poly, that if $\mathcal{D}_1, \mathcal{D}_2$ are weakly $(\mathsf{poly}(S), 1/\mathsf{poly}(S))$-AI-IND, then $\mathcal{D}_1, \mathcal{D}_2$ are $(S, 1/S)$-AI-IND. ♢

Proof. We first show that, for large enough λ,

$$\max_{\mathcal{D}_z\in\mathcal{D}_F} \max_{\mathcal{D}_C\in\mathsf{Dist}(\mathsf{Size}(S+1))} \min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F} \left(\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x), C\sim\mathcal{D}_C}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x), C\sim\mathcal{D}_C}[C(x,\tilde{z})=1] \right) \le \frac{1}{S}. \tag{1}$$

Fix $\mathcal{D}_z \in \mathcal{D}_F$ and an arbitrary distribution $\mathcal{D}_C$ over size-$(S+1)$ circuits. We write the total length of $x \leftarrow \mathcal{D}_1$ and auxiliary input z being $\ell \le O(S)$.

We construct a circuit C' that simulates a random circuit from $\mathcal{D}_C$, by hard-coding $m = \mathsf{poly}(\ell, S)$ random samples from $\mathcal{D}_C$ into the description of C': C' evalutes all of these sampled circuits on the input and outputs 1 by the same probability as their average. By Chernoff's Bound, for every fixed input (x, z), with probability $1 - 2^{-(\ell+1)}$ over the generation of C',

$$|\Pr[C'(x,z)=1] - \Pr_{C\sim\mathcal{D}_C}[C(x,z)=1]| \leq \frac{1}{3S}.$$

By union bound, there exists some C' with size $O(m \cdot S) = \mathsf{poly}(S)$, such that for any (x, z), $|\Pr[C'(x,z)=1] - \Pr_{C\sim\mathcal{D}_C}[C(x,z)=1]| \leq \frac{1}{3S}$.

By applying the weakly $(\mathsf{poly}(S), 1/\mathsf{poly}(S))$-AI-IND of $\mathcal{D}_1, \mathcal{D}_2$ to C', for large enough λ, there exists $\mathcal{D}_{\tilde{z}} \in \mathcal{D}_F$ such that,

$$\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C'(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C'(x,\tilde{z})=1] \leq \frac{1}{3S}.$$

This implies that,

$$\begin{aligned}
&\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x), C\sim\mathcal{D}_C}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x), C\sim\mathcal{D}_C}[C(x,\tilde{z})=1] \\
\leq &\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C'(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C'(x,\tilde{z})=1] + 2\max_{(u,v)}\left|\Pr[C'(u,v)=1] - \Pr_{C\sim\mathcal{D}_C}[C(u,v)=1]\right| \\
\leq &\frac{1}{3S} + 2\cdot\frac{1}{3S} = \frac{1}{S}.
\end{aligned}$$

Then we show how to use inequality 1 to imply $(S, 1/S)$-AI-IND against adversaries with auxiliary input. Fix $\mathcal{D}_z \in \mathcal{D}_F$. We write

$$\Delta := \Pr_{\substack{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x),\\ C\sim\mathcal{D}_C}}[C(x,z)=1] - \Pr_{\substack{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x),\\ C\sim\mathcal{D}_C}}[C(x,\tilde{z})=1].$$

We observe that, by von Neumann's min-max theorem [Neu28],

$$\begin{aligned}
&\max_{\mathcal{D}_z\in\mathcal{D}_F}\max_{\mathcal{D}_C\in\mathsf{Dist}(\mathsf{Size}(S+1))}\min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F}\Delta \\
=&\max_{\mathcal{D}_z\in\mathcal{D}_F}\min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F}\max_{\mathcal{D}_C\in\mathsf{Dist}(\mathsf{Size}(S+1))}\Delta,
\end{aligned}$$

as $\mathcal{D}_Z$, $\mathcal{D}_C$ can be regarded as mixed strategies over F and $\mathsf{Size}(S+1)$ respectively.

By inequality 1, we have

$$\begin{aligned}
\frac{1}{S} &\geq \max_{\mathcal{D}_z\in\mathcal{D}_F}\min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F}\max_{\mathcal{D}_C\in\mathsf{Dist}(\mathsf{Size}(S+1))}\Delta \\
&\geq \max_{\mathcal{D}_z\in\mathcal{D}_F}\min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F}\max_{C\in\mathsf{Size}(S+1)}\left(\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C(x,\tilde{z})=1]\right) \\
&\geq \max_{\mathcal{D}_z\in\mathcal{D}_F}\min_{\mathcal{D}_{\tilde{z}}\in\mathcal{D}_F}\max_{C\in\mathsf{Size}(S)}\left|\Pr_{x\leftarrow\mathcal{D}_1, z\sim\mathcal{D}_z(x)}[C(x,z)=1] - \Pr_{x\leftarrow\mathcal{D}_2, \tilde{z}\sim\mathcal{D}_{\tilde{z}}(x)}[C(x,\tilde{z})=1]\right|,
\end{aligned}$$

where the last inequality is by the fact that we can always hard-code a bit into C to flip its output.□

Corollary 5.8. For any two family of distributions $\mathcal{D}_1 = \{\mathcal{D}_1^\lambda\}, \mathcal{D}_2 = \{\mathcal{D}_2^\lambda\}$ and constant $c > 0$, if for any $S = 2^{O(n^c)}$, $\mathcal{D}_1, \mathcal{D}_2$ are weakly $(S, 1/S)$-AI-IND, then for any $S = 2^{O(n^c)}$, $\mathcal{D}_1, \mathcal{D}_2$ are $(S, 1/S)$-AI-IND. ♢

5.2 A Simulatable Adversary Fooling Query-bounded Distinguishers

We prove the lower bound for non-adaptive SNARGs, for non-uniform black-box reductions that make a small number of non-adaptive queries to the adversary during the security game.

Theorem 5.9. Assume the existence of super-bit generator (Definition 5.2) with $(2^{n^c}, 2^{-n^c})$ security. Let Π be any SNARG scheme with $|\mathsf{crs}|+|\pi| = L(n)$. Let $\mathcal{D}^{(\cdot)}(\cdot)$ be an oracle aided distinguisher that runs in polynomial time and makes *at most q non-adaptive queries* to its oracle, and F be a randomized *inefficient* function that maps an instance to its advice. Then as long as $Lq \leq n^{c'}$ for some constant $c' < c$, there exists polynomial $n = n(\lambda)$ that there is a machine $\tilde{\mathcal{P}}$ called a simulatable Π-adversary satisfying the following:

- $\tilde{\mathcal{P}}$ is a computationally unbounded Π-adversary. On input 1^λ it outputs x of size n and a (computationally unbounded) machine $\tilde{\mathcal{P}}'$, such that x and $\tilde{\mathcal{P}}'$ breaks the non-adaptive soundness of Π with probability $1/2$.
- $\tilde{\mathcal{P}}$ is poly-time simulatable against *non-uniform* reduction algorithms. That is, there exists a simulator $\mathcal{S}$ that on input 1^λ outputs x and an *efficient* machine $\mathcal{S}'$, and a randomized inefficient function G that maps an instance to its advice, such that

$$\left| \Pr_{x,\tilde{\mathcal{P}}' \leftarrow \tilde{\mathcal{P}}(1^\lambda)} [\mathcal{D}^{\tilde{\mathcal{P}}'}(1^\lambda, x, F(x)) = 1] - \Pr_{x,\mathcal{S}' \leftarrow \mathcal{S}(1^\lambda)} [\mathcal{D}^{\mathcal{S}'}(1^\lambda, x, G(x)) = 1] \right| \leq \mathsf{negl}(\lambda). \qquad \diamondsuit$$

Similar to the proofs of Theorem 4.4 and the Gentry-Wichs lower bound [GW11], the simulatable adversary implies a barrier for constructing SNARGs.

Theorem 5.10. Assume the existence of super-bit generator (Definition 5.2) with sub-exponential security. Then any SNARG with instance-independent crs and proof size does not have a polynomial time non-uniform black-box security reduction that only makes $\mathsf{poly}(\lambda)$ non-adaptive oracle queries during the security game, unless the assumptions are not secure against non-uniform poly-time algorithms. $\diamondsuit$

Proof. Take statement size n as specified by Theorem 5.9, and $\mathcal{D}^{(\cdot)}(\cdot)$ as the combined algorithm of the reduction algorithm $\mathcal{R}$ and the falsifiable assumption security game. Since $\mathcal{R}$ makes only non-adaptive oracle queries, this combined algorithm is a non-adaptive q-query distinguisher for some polynomial $q = q(\lambda)$. Let F be the inefficient function that decides the advice z of any instance x.

Then take $\tilde{\mathcal{P}}$ and $\mathcal{S}$ as specified by Theorem 5.9:

- $\mathcal{R}$ supplied with $\tilde{\mathcal{P}}$ breaks the assumption with winning advantage $1/\mathsf{poly}(\lambda)$, as $\tilde{\mathcal{P}}$ breaks non-adaptive security of the SNARG scheme.
- As $\mathcal{S}$ simulates $\tilde{\mathcal{P}}$, there exists an inefficient function G that $\mathcal{R}$ supplied with $\mathcal{S}$ and advice $\tilde{z} \sim G(x)$ also breaks the assumption with winning advantage $1/\mathsf{poly}(\lambda)$.

Although $\tilde{z} \sim G(x)$ may not be efficiently computable in the last step, we can fix the random coin over the best choice of $(x, \mathcal{S}', \tilde{z} \sim G(x))$ by non-uniformity. Therefore we conclude that the non-uniform poly-time algorithm $\mathcal{R}^{\mathcal{S}'}(1^\lambda, x, \tilde{z})$ breaks the underlying assumption with $1/\mathsf{poly}(\lambda)$ winning advantage. $\square$

5.3 Proof of Theorem 5.9

Our construction of the NP language is similar to that of [GW11]: we take the language of PRG (in our case, super-bit generator) images, that is, $x \in \mathcal{L}$ iff there is some y such that $x = g(y)$. Looking ahead, $\tilde{\mathcal{P}}$ will choose x as a random string, and $\mathcal{S}$ will choose x as a super-bit generator image. While $\mathcal{S}$ can compute SNARG proof honestly, $\tilde{\mathcal{P}}$ generates "simulated leakage" as its proofs.

Ingredients. We write the super-bit generator as $g : \{0,1\}^{n/2} \to \{0,1\}^n$. Note that super-bit generator can be stretched as shown by [Rud97].

Parameter Settings.

- We use $L = L(\lambda, n)$ to bound the length of proof π and crs in the SNARG scheme Π.
- Write $\tilde{L} = Lq$, where q is the query complexity of the distinguisher.
- We take $n = \mathsf{poly}(\lambda)$ such that the super-bit generator has $(2^{O(\tilde{L}^{1+\alpha})}, 2^{-O(\tilde{L}^{1+\alpha})})$-ND-IND for some constant $\alpha > 0$.

Construction. The language $\mathcal{L}$ is taken as the image of G, and the witness is the preimage. We describe $\mathcal{S}$ first.

Algorithm of simulator $\mathcal{S}$

1. Sample $y \leftarrow \{0,1\}^{n/2}$ and compute $x = g(y)$. Output x as the instance.
2. Output $\mathcal{S}'$ with x, y hard-coded in, that performs the following computation on an ordered CRS list $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$ for $t \le q$:
   (a) Honestly compute and output $(\pi^{\rho_j})_{j \in [t]}$, where $\pi^{\rho_j} = \Pi.\mathcal{P}(\rho_j, x, y)$.

Figure 3: Algorithm of the efficient simulator $\mathcal{S}$.

Before describing $\tilde{\mathcal{P}}$, we first introduce a few distributions. Let $\mathcal{D}_1 = \mathcal{U}_n$, and $\mathcal{D}_2 = g(\mathcal{U}_{n/2})$. The non-uniform reduction advice introduces an augmented distribution $\mathcal{D}_1' = \{(x, F(x)) : x \leftarrow \mathcal{U}_n\}$. By definition of AI-IND, and its equivalence to ND-IND (see Lemma 5.5 and Corollary 5.8), there exists a randomized inefficient function G, such that with

$$\mathcal{D}_2' := \left\{(x, G(x)) : x \leftarrow g(\mathcal{U}_{n/2})\right\},$$

$\mathcal{D}_1', \mathcal{D}_2'$ are $(2^{O(\tilde{L}^{1.1})}, 2^{-O(\tilde{L}^{1.1})})$-indistinguishable.

The detailed description of $\tilde{\mathcal{P}}$ is given in Figure 4.

Analysis. We show that $\tilde{\mathcal{P}}$ and $\mathcal{S}$ constructed as above satisfies the requirement of Theorem 5.9.

Algorithm of inefficient prover $\tilde{\mathcal{P}}$

1. Sample $x \leftarrow \{0,1\}^n$ and output x as the instance.
2. For every possible ordered size-q crs list $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_q) \in (\{0,1\}^L)^q$, sample $(\tilde{\pi}^{\rho_j}_{\boldsymbol{\rho}})_{j\in[q]}$ from a $2^{O(\tilde{L})}$-size circuit $\mathcal{F}^{\boldsymbol{\rho}}$ which takes input (x, z), such that

$$\left\{ (x, z, (\pi^{\rho_j})_{j\in[q]}) : \begin{array}{c} y \leftarrow \mathcal{U}_{n/2} \\ x = g(y) \\ z = G(x) \\ \pi^{\rho_j} = \Pi.\mathcal{P}(\rho_j, x, y) \end{array} \right\}, \left\{ (x, z, (\tilde{\pi}^{\rho_j}_{\boldsymbol{\rho}})_{j\in[q]}) \begin{array}{c} x \leftarrow \mathcal{U}_n \\ z = F(x) \\ (\tilde{\pi}^{\rho_j}_{\boldsymbol{\rho}})_{j\in[q]} = \mathcal{F}^{\boldsymbol{\rho}}(x, z) \end{array} \right\}$$

   are $(2^{2\tilde{L}}, 2^{-2\tilde{L}})$-indistinguishable. The existence of such circuit $\mathcal{F}^{\boldsymbol{\rho}}$ is guaranteed by the fact that $\mathcal{D}'_1, \mathcal{D}'_2$ are $(2^{O(\tilde{L}^{1.1})}, 2^{-O(\tilde{L}^{1.1})})$-indistinguishable, and Corollary 3.4.
3. Output $\tilde{\mathcal{P}}'$ hard-coded with all tuples $(\tilde{\pi}^{\rho_j}_{\boldsymbol{\rho}})_{j\in[q]}$ for all possible $\boldsymbol{\rho} \in (\{0,1\}^L)^q$. On a batched oracle query $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$ for $t \le q$, pad $\boldsymbol{\rho}$ to length q, look up the corresponding ordered tuple, and return its first t simulated proofs. Repeated CRS strings are handled by their positions in the ordered list.

Figure 4: Algorithm of the inefficient prover $\tilde{\mathcal{P}}$.

Lemma 5.11. For any non-adaptive q-query distinguisher $\mathcal{D}^{(\cdot)}(\cdot)$ that runs in polynomial time, where its oracle access is viewed as one ordered batched query in $(\{0,1\}^L)^q$,

$$\left| \Pr_{x,\tilde{\mathcal{P}}'\leftarrow\tilde{\mathcal{P}}(1^\lambda)} [\mathcal{D}^{\tilde{\mathcal{P}}'}(1^\lambda, x, F(x)) = 1] - \Pr_{x,\mathcal{S}'\leftarrow\mathcal{S}(1^\lambda)} [\mathcal{D}^{\mathcal{S}'}(1^\lambda, x, G(x)) = 1] \right| \le \mathsf{negl}(\lambda). \qquad \diamondsuit$$

Proof. We enumerate over all possible ordered lists of oracle queries $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_q) \in (\{0,1\}^L)^q$. For a fixed $\boldsymbol{\rho}$, consider the distinguisher that emulates $\mathcal{D}$ but aborts if the non-adaptive oracle query list is not $\boldsymbol{\rho}$. Since the event that $\mathcal{D}$ chooses $\boldsymbol{\rho}$ is determined before any oracle answer is seen, the construction of $\tilde{\mathcal{P}}$ gives

$$\left| \Pr_{x,\tilde{\mathcal{P}}'\leftarrow\tilde{\mathcal{P}}(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\tilde{\mathcal{P}}'}(1^\lambda, x, F(x)) = 1 \\ \wedge \text{ the query batch is } \boldsymbol{\rho} \end{array} \right] - \Pr_{x,\mathcal{S}'\leftarrow\mathcal{S}(1^\lambda)} \left[\begin{array}{c} \mathcal{D}^{\mathcal{S}'}(1^\lambda, x, G(x)) = 1 \\ \wedge \text{ the query batch is } \boldsymbol{\rho} \end{array} \right] \right| \le 2^{-2\tilde{L}}.$$

Summing up over all $2^{Lq} = 2^{\tilde{L}}$ possible query batches yields the lemma. □

Given the above lemma, $\tilde{\mathcal{P}}$ is indistinguishable from $\mathcal{S}$ for every non-adaptive q-query distinguisher. The ordinary SNARG non-adaptive soundness experiment is the special case of a one-query distinguisher: it samples $(\mathsf{crs}_{\mathcal{P}}, \mathsf{crs}_{\mathcal{V}}) \leftarrow \Pi.\mathcal{G}(1^\lambda)$, queries the prover on $\mathsf{crs}_{\mathcal{P}}$, and runs the verifier. In the $\mathcal{S}$ experiment, completeness makes this verifier accept with probability 1. Indistinguishability therefore implies that it also accepts proofs produced by $\tilde{\mathcal{P}}$ with probability $1 - \mathsf{negl}(\lambda)$. Since $\tilde{\mathcal{P}}$ samples $x \leftarrow \{0,1\}^n$, the instance is outside the image of $g : \{0,1\}^{n/2} \to \{0,1\}^n$ except with probability at most $2^{-n/2}$, so $\tilde{\mathcal{P}}$ breaks non-adaptive soundness with probability at least $1/2$.

Acknowledgment

We thank the anonymous FOCS reviewers for their helpful feedback. Zhengzhong Jin thanks Sanjam Garg and Abhishek Jain for initial discussions about this problem. Part of this research was conducted while Liyan Chen was at Tsinghua University.

References

[BKKSW18] Saikrishna Badrinarayanan, Yael Tauman Kalai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. "Succinct delegation for low-space non-deterministic computation". In: *50th ACM STOC*. Ed. by Ilias Diakonikolas, David Kempe, and Monika Henzinger. ACM Press, June 2018, pp. 709–721. DOI: 10.1145/3188745.3188924 (cit. on p. 1).

[BDKP22] Shany Ben-David, Yael Tauman Kalai, and Omer Paneth. "Verifiable Private Information Retrieval". In: *TCC 2022, Part III*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Vol. 13749. LNCS. Springer, Cham, Nov. 2022, pp. 3–32. DOI: 10.1007/978-3-031-22368-6_1 (cit. on p. 1).

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 459–474. DOI: 10.1109/SP.2014.36 (cit. on p. 1).

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. "Recursive composition and bootstrapping for SNARKS and proof-carrying data". In: *45th ACM STOC*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 111–120. DOI: 10.1145/2488608.2488623 (cit. on p. 2).

[BDFH08] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. "On Problems without Polynomial Kernels (Extended Abstract)". In: *ICALP 2008, Part I*. Ed. by Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz. Vol. 5125. LNCS. Springer, Berlin, Heidelberg, July 2008, pp. 563–574. DOI: 10.1007/978-3-540-70575-8_46 (cit. on p. 12).

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. *Recursive Proof Composition without a Trusted Setup*. Cryptology ePrint Archive, Paper 2019/1021. 2019. URL: https://eprint.iacr.org/2019/1021 (cit. on p. 2).

[BBKLP23] Zvika Brakerski, Maya Farber Brodsky, Yael Tauman Kalai, Alex Lombardi, and Omer Paneth. "SNARGs for Monotone Policy Batch NP". In: *CRYPTO 2023, Part II*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14082. LNCS. Springer, Cham, Aug. 2023, pp. 252–283. DOI: 10.1007/978-3-031-38545-2_9 (cit. on pp. 3, 12, 14).

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. "Non-interactive delegation and batch NP verification from standard computational assumptions". In: *49th ACM STOC*. Ed. by Hamed Hatami, Pierre McKenzie, and Valerie King. ACM Press, June 2017, pp. 474–482. DOI: 10.1145/3055399.3055497 (cit. on p. 1).

[BCMS20] Benedikt Bünz, Alessandro Chiesa, Pratyush Mishra, and Nicholas Spooner. "Recursive Proof Composition from Accumulation Schemes". In: *TCC 2020, Part II*. Ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12551. LNCS. Springer, Cham, Nov. 2020, pp. 1–18. DOI: 10.1007/978-3-030-64378-2_1 (cit. on p. 2).

[CGKS23] Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. "Impossibilities in Succinct Arguments: Black-Box Extraction and More". In: *AFRICACRYPT 23*. Ed. by Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne. Vol. 14064. LNCS. Springer, Cham, July 2023, pp. 465–489. DOI: 10.1007/978-3-031-37679-5_20 (cit. on pp. 1, 4).

[CCL18] Yi-Hsiu Chen, Kai-Min Chung, and Jyun-Jie Liao. "On the Complexity of Simulating Auxiliary Input". In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Cham, 2018, pp. 371–390. DOI: 10.1007/978-3-319-78372-7_12 (cit. on pp. 8, 15).

[CL24] Yilei Chen and Jiatu Li. "Hardness of Range Avoidance and Remote Point for Restricted Circuits via Cryptography". In: *56th ACM STOC*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM Press, June 2024, pp. 620–629. DOI: 10.1145/3618260.3649602 (cit. on pp. 3, 27).

[CT10] Alessandro Chiesa and Eran Tromer. "Proof-Carrying Data and Hearsay Arguments from Signature Cards". In: *ICS 2010*. Ed. by Andrew Chi-Chih Yao. Tsinghua University Press, Jan. 2010, pp. 310–331 (cit. on pp. 1, 2).

[CGJJZ23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. "Correlation Intractability and SNARGs from Sub-exponential DDH". In: *CRYPTO 2023, Part IV*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14084. LNCS. Springer, Cham, Aug. 2023, pp. 635–668. DOI: 10.1007/978-3-031-38551-3_20 (cit. on p. 1).

[CHK+19] Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. "Finding a Nash equilibrium is no easier than breaking Fiat-Shamir". In: *51st ACM STOC*. Ed. by Moses Charikar and Edith Cohen. ACM Press, June 2019, pp. 1103–1114. DOI: 10.1145/3313276.3316400 (cit. on p. 1).

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. "Non-interactive Batch Arguments for NP from Standard Assumptions". In: *CRYPTO 2021, Part IV*. Ed. by Tal Malkin and Chris Peikert. Vol. 12828. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 394–423. DOI: 10.1007/978-3-030-84259-8_14 (cit. on pp. 1, 3).

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. "SNARGs for $\mathcal{P}$ from LWE". In: *62nd FOCS*. IEEE Computer Society Press, Feb. 2022, pp. 68–79. DOI: 10.1109/FOCS52979.2021.00016 (cit. on pp. 1, 3, 12, 13).

[DJJ+25] Pratish Datta, Abhishek Jain, Zhengzhong Jin, Alexis Korb, Surya Mathialagan, and Amit Sahai. "Incrementally Verifiable Computation for NP from Standard Assumptions". In: *Advances in Cryptology – CRYPTO 2025*. To appear. 2025 (cit. on p. 2).

[Del16] Holger Dell. "AND-compression of NP-complete problems: Streamlined proof and minor observations". In: *Algorithmica* 75 (2016), pp. 403–423 (cit. on pp. 12, 20).

[DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. "Rate-1 Non-Interactive Arguments for Batch-NP and Applications". In: *63rd FOCS*. IEEE Computer Society Press, 2022, pp. 1057–1068. DOI: 10.1109/FOCS54457.2022.00103 (cit. on p. 4).

[Dru12] Andrew Drucker. "New Limits to Classical and Quantum Instance Compression". In: *53rd FOCS*. IEEE Computer Society Press, Oct. 2012, pp. 609–618. DOI: 10.1109/FOCS.2012.71 (cit. on pp. 12, 20).

[FS08] Lance Fortnow and Rahul Santhanam. "Infeasibility of instance compression and succinct PCPs for NP". In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 133–142. DOI: 10.1145/1374376.1374398 (cit. on pp. 12, 20).

[GW11] Craig Gentry and Daniel Wichs. "Separating succinct non-interactive arguments from all falsifiable assumptions". In: *43rd ACM STOC*. Ed. by Lance Fortnow and Salil P. Vadhan. ACM Press, June 2011, pp. 99–108. DOI: 10.1145/1993636.1993651 (cit. on pp. 1, 2, 4, 5, 17, 20, 30, 31).

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. "Delegating Computation: Interactive Proofs for Muggles". In: *J. ACM* 62.4 (2015), 27:1–27:64 (cit. on p. 1).

[HAN23] Mathias Hall-Andersen and Jesper Buus Nielsen. "On Valiant's Conjecture: Impossibility of Incrementally Verifiable Computation from Random Oracles". In: *EUROCRYPT 2023, Part II*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14005. LNCS. Springer, Cham, Apr. 2023, pp. 438–469. DOI: 10.1007/978-3-031-30617-4_15 (cit. on p. 4).

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. "SNARGs for P from Sub-exponential DDH and QR". In: *EUROCRYPT 2022, Part II*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13276. LNCS. Springer, Cham, 2022, pp. 520–549. DOI: 10.1007/978-3-031-07085-3_18 (cit. on p. 1).

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability obfuscation from well-founded assumptions". In: *53rd ACM STOC*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM Press, June 2021, pp. 60–73. DOI: 10.1145/3406325.3451093 (cit. on p. 4).

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability Obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0". In: *EUROCRYPT 2022, Part I*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13275. LNCS. Springer, Cham, 2022, pp. 670–699. DOI: 10.1007/978-3-031-06944-4_23 (cit. on p. 4).

[JKKZ21] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Yun Zhang. "SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE". In: *53rd ACM STOC*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM Press, June 2021, pp. 708–721. DOI: 10.1145/3406325.3451055 (cit. on p. 1).

[JP14] Dimitar Jetchev and Krzysztof Pietrzak. "How to Fake Auxiliary Input". In: *TCC 2014*. Ed. by Yehuda Lindell. Vol. 8349. LNCS. Springer, Berlin, Heidelberg, Feb. 2014, pp. 566–590. DOI: 10.1007/978-3-642-54242-8_24 (cit. on p. 15).

[JKLV24] Zhengzhong Jin, Yael Kalai, Alex Lombardi, and Vinod Vaikuntanathan. "SNARGs under LWE via Propositional Proofs". In: *56th ACM STOC*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM Press, June 2024, pp. 1750–1757. DOI: 10.1145/3618260.3649770 (cit. on pp. 2, 3, 12, 14, 18).

[JKLM25] Zhengzhong Jin, Yael Tauman Kalai, Alex Lombardi, and Surya Mathialagan. "Universal SNARGs for NP from Proofs of Correctness". In: *57th ACM STOC*. Ed. by Michal Koucký and Nikhil Bansal. ACM Press, June 2025, pp. 933–943. DOI: 10.1145/3717823.3718104 (cit. on pp. 1, 3, 4, 14).

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. "Boosting Batch Arguments and RAM Delegation". In: *55th ACM STOC*. Ed. by Barna Saha and Rocco A. Servedio. ACM Press, June 2023, pp. 1545–1552. DOI: 10.1145/3564246.3585200 (cit. on p. 1).

[KLV23] Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. "SNARGs and PPAD Hardness from the Decisional Diffie-Hellman Assumption". In: *EUROCRYPT 2023, Part II*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14005. LNCS. Springer, Cham, Apr. 2023, pp. 470–498. DOI: 10.1007/978-3-031-30617-4_16 (cit. on p. 1).

[KPY19] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. "How to delegate computations publicly". In: *51st ACM STOC*. Ed. by Moses Charikar and Edith Cohen. ACM Press, June 2019, pp. 1115–1124. DOI: 10.1145/3313276.3316411 (cit. on p. 1).

[KPY20] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. "Delegation with Updatable Unambiguous Proofs and PPAD-Hardness". In: *CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, Cham, Aug. 2020, pp. 652–673. DOI: 10.1007/978-3-030-56877-1_23 (cit. on p. 1).

[KRR13] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. "Delegation for bounded space". In: *45th ACM STOC*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 565–574. DOI: 10.1145/2488608.2488679 (cit. on p. 1).

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. "How to delegate computations: the power of no-signaling proofs". In: *46th ACM STOC*. Ed. by David B. Shmoys. ACM Press, 2014, pp. 485–494. DOI: 10.1145/2591796.2591809 (cit. on p. 1).

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. "Somewhere Statistical Soundness, Post-Quantum Security, and SNARGs". In: *TCC 2021, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. LNCS. Springer, Cham, Nov. 2021, pp. 330–368. DOI: 10.1007/978-3-030-90459-3_12 (cit. on p. 1).

[Kiy22] Susumu Kiyoshima. "Public-Coin 3-Round Zero-Knowledge from Learning with Errors and Keyless Multi-Collision-Resistant Hash". In: *CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. LNCS. Springer, Cham, Aug. 2022, pp. 444–473. DOI: 10.1007/978-3-031-15802-5_16 (cit. on p. 1).

[KS22] Abhiram Kothapalli and Srinath Setty. *SuperNova: Proving universal machine executions without universal circuits*. Cryptology ePrint Archive, Paper 2022/1758. 2022. URL: https://eprint.iacr.org/2022/1758 (cit. on p. 2).

[KS23] Abhiram Kothapalli and Srinath Setty. *CycleFold: Folding-scheme-based recursive arguments over a cycle of elliptic curves*. Cryptology ePrint Archive, Paper 2023/1192. 2023. URL: https://eprint.iacr.org/2023/1192 (cit. on p. 2).

[KST22] Abhiram Kothapalli, Srinath Setty, and Ioanna Tzialla. "Nova: Recursive Zero-Knowledge Arguments from Folding Schemes". In: *CRYPTO 2022, Part IV*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13510. LNCS. Springer, Cham, Aug. 2022, pp. 359–388. DOI: 10.1007/978-3-031-15985-5_13 (cit. on p. 2).

[KS24] Abhiram Kothapalli and Srinath T. V. Setty. "HyperNova: Recursive Arguments for Customizable Constraint Systems". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, Aug. 2024, pp. 345–379. DOI: 10.1007/978-3-031-68403-6_11 (cit. on p. 2).

[MPV24] Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. "Adaptively Sound Zero-Knowledge SNARKs for UP". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, Aug. 2024, pp. 38–71. DOI: 10.1007/978-3-031-68403-6_2 (cit. on p. 6).

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM J. Comput.* 30.4 (2000), pp. 1253–1298 (cit. on p. 1).

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges (Invited Talk)". In: *CRYPTO 2003*. Ed. by Dan Boneh. Vol. 2729. LNCS. Springer, Berlin, Heidelberg, Aug. 2003, pp. 96–109. DOI: 10.1007/978-3-540-45146-4_6 (cit. on pp. 1, 3).

[Neu28] J. von Neumann. "Zur Theorie der Gesellschaftsspiele". In: *Mathematische Annalen* 100 (1928), pp. 295–320. URL: http://eudml.org/doc/159291 (cit. on pp. 11, 29).

[PP22] Omer Paneth and Rafael Pass. "Incrementally Verifiable Computation via Rate-1 Batch Arguments". In: *63rd FOCS*. IEEE Computer Society Press, 2022, pp. 1045–1056. DOI: 10.1109/FOCS54457.2022.00102 (cit. on p. 4).

[RVV24] Seyoon Ragavan, Neekon Vafa, and Vinod Vaikuntanathan. "Indistinguishability Obfuscation from Bilinear Maps and LPN Variants". In: *Theory of Cryptography: 22nd International Conference, TCC 2024, Milan, Italy, December 2–6, 2024, Proceedings, Part IV*. Milan, Italy: Springer-Verlag, 2024, 3–36. ISBN: 978-3-031-78022-6. DOI: 10.1007/978-3-031-78023-3_1. URL: https://doi.org/10.1007/978-3-031-78023-3_1 (cit. on p. 4).

[RR97] Alexander A. Razborov and Steven Rudich. "Natural Proofs". In: *J. Comput. Syst. Sci.* 55.1 (1997), pp. 24–35 (cit. on p. 3).

[Rud97] Steven Rudich. "Super-bits, Demi-bits, and NP/qpoly-natural Proofs". In: *RANDOM*. Vol. 1269. Lecture Notes in Computer Science. Springer, 1997, pp. 85–93 (cit. on pp. 11, 27, 31).

[SW21] Amit Sahai and Brent Waters. "How to Use Indistinguishability Obfuscation: Deniable Encryption, and More". In: *SIAM J. Comput.* 50.3 (2021), pp. 857–908 (cit. on pp. 1, 4, 12, 13).

[Skó16] Maciej Skórski. "Simulating Auxiliary Inputs, Revisited". In: *TCC 2016-B, Part I*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9985. LNCS. Springer, Berlin, Heidelberg, 2016, pp. 159–179. DOI: 10.1007/978-3-662-53641-4_7 (cit. on p. 15).

[VZ13] Salil P. Vadhan and Colin Jia Zheng. "A Uniform Min-Max Theorem with Applications in Cryptography". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Berlin, Heidelberg, Aug. 2013, pp. 93–110. DOI: 10.1007/978-3-642-40041-4_6 (cit. on pp. 8, 15).

[Val08] Paul Valiant. "Incrementally Verifiable Computation or Proofs of Knowledge Imply Time/Space Efficiency". In: *TCC 2008*. Ed. by Ran Canetti. Vol. 4948. LNCS. Springer, Berlin, Heidelberg, Mar. 2008, pp. 1–18. DOI: 10.1007/978-3-540-78524-8_1 (cit. on pp. 1, 2).

[WW22] Brent Waters and David J. Wu. "Batch Arguments for NP and More from Standard Bilinear Group Assumptions". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Cham, Aug. 2022, pp. 433–463. DOI: 10.1007/978-3-031-15979-4_15 (cit. on p. 1).

[WW24a] Brent Waters and David J. Wu. *A Pure Indistinguishability Obfuscation Approach to Adaptively-Sound SNARGs for NP*. Cryptology ePrint Archive, Paper 2024/933. 2024. URL: https://eprint.iacr.org/2024/933 (cit. on pp. 1, 4).

[WW24b] Brent Waters and David J. Wu. "Adaptively-Sound Succinct Arguments for NP from Indistinguishability Obfuscation". In: *56th ACM STOC*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM Press, June 2024, pp. 387–398. DOI: 10.1145/3618260.3649671 (cit. on pp. 1, 4).

[WZ24] Brent Waters and Mark Zhandry. "Adaptive Security in SNARGs via iO and Lossy Functions". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, Aug. 2024, pp. 72–104. DOI: 10.1007/978-3-031-68403-6_3 (cit. on pp. 1, 4).