Unique SNARGs with Adaptive Security: Constructions and Black-Box Separations

Cody Freitag*
Northeastern University and
Hebrew University of Jerusalem
codyrfreitag@gmail.com

Daniel Wichs†
Northeastern University
and NTT Research
wichs@ccs.neu.edu

Abstract

Succinct non-interactive arguments (SNARGs) for NP allow an efficient prover to convince a verifier that an NP statement is true with a proof that is much shorter than the original NP witness. Gentry and Wichs (STOC '11) showed that adaptive soundness of such SNARGs cannot be proven via a black-box reduction from any falsifiable assumption. However, recent works by Waters, Wu and Zhandry (STOC '24, CRYPTO '24, CRYPTO '25) circumvent this negative result by relying on subexponential hardness assumptions and having a long common reference string (CRS) that is longer than the statement size.

In this work, we study *unique* SNARGs where each statement has at most one accepting proof. The above constructions of adaptively sound SNARGs are *not* unique and crucially rely on the existence of multiple valid proofs in their security analysis. We explore to what extent this is inherent as follows:

- On the negative side, we give a strengthened Gentry-Wichs style black-box separation for the case of perfectly complete and unique SNARGs for NP with adaptive security. Our black-box separation extends *even* to reductions that rely on subexponentially hard falsifiable assumptions *and* to SNARGs that have an arbitrarily long CRS.
- On the positive side, we construct perfectly unique and adaptively secure SNARGs for NP with a long CRS based on subexponentially hard one-way functions and indistinguishability obfuscation. We do so by relaxing *perfect completeness* and allowing for a negligible completeness error. This is the first unique SNARG with a proof of adaptive security from falsifiable assumptions, even restricted to P.

*Supported in part by a Khoury College Distinguished Postdoctoral Fellowship and the European Union (ERC, SCALE, 101162665). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

†Research supported by NSF grants CNS-2349972 and CNS-2055510.

1 Introduction

SNARGs. A succinct non-interactive argument (SNARG) for NP gives a way for an efficient prover to convince a verifier that an NP statement x is true with a proof π that is much shorter than the original NP witness w or the statement x. More formally, a SNARG must satisfy *completeness* and *soundness*: an honest prover should give convincing proofs for true statements, yet a (computationally-bounded) cheating prover should not be able to generate convincing proofs for false statements.

While SNARGs for NP are known to exist in the random oracle model [Kil92, Mic94], such proofs cannot exist in the plain model under the widely believed complexity assumption that $\mathsf{NP} \not\subseteq \mathsf{BPTIME}(2^{o(n)})$ [BP04, Wee05].[1] For this reason, we consider SNARGs in the *common reference string* (CRS) model that are defined relative to a (potentially structured) random public value. A long line of work has studied the minimal assumptions needed to construct SNARGs in the CRS model for various sub-classes of NP (see, e.g., [GW11, SW14, KRR14, KPY19, CJJ21, KVZ21, HJKS22, WW22, BBK+23, CGJ+23, JKLV24, WW24, WZ24, WW25, JKLM25]).

The GW barrier. The work of Gentry and Wichs [GW11] provides a fundamental barrier for SNARGs for NP by showing that (under mild hardness assumptions) it is impossible to construct SNARGs for all of NP where adaptive soundness is proven via a black-box reduction from any falsifiable assumption. Following [Nao03], an assumption is *falsifiable* if it can be formulated as an interactive game between an efficient challenger and adversary. In modern cryptography, falsifiable assumptions are greatly preferred since the resulting security reductions give rise to natural "win-win" arguments (breaking a cryptographic system falsifies the underlying assumption, which often resolves a long-standing open mathematical problem). All "standard" cryptographic assumptions such as DDH or LWE are falsifiable. Of course, one could simply assume that a given SNARG for a particular NP language is sound, but this is inherently *not* falsifiable since it requires the challenger to efficiently test that an NP statement is false.

Overcoming the barrier. There are major caveats to the "Gentry-Wichs" (GW) barrier, which offer a path for overcoming it:

1. It only applies to adaptively sound SNARGs, where a cheating prover can choose which false statement to prove adaptively after seeing the CRS.

2. It only applies to either:
   (a) SNARGs from polynomial hardness assumptions (even with a long CRS), or
   (b) SNARGs with a short CRS (even from sub-exponential hardness assumptions).

And indeed, prior works were able to exploit the above caveats to overcome the barrier. Firstly, Sahai and Waters [SW14] showed how to construct *selectively secure* SNARGs for NP from indistinguishability obfuscation (iO) and one-way functions (OWF). Following the breakthrough works of Jain, Lin, and Sahai [JLS21, JLS22] that construct iO from sub-exponentially secure falsifiable assumptions, the Sahai-Waters SNARG can be proven secure via a black-box reduction from a sub-exponentially hard falsifiable assumption.

More recently, a series of works [WW24, WZ24, WW25] showed how to construct *adaptively secure* SNARGs for NP from sub-exponentially secure iO and OWF. These results are not subject

[1] At least with respect to the standard notion of non-uniform security that reduces to the statistical setting.

to the GW barrier since they rely on sub-exponential hardness assumptions *and* have a long CRS that exceeds the statement size.

Uniqueness. Interestingly, there is a downside of the above adaptively secure constructions. The original selectively secure SNARG of Sahai and Waters can achieve a strong *uniqueness* property, where for every true statement x *there exists* a single unique accepting proof π.[2] Aside from being a natural property, uniqueness also has interesting applications (e.g.,) for constructing constant-round concurrent zero-knowledge protocols [CLP13, CLP15]. Unfortunately, in the adaptively secure constructions of [WW24, WZ24, WW25], each statement has at least two accepting proofs. Furthermore, this appears to be critical to the proof of security in those works, which follows a general "two-challenge" approach that seems at odds with uniqueness. In this work, we ask if relaxing uniqueness is necessary:

Can we construct unique *SNARGs for NP with adaptive security from falsifiable assumptions, even based on sub-exponential hardness or with a long CRS?*

1.1 Our Results

We provide both positive and negative results to answer the above question. First, we give a negative result for SNARGs that satisfy perfect completeness.

Theorem 1.1 (Informal; see Theorem 3.1)**.** *Assume sub-exponentially secure one-way functions exist. Then, for any candidate SNARG for NP with perfect completeness and uniqueness, security cannot be proven via a black-box reduction from any falsifiable assumption. This holds even for SNARGs with long CRS or assuming sub-exponential security for the underlying assumption.*

To the best of our knowledge, the above result is the first black-box separation result for a natural class of SNARGs for NP that applies to sub-exponential hardness assumptions and constructions with a long CRS. The result covers SNARGs where the proof size is $(|x|+|w|)^{o(1)} \cdot \mathsf{poly}(\lambda)$ for security parameter λ. If we assume PRGs with stronger exact security, the result can also cover SNARGs with proof size $(|x| + |w|)^{1-\varepsilon} \cdot \mathsf{poly}(\lambda)$ for any constant $\varepsilon > 0$. The result also extends beyond perfect uniqueness to cover SNARGs with even "weak statistical uniqueness" where, for every true statement x there is at least some inverse polynomial probability over an honestly generated CRS that x has a unique valid proof. It also extends beyond perfect completeness to cover SNARGs with "super-strong statistical completeness" where for every true statement x and witness w the probability over an honestly generated CRS that the honestly generated proof π fails to verify is at most $2^{-|\pi|}$.

We complement the negative result above with a positive result by relaxing perfect completeness. At a very high level, we show how to implement the "two-challenge" approach of [WW24] with just *one challenge*, though at the cost of introducing a completeness error. Concretely, for any statement x, our prover will fail to find a proof with negligible probability $2^{-\lambda}$ over the choice of the CRS. The SNARG has perfect uniqueness, where for every honesty generated CRS and every true statement x there is at most a single accepting proof. Similar to all other prior positive results,

[2] We note that our notion of uniqueness is statistical and requires that there do not exist multiple valid proofs. This is in contrast to the notion of unambiguous SNARGs of [CFJW25], which only require that it should be hard to find multiple valid proofs. Achieving unique selectively secure SNARGs can be done easily by using injective OWFs in the original construction of [SW14], or via a slight modification based on arbitrary OWFs; see Section 1.3.2.

our SNARG has a long CRS that grows with the statement and witness length and our reduction relies on sub-exponential hardness assumptions.

Theorem 1.2 (Informal; see Theorem 4.1)**.** *Assuming sub-exponentially secure iO and OWF, there exist adaptively secure perfectly unique SNARGs for NP. The SNARGs have a long CRS and a negligible completeness error. In particular, this holds assuming sub-exponential security of the falsifiable assumptions shown to imply iO in [JLS21, JLS22].*

To the best of our knowledge, our construction is the first unique SNARG with adaptive security from falsifiable assumptions, even restricted to the subclass P. We also note that unique SNARGs for NP (plus OWFs) are known to imply witness encryption [CPW20], and therefore the reliance on some strong falsifiable assumptions in the above theorem is necessary.

1.2 Related Work

Black-box separations for cryptographic primitives. In this work, we consider separations ruling out constructions with black-box security reductions that may make non-black-box use of the underlying primitives. This is in contrast to ruling out "fully" black-box constructions that assume black-box use of the underlying primitive à la the seminal work of Impagliazzo and Rudich [IR88] (see [RTV04] for a taxonomy on black-box separations).

Along these lines, there have been many works ruling out various cryptographic primitives with black-box reductions from various cryptographic assumptions (see, e.g., [Pas06, PTV11, Pas11, GW11, Pas13, BL13, CLMP13, BDG+13, MW20, Kiy21, CGKS23, ADSS24, BD25, CJ25]). Most relevant to this work, [GW11, Pas13] rule out types of adaptively secure non-interactive arguments from efficiently falsifiable assumptions (under relatively mild additional assumptions). Specifically, [GW11] focuses on succinctness whereas [Pas13] focuses on zero-knowledge. As in our work, these black-box separations follow from the general "meta-reduction" paradigm introduced by [BV98].

Computationally unique SNARGs. In this work, we focus on SNARGs with statistical uniqueness, where there *exists* at most one accepting proof for each statement. This notion can be relaxed to computational uniqueness—also known as *unambiguity*—where it is only computationally hard to find two distinct accepting proofs for any statement. Unambiguous SNARGs have been constructed for various deterministic computations from a variety of assumptions (see, e.g., [CHK+19a, CHK+19b, EFKP20, KPY20, LV20, JKKZ21, BCH+22, CFJW25]) most notably for their applications to constructing hard-on-average instances in PPAD. We emphasize that the adaptively secure SNARGs for NP of [WW24, WZ24, WW25] based on [SW14] lose perfect uniqueness though are still computationally unique.

1.3 Technical Overview

We consider SNARGs for NP in the CRS model. For both our positive and negative results, we allow SNARGs with a long CRS that can depend polynomially on the statement and witness size. A SNARG for an NP language L consists of three algorithms $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Ver})$. The setup algorithm outputs the common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$ indexed by the security parameter λ and input-length m. On input a statement x with a valid witness w of combined length at most m, the SNARG prover computes the proof $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$. The verifier accepts if $\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1$. Completeness ensures that the verifier accepts honestly generated proofs with

overwhelming probability, and perfect completeness ensures that this holds with probability 1. Adaptive soundness ensures that for any polynomial-time adversary who is given crs and adaptively chooses a false statement x along with a proof π, the probability that $\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1$ is negligible.

1.3.1 A Black-Box Separation for Unique SNARGs with Perfect Completeness

Overview of GW. We first recall the GW [GW11] black-box separation result for general SNARGs. Following the "meta-reduction" paradigm, the main component of that work is a simulatable attacker against any SNARG. This is a carefully crafted inefficient attacker $P^\star$ that breaks the adaptive security of the SNARG, so on input a CRS it generates a false statement x and a valid proof π for x. At the same time, there is an efficient simulator $P^\star_{\mathsf{sim}}$ such that no efficient algorithm can distinguish the output of $P^\star$ from $P^\star_{\mathsf{sim}}$. The simulator $P^\star_{\mathsf{sim}}$ outputs true statements x and proofs π, and therefore does not break the security of the SNARG. Constructing a simulatable attacker is the main technical challenge in GW and relies on a fundamental *leakage lemma* with deep connections across cryptography and complexity theory.

A simulatable attacker rules out black-box reductions from falsifiable assumptions. Intuitively, this is because simulatable attackers are *useless* from the point of view of any reduction: anything a reduction can do using this attacker, it could also have done on its own by running the efficient simulator. In more detail, suppose there is a black-box reduction R such that $R^{P^\star}$ breaks some underlying assumption with non-negligible advantage. Then $R^{P^\star_{\mathsf{sim}}}$ must also necessarily break the assumption by the indistinguishability of $P^\star$ and $P^\star_{\mathsf{sim}}$. However, since $P^\star_{\mathsf{sim}}$ is efficient, $R^{P^\star_{\mathsf{sim}}}$ is an efficient attack on the assumption, meaning that the assumption was false to begin with!

There is an important subtlety in the above argument. The attacker $P^\star$ and the simulator $P^\star_{\mathsf{sim}}$ are only indistinguishable by polynomial time distinguishers, or at best distinguishers of size $2^{o(|x|)}$. However, they are easily distinguishable by distinguishers of size $2^{|x|}$ who can recognize whether the statements x are true or false.[3] Therefore, the above argument only covers polynomial reductions breaking polynomial security assumptions, but it does not cover subexponential reductions that break subexponential security assumptions. And indeed, the prior works of [SW14, WW24, WZ24, WW25] exactly make use of such reductions to prove the security of SNARGs from subexponential assumptions.

Strengthening GW for unique SNARGs. We strengthen the GW black-box separation result for *unique SNARGs with perfect completeness*, showing that we cannot have a black-box reduction even from subexponentially secure falsifiable assumptions. More generally, we consider c-exponential security assumptions for arbitrary constant $c > 0$, which stipulate that attackers running in time $\mathsf{poly}(2^{\lambda^c}) = 2^{O(\lambda^c)}$ have advantage $\mathsf{negl}(2^{\lambda^c}) = 2^{-\omega(\lambda^c)}$.

Our main technical tool for proving the above is to define a new simulatable attacker for these types of SNARGs. As in GW, the simulatable attacker consists of an inefficient attacker $P^\star$ that breaks SNARG security and a polynomial-time simulator $P^\star_{\mathsf{sim}}$ such that the two are indistinguishable. The main new feature of our simulatable attacker $P^\star$ is that it is *semi-efficient*. In particular, it runs in time $2^{O(p(\lambda))}$ for some polynomial $p(\lambda)$, and ensures that distinguishers of size $2^{p(\lambda)}$ cannot distinguish it from $P^\star_{\mathsf{sim}}$ with advantage better than $2^{-p(\lambda)}$. In other words, the attacker's run-time is *polynomial* in the size (and inverse-advantage) of the distinguishers it protects against. This is contrast to GW where $P^\star$ had size $2^{|\mathsf{crs}|}$, but was easy to distinguish

[3] For simplicity we focus on run-time, but the above also applies to the inverse advantage of the reduction and the assumption.

from the simulator by distinguishers of size $2^{|x|}$; when $|\mathsf{crs}| = \omega(|x|)$ then the attacker's size is super-polynomially larger than the size of the distinguishers it protects against.[4]

The existence of a semi-efficient simulatable attacker rules out black-box reductions even from subexponential falsifiable assumptions. Intuitively, black-box access to a semi-efficient simulatable attacker is useless for reductions that run in less than $2^{p(\lambda)}$ time, since they cannot distinguish it from an efficient simulator, but it is also useless for reductions that run in more than $2^{p(\lambda)}$ time since they can simply run the attack themselves. In more detail, suppose we have some reduction R based on the c-exponential security of a falsifiable assumption. This means that, for some $T(\lambda) = 2^{O(\lambda^c)}$ the reduction R runs in time $T(\lambda)$ and $R^{P^\star}$ has advantage $1/T(\lambda)$ in breaking the assumption (for infinitely many λ). We can consider two cases. If $2T^2(\lambda) \leq 2^{p(\lambda)}$ then we can replace $P^\star$ by the polynomial-time $P^\star_{\mathsf{sim}}$ and rely on indistinguishability to argue that $R^{P^\star_{\mathsf{sim}}}$, which runs in time $T(\lambda) = 2^{O(\lambda^c)}$ and makes at most $T(\lambda)$ calls to the oracle, has advantage at least $1/T(\lambda) - T(\lambda)/2^{p(\lambda)} \geq 1/(2T(\lambda)) = 2^{-O(\lambda^c)}$ in breaking the assumption (for infinitely many λ), meaning that the assumption is not c-exponentially secure to begin with. This part mirrors GW. On the other hand if $2T^2(\lambda) > 2^{p(\lambda)}$ then we can rely on the semi-efficiency of $P^\star$ to argue that the total run time of $R^{P^\star}$ is at most $T(\lambda) \cdot 2^{O(p(\lambda))} = \mathsf{poly}(T(\lambda)) = 2^{O(\lambda^c)}$ and therefore this directly gives an attack on the c-exponential security of the assumption. In either case, the existence of a black-box reduction implies that the assumption is false to being with!

To summarize, black-box access to such a semi-efficient simulatable attacker is "useless" for reductions that run in *too little* time to distinguish it from an efficient simulator, and also for reductions that run in *too much* time so as to able to run the attacker themselves. For the GW simulatable attacker there was a large gap between these two extremes, and in that gap the simulatable attacker was useful and allowed a meaningful reduction to go through. In the special case of unique SNARGs with perfect completeness, we show how to close the gap ensuring that our semi-efficient simulatable attacker is "useless" for all reductions no matter their running time.

A semi-efficient simulatable attack on unique SNARGs. We now describe our semi-efficient simulatable attacker against unique SNARGs with perfect completeness. The simulatable attacker of GW relied on the leakage lemma, which made it inefficient. It did so to carefully sample the correct distribution of proofs for false statements that would be indistinguishable from honestly generated proofs for true statements. In the case of unique SNARGs with perfect completeness, we show how to avoid the leakage lemma entirely and give a much simpler simulatable attacker.

Let $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Ver})$ be any candidate SNARG for NP that has perfect completeness and uniqueness. As in GW, we consider a language L with a subexponentially hard subset membership problem. For concreteness, let $G\colon \{0,1\}^n \to \{0,1\}^{2n}$ be a subexponentially secure pseudorandom generator (PRG) and consider the language of PRG images: $L = \{x : \exists w, G(w) = x\}$. For a sufficiently large seed length $n = \mathsf{poly}(\lambda)$, there is some polynomial p such that for witnesses $w \in \{0,1\}^n$ and statements $x = G(w)$ the SNARG proofs have length $|\pi| \leq p(\lambda)$ and the underlying PRG has $2^{p(\lambda)}$-security, meaning that distinguishers with run-time $\mathsf{poly}(2^{p(\lambda)})$ have advantage $\mathsf{negl}(2^{p(\lambda)})$.

As an initial incomplete attempt, our semi-efficient attacker $P^\star$ is easy to describe; it samples a random false statement and does a brute-force search to find a unique accepting proof. Specifically,

[4]For those familiar with GW, the attacker $P^\star$ there relies on the leakage lemma to sample some distribution, and semi-efficient variants of the leakage lemma exist. However, it needs to sample a different distribution for each choice of the crs and the mapping from crs to the appropriate distribution is not efficient; the only known strategy is to simply write down the appropriate distribution for each possible choice of the crs, making the total size of the attacker exponential in the size of crs. The run-time of $P^\star$ was irrelevant for GW.

on input crs, it does the following:

- $P^\star(\mathsf{crs})$ samples $x^\star \leftarrow \{0,1\}^{2n}$ and checks if there is a unique proof $\pi^\star$ such that $\mathsf{Ver}(\mathsf{crs}, x^\star, \pi^\star) = 1$. If so it outputs $(x^\star, \pi^\star)$ and else it outputs $\perp$.

Note that $x^\star$ is unlikely to be in the image of G, so $x^\star$ will be false with overwhelming probability. It is easy to see that $P^\star$ runs in time $2^{O(p(\lambda))}$. It remains to show that (1) $P^\star$ breaks soundness and (2) $P^\star$ can be efficiently simulated.

For (1), we crucially leverage the fact that $P^\star$ runs in time $\mathsf{poly}(2^{p(\lambda)})$, which is notably *less time than needed to break the PRG*. Therefore, if a random true statement $x = G(w)$ has a unique accepting proof, then a random false statement $x^\star \leftarrow \{0,1\}^{2n}$ will have a unique accepting proof too with overwhelming probability—otherwise the PRG would not be $2^{p(\lambda)}$ secure.

For (2), we consider a simple simulation strategy that chooses a random true statement and gives an honestly generated proof:

- $P^\star_{\mathsf{sim}}(\mathsf{crs})$ samples $w \leftarrow \{0,1\}^n$, $x := G(w)$ and outputs (x, π) where $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$.

Note that $P^\star_{\mathsf{sim}}$ runs in time $\mathsf{poly}(\lambda, n)$ since the SNARG prover is efficient. For any crs where perfect completeness and uniqueness hold we have:

$$\{(x^\star, \pi^\star) \leftarrow P^\star(\mathsf{crs})\} \approx \{(x, \pi) \leftarrow P^\star_{\mathsf{sim}}(\mathsf{crs})\},$$

where distinguishers running in time $2^{p(\lambda)}$ will have distinguishing advantage $2^{-p(\lambda)}$. The proof follows in two steps: first we replace a random $x^\star \leftarrow \{0,1\}^{2n}$ by a random $x = G(w)$ for $w \leftarrow \{0,1\}^n$ while still generating $\pi^\star$ using brute force. Indistinguishability follows from PRG security. We then replace the brute force $\pi^\star$ by an honesty generated proof, and by perfect completeness/uniqueness we ensure that the two are the same.

Unfortunately, there is a issue with the above strategy, and it does not quite work as is. Perfect completeness/uniqueness are only defined with respect to honestly generated CRS, but the CRS that is given to $P^\star$ can be chosen arbitrarily by the reduction (potentially from outside of the support of honestly generated ones) and there is no guarantee that completeness/uniqueness should hold for it. To handle this, we can have $P^\star$ first do a *completeness/uniqueness check* on the given crs: it samples $2^{p(\lambda)}$ random true statements and checks that the honestly generated proof for them is the unique proof that verifies. If so it proceeds as before, and if not it simply acts as $P^\star_{\mathsf{sim}}$ above. This then ensures that, if the check passes, then for an honestly generated true statement x we can replace the brute-force proof $\pi^\star$ with an honestly generated proof π and the two are likely to be the same with all but $2^{-p(\lambda)}$ probability. Our full construction is somewhat more complex to also handle the case of weak uniqueness, and we defer to Section 3 for the full details.

Reductions that do not preserve the security parameter. In our discussion of both GW and our result so far, we only considered restricted reductions R that, when given security parameter λ, only call their oracle on the same value λ. We call such reduction "security parameter preserving". For security parameter preserving reductions, GW didn't need any restriction on the CRS size. Therefore, for such reductions, the main difference is that GW's result only applied to polynomial security assumptions, while our result applies even to subexponential security assumptions.

But general reductions may not be security parameter preserving and may call their oracle on different values of $\lambda' \neq \lambda$. Even if we have a simulatable attacker $(P^\star, P^\star_{\mathsf{sim}})$, a reduction running in time $\mathsf{poly}(\lambda)$ that calls its oracle on a sufficiently small values of $\lambda' \ll \lambda$ will be able to distinguish

between $P^\star$ and $P^\star_{\mathsf{sim}}$. Therefore, for general reductions, we need to be more careful when arguing that simulatable attackers are useless. To handle this case, GW crucially imposed an additional requirement that the CRS also needs to be as short as the proof size. They showed that in this case there is some sharp threshold such that, if λ' is below that threshold then the output of $P^\star$ can be generated efficiently and if λ' is above the threshold then the output is indistinguishable from $P^\star_{\mathsf{sim}}$. In either case, the output of $P^\star$ can be efficiently simulated and so oracle access to $P^\star$ is useless.

Using our semi-efficient simulatable attacker, we can instantiate a similar strategy without requiring a short CRS.[5] In particular, suppose there is a general reduction R based on the c-subexponential security of a falsifiable assumption. This means that, for some $T(\lambda) = 2^{O(\lambda^c)}$ the reduction R runs in time $T(\lambda)$ and $R^{P^\star}(1^\lambda)$ has advantage $1/T(\lambda)$ in breaking the assumption (for infinitely many λ). The reduction may call its oracle on different values of $\lambda' \neq \lambda$. Consider an algorithm $R^\star$ (without oracles) that runs R and for each oracle call that R makes with security parameter λ': if $2T^2(\lambda) \leq 2^{p(\lambda')}$ then $R^\star$ responds using $P^\star_{\mathsf{sim}}$ and else it responds by running $P^\star$ itself. The run-time of $R^\star$ is $\mathsf{poly}(T(\lambda)) = 2^{O(\lambda^c)}$ and its advantage in breaking the assumption is at least $T(\lambda)/2 \geq 2^{-O(\lambda^c)}$ by essentially the same analysis as before. This means that $R^\star$ gives an attack on the c-subexponential security of the assumption, so the assumption was false to begin with!

Remark: On subexponential hardness assumptions via complexity-leveraging. We note that once we allow for general reductions, any subexponential security assumption can be rewritten as an equivalent polynomial security assumption. This is because, for any constant $c > 0$, assuming that every $2^{O(\lambda^c)}$-size attacker has advantage $2^{-\omega(\lambda^c)}$ in some game with a challenger defined by $C(1^\lambda)$ is the same as assuming that every polynomial attacker has negligible advantage in a "rescaled game" with challenger $\hat{C}(1^\lambda)$ who runs $C(1^{\hat{\lambda}})$ for $\hat{\lambda} = \log^{1/c} \lambda$; this holds since $\mathsf{poly}(\lambda) \equiv 2^{O(\hat{\lambda}^c)}$ and $\mathsf{negl}(\lambda) \equiv 2^{-\omega(\hat{\lambda}^c)}$. However, showing the black-box equivalence of these assumptions relies on a general non-security parameter preserving reduction! Therefore, the black-box separation of GW for SNARGs with a short CRS, which worked for general reductions from polynomial security assumptions, already (implicitly) also covers subexponential security assumptions due to the above equivalence.

Fine-grained exponential security. We note that our negative result rules out black-box reductions from any subexponential falsifiable assumptions, or more concretely, from any c-exponential assumption for any constant $c > 0$. However, it still leaves open the possibility of a black-box reduction from the fine-grained exact security of a falsifiable assumption. For example, it may be possible to get a proof by assuming that some assumption is secure against 2^λ time attackers, even though it is not secure against $2^{2\lambda}$ attackers. This is because we showed that a reduction from a $2^{O(\lambda^c)}$-security assumption implies a $2^{O(\lambda^c)}$ attack, but the constant hidden in the $O(\cdot)$ may be larger in the second term than the first. Exploiting or closing this loophole is a fascinating open problem for future work.

[5] GW essentially relied on the fact that their simulatable attacker is semi-efficient when the CRS is short to get their result, while our simulatable attacker is semi-efficient even if the CRS is long.

1.3.2 Constructing Unique SNARGs with Imperfect Completeness

Overview of the Sahai-Waters SNARG. We first recall the construction of Sahai and Waters [SW14], which is unique but only proven to be selectively secure.[6] Let L be an NP language with corresponding NP relation R. Their construction relies on an iO scheme Obf and a puncturable PRF F. The Setup algorithm generates two obfuscated programs for the CRS: one to generate proofs and the other to verify proofs. The proof generation algorithm $\mathsf{GenProof}$ has a hard-coded random PRF key k, takes as input (x, w), checks that w is a good witness for x, and if so, outputs a "signature" of the form $F(k, x)$, else $\perp$. The proof verification algorithm $\mathsf{VerProof}$ takes (x, π) as input and verifies the "signature" by checking if $\pi = F(k, x)$. The CRS then consists of obfuscated versions of these programs $\mathsf{Obf}(\mathsf{GenProof}[k])$ and $\mathsf{Obf}(\mathsf{VerProof}[k])$, and the SNARG prover and verifier simply run the respective obfuscated programs. We summarize the $\mathsf{GenProof}$ and $\mathsf{VerProof}$ programs for the Sahai-Waters construction below.

$\mathsf{GenProof}[k](x, w)$:

- If $(x, w) \notin R$, output $\perp$.
- Output $F(k, x)$.

$\mathsf{VerProof}[k](x, \pi)$:

- Output 1 iff $\pi = F(k, x)$.

It is clear by construction that this results in perfectly unique SNARGs; every (true or false) statement has exactly one proof $\pi = F(k, x)$ that verifies. Furthermore, it is a SNARG where the CRS depends on (and grows polynomially with) a specific input-length for $(x, w) \in R$.

In terms of security, this construction is only *selectively* rather than adaptively secure. In the selective security game, the adversarial prover $P^\star$ must declare the false statement x that it will find an accepting proof for before seeing the CRS. Security is then shown via the following sequence of hybrid experiments:

- We first switch the PRF key k in the CRS by a "punctured" key $k^{(x)}$ and set $F(k, x)$ to be a random value $x^\star$ in $\mathsf{VerProof}$. Indistinguishability follows from iO and the punctured security of F and relies on the fact that $\mathsf{GenProof}$ never queries $F(k, x)$ for false statements $x \notin L$.[7]
- We next replace the check in $\mathsf{VerProof}$ that $\pi = x^\star$ with the check that $f(\pi) = f(x^\star)$ for any injective OWF f. Indistinguishability follows from iO security since f is injective.[8]

Security then follows via a simple reduction to the one-wayness of f. Specifically, the reduction can embed the one-way function challenge $y^\star = f(x^\star)$ into the CRS so any accepting proof π produced by $P^\star$ for x is a valid pre-image of $y^\star$ by construction.

Our construction. Our construction is a very simple modification of the Sahai-Waters SNARG above. The only difference is that we use a "selector" PRF with key k_{sel} to determine whether or not the $\mathsf{GenProof}$ program gives out an honest proof for each statement x.

[6] Technically, we describe a slight simplification of their original protocol, which can be proven secure based on the new techniques of [WW25].

[7] Recall that punctured PRF security guarantees that the punctured key preserves functionality for all $x' \neq x$ so $F(k, x') = F(k^{(x)}, x')$, but $F(k, x)$ remains pseudorandom even given $k^{(x)}$.

[8] We recall that [BPW16] show that an injective OWF f can be constructed from iO and any plain OWF. Waters and Wu [WW25] show that it suffices to use an injective OWF f with an inefficient sampler, which can be built from OWF alone.

In more detail, for valid statements x, the $\mathsf{GenProof}$ algorithm computes a λ-bit selector PRF output $F(k_{\mathsf{sel}}, x) \in \{0,1\}^\lambda$. If the output is equal to a random string $z \leftarrow \{0,1\}^\lambda$, the $\mathsf{GenProof}$ algorithm aborts and outputs $\perp$. Otherwise, it gives out a proof $F(k, x)$ just as in the Sahai-Waters construction. The $\mathsf{VerProof}$ program is identical to the Sahai-Waters SNARG in that it just outputs 1 iff $\pi = F(k, x)$. The CRS then consists of the obfuscated programs $\mathsf{Obf}(\mathsf{GenProof}[k, k_{\mathsf{sel}}, z])$ and $\mathsf{Obf}(\mathsf{VerProof}[k])$, and the prover and verifier simply run their associated programs. We summarize the $\mathsf{GenProof}$ and $\mathsf{VerProof}$ programs for our construction below.

$\mathsf{GenProof}[k, k_{\mathsf{sel}}, z](x, w)$:

- If $(x, w) \notin R$, output $\perp$.
- If $F(k_{\mathsf{sel}}, x) = z$, output $\perp$.
- Output $F(k, x)$.

$\mathsf{VerProof}[k](x, \pi)$:

- Output 1 iff $\pi = F(k, x)$.

It's clear by inspection that our SNARG construction is unique as the verification program is identical to Sahai-Waters. However, we have lost perfect completeness since there is a $2^{-\lambda}$ chance that $F(k_{\mathsf{sel}}, x) = z$ for a random $z \leftarrow \{0,1\}^\lambda$, at which point the honest prover cannot find $F(k, x)$.

We next argue adaptive security using a very similar argument as in the work of Waters and Wu [WW25]. Suppose there is an adversarial prover $P^\star$ that, on input a honestly sampled CRS, breaks adaptive soundness and outputs a false statement x and accepting proof π with noticeable probability. We show how to embed an injective one-way function challenge $y^\star = f(x^\star)$ into the CRS as follows:

- First, we argue that the success probability of $P^\star$ is essentially independent of the value $F(k_{\mathsf{sel}}, x)$. As such, the probability that $P^\star$ succeeds on statements x such that $F(k_{\mathsf{sel}}, x) = z$ is
$$\Pr[P^\star \text{ wins} \wedge F(k_{\mathsf{sel}}, x) = z] \approx \frac{1}{2^\lambda} \cdot \Pr[P^\star \text{ wins}].$$
At a high level, this holds for each false statement $x \notin L$ since the value $F(k_{\mathsf{sel}}, x)$ is indistinguishable from random as $\mathsf{GenProof}$ never queries $F(k_{\mathsf{sel}}, x)$ on false x. To argue this holds for all false statements x, we use $O(2^{|x|})$-many hybrids relying on subexponential iO and puncturable PRF security.

- Next, we embed a single injective OWF challenge $y^\star = f(x^\star)$ into the $\mathsf{VerProof}$ program in the "off-path" case, resulting in the following program:

  $\mathsf{VerProof}^\star[k, k_{\mathsf{sel}}, z](x, \pi)$:

  - If $F(k_{\mathsf{sel}}, x) = z$, output 1 iff $y^\star = f(\pi \oplus F(k, x))$.
  - Otherwise, output 1 iff $\pi = F(k, x)$.

  At a high level, this change is indistinguishable to $P^\star$ since $\mathsf{GenProof}$ reveals nothing about $F(k, x)$ for any statement x where $F(k_{\mathsf{sel}}, x) = z$. Following [WW25], we rely on the fact that for any $x^\star$
$$\pi = F(k, x) \text{ iff } x^\star = x^\star \oplus F(k, x) \oplus \pi.$$

At the same time, it holds that

$$x^\star \oplus F(k,x) \approx F(k,x)$$

since both sides are indistinguishable from random. Then by injectivity of f, it follows that

$$x^\star = F(k,x) \oplus \pi \text{ iff } y^\star = f(F(k,x) \oplus \pi).$$

Notably, indistinguishably switching to the program $\mathsf{VerProof}^\star$ only relies on $O(2^{|x|})$-many hybrids based on iO and puncturable PRF security, and never invokes the injective OWF security.

Finally, security follows via a reduction to the one-wayness of f. Namely, if $P^\star$ wins and outputs (x,π) such that $F(k_{\mathsf{sel}},x) = z$ and $f(\pi \oplus F(k,x)) = y^\star$, then $x^\star = \pi \oplus F(k,x)$ is a valid pre-image of $y^\star$. Compared to [WW25], our reduction succeeds with roughly $2^{-\lambda}$ times the probability that $P^\star$ wins. As a result, we need to rely on subexponential security of the injective OWF f. Importantly for this step, we only have to increase the domain and range of f to be polynomial in λ instead of $|x|$, ensuring our resulting SNARG maintains succinctness.

2 Preliminaries

We let $\mathbb{N} = \{1,2,3,\ldots\}$ denote the set of natural numbers, and for any $n \in \mathbb{N}$, we write $[n]$ to denote the set $[n] = \{1,\ldots,n\}$. For integers $a,b \in \mathbb{Z}$ with $a \le b$, we write $[a,b]$ to denote the set $\{a,a+1,\ldots,b\}$. For a string $s \in \{0,1\}^*$, we let $|s|$ denote the length of s.

We say that a function $p\colon \mathbb{N} \to \mathbb{N}$ is in the set $\mathsf{poly}(n)$ and is *polynomially-bounded* if there exists a constant c and an index $i \in \mathbb{N}$ such that $p(n) \le n^c$ for all $n \ge i$. We say that a function $\epsilon\colon \mathbb{N} \to \mathbb{R}$ is in the set $\mathsf{negl}(n)$ and is *negligible* if for every constant $c > 0$ there exists $i \in \mathbb{N}$ such that $\epsilon(n) \le n^{-c}$ for all $n \ge i$.

A uniform algorithm A is one with a constant description size, and a non-uniform algorithm A consists of a sequence of algorithms $\{A_\lambda\}_{\lambda\in\mathbb{N}}$. We assume for simplicity that A_λ always receives 1^λ as its first input and may write $A(\cdot)$ instead of $A_\lambda(\cdot)$ when the index λ is clear from context. We use PPT to denote the acronym *probabilistic, polynomial time*. A non-uniform PPT algorithm is one where the description size of A_λ as a function of λ is in $\mathsf{poly}(\lambda)$. When referring to an algorithm implemented as a circuit C, we let $|C|$ denote the size of the circuit. For two circuits C, C', we write $C \equiv C'$ to denote that they are functionally equivalent, so for all inputs x, it holds that $C(x) = C'(x)$.

For a distribution D, we write $x \leftarrow D$ to denote the process of sampling a value x from the distribution D. For a set S, we use $x \leftarrow S$ to denote the process of sampling a value x from the uniform distribution over S. We use $x = A(\cdot)$ to denote the output of a deterministic algorithm and $x \leftarrow A(\cdot)$ to denote the output of a randomized algorithm. We write $x := y$ to denote the assignment of value y to x. For a distribution D, we define $\mathrm{Supp}\,(D)$ to denote the support of the distribution D. We write U_n to denote the uniform distribution over $\{0,1\}^n$.

We say that a language L over $\{0,1\}^*$ is in the class NP if there exists a polynomial-time verifiable witness relation R such that $x \in L$ if and only if there exists a witness $w \in \{0,1\}^{\mathsf{poly}(|x|)}$ such that $(x,w) \in R$. For simplicity in this work, we identify an NP language L by its corresponding witness relation R, although we note that such a language L may have many valid witness relations.

2.1 Succinct Non-interactive Arguments

Succinct non-interactive arguments (SNARGs) for an NP witness relation R are non-interactive and computationally sound proofs in the CRS model, where the proofs are significantly shorter than the corresponding NP witnesses. Following the notion from [SW14, WW24], we also consider SNARGs where the CRS may depend on a specific input-length m for the relation R. We first define the more standard notion perfect completeness and adaptive soundness, and discuss imperfect completeness and uniqueness below.

Definition 2.1 (SNARGs for NP)**.** Let L be an NP language with witness relation R. We say that a tuple $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ is a *succinct non-interactive argument (SNARG) for* R if it has the following syntax and efficiency:

- $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^m)$: A randomized polynomial-time algorithm that on input the security parameter 1^λ and input length 1^m in unary outputs a common reference string crs. We assume for simplicity that crs contains 1^λ.
- $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$: A deterministic polynomial-time algorithm that on input the common reference string crs, an input x, and a witness w, outputs a proof π.
- $b = \mathsf{Ver}(\mathsf{crs}, x, \pi)$: A deterministic polynomial-time algorithm that on input the common reference string crs, an input x, and a proof π, outputs a bit b indicating whether to accept or reject.

We require that $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ satisfy the following properties:

- **(Perfect) Completeness:** For all $\lambda, m \in \mathbb{N}$ and $(x, w) \in R \cap \{0,1\}^m$, it holds that
$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^m) \\ \pi = \mathsf{Prove}(\mathsf{crs}, x, w)\end{array} : \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1\right] = 1.$$
- **(Adaptive) Soundness:** For all non-uniform PPT attackers A and polynomials m, there exists a negligible function ϵ such that for all $\lambda \in \mathbb{N}$, it holds that
$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)}) \\ (x, \pi) \leftarrow A(\mathsf{crs})\end{array} : \begin{array}{l}\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \\ \wedge\ x \notin L\end{array}\right] \leq \epsilon(\lambda).$$
- **Succinct Proofs:** There exists a function $p(\lambda, m) \in \mathsf{poly}(\lambda) \cdot m^{o(1)}$ such that for all $\lambda, m \in \mathbb{N}$, $(x, w) \in R \cap \{0,1\}^m$, $\mathsf{crs} \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda, 1^m)\right)$, $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$, it holds that
$$|\pi| \leq p(\lambda, m).$$

In this work, we consider relaxing the notion of completeness to allow for some error, formalized below.

Definition 2.2 (Completeness Error)**.** Let $\delta \colon \mathbb{N} \to [0,1]$. We say that a SNARG $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ for a language L with witness relation R has δ-completeness error if for all $\lambda, m \in \mathbb{N}$ and $(x, w) \in R \cap \{0,1\}^m$, it holds that
$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^m) \\ \pi = \mathsf{Prove}(\mathsf{crs}, x, w)\end{array} : \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1\right] \geq 1 - \delta(\lambda).$$

We also consider a notion of (statistical) uniqueness, which says that any statement x has at most one accepting proof. We formalize this below.

Definition 2.3 (Uniqueness)**.** Let $\alpha\colon \mathbb{N} \to [0,1]$. We say that a SNARG $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ for a language L with witness relation R has α-uniqueness if for all $\lambda, m \in \mathbb{N}$ and $(x,w) \in R \cap \{0,1\}^m$, it holds that

$$\Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^m) \\ : \exists \pi \neq \pi' \text{ s.t.} \end{array} \begin{array}{l} \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge \\ \mathsf{Ver}(\mathsf{crs}, x, \pi') = 1 \end{array}\right] \leq 1 - \alpha(\lambda).$$

2.2 Falsifiable Cryptographic Assumptions

Falsifiable cryptographic assumptions are modeled via security games between interactive algorithm. To model communication between two interactive algorithms, we assume each algorithm has a specified part of its memory for messages to and from another algorithm. When one algorithm temporarily halts with a message to another algorithm, we assume that this message is copied as an input for the other algorithm. Interaction proceeds until one algorithm halts communication and produces an overall output for the interaction.

Let C and A be two interactive algorithms representing the "challenger" and "attacker", respectively. We denote by $b \leftarrow \langle C, A\rangle(1^\lambda)$ the random variable representing the output $b \in \{0,1\}$ of C when interacting with A on common input 1^λ, when the random tape of each algorithm is uniformly and independently chosen. When $b = 1$, we say that the attacker "wins" in the interaction.

In contrast to [GW11], we define subexponential time falsifiable cryptographic assumptions by default.

Definition 2.4 (Subexponential Time Falsifiable Cryptographic Assumptions)**.** Let $c > 0$. Let C be an interactive probabilistic $2^{O(\lambda^c)}$-time algorithm and $\alpha \geq 0$ a constant. We say that the pair (C, α) is a *c-subexponential time falsifiable cryptographic assumption* if for every non-uniform $2^{O(\lambda^c)}$-time interactive algorithm A, there exists a negligible function ϵ such that for all $\lambda \in \mathbb{N}$, it holds that

$$\Pr\left[b \leftarrow \langle C, A\rangle(1^\lambda) : b = 1\right] \leq \alpha + \epsilon(2^{\lambda^c}).$$

2.3 Black-Box Reductions

We define a notion of black-box reductions in line with [GW11]. For concreteness, we focus specifically on reductions showing the soundness of some SNARG based on a subexponential time falsifiable cryptographic assumption.

Definition 2.5 (SNARG Adversaries)**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a SNARG. We say that an (possibly inefficient) algorithm A is a Π-adversary if there exist polynomials m, q such that for infinitely many $\lambda \in \mathbb{N}$, it holds that

$$\Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)}) \\ (x, \pi) \leftarrow A(\mathsf{crs}) \end{array} : \begin{array}{l} \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \\ \wedge\ x \notin L \end{array}\right] > 1/q(\lambda).$$

Definition 2.6 (Black-Box Reductions)**.** Let Π be a SNARG and (C, α) be a c-subexponential time falsifiable cryptographic assumption for $c > 0$. A probabilistic $2^{O(\lambda^c)}$-time oracle machine $R^{(\cdot)}$ is an *c-subexponential time black-box reduction from the soundness of Π to (C, α)* if for every Π-adversary A, there exists a polynomial q' such that for infinitely many $\lambda \in \mathbb{N}$, it holds that

$$\Pr\left[b \leftarrow \langle C, R^A\rangle(1^\lambda) : b = 1\right] > \alpha + 1/q'(2^{\lambda^c}).$$

2.4 Efficiently Samplable Hard Subset Membership Problems

We define a new notion of hard subset membership problems, slightly modifying the corresponding notion from [GW11]. The definition from [GW11] stipulates that there are two indistinguishable sequences of distributions $\mathcal{L} = \{\mathcal{L}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\bar{\mathcal{L}} = \{\bar{\mathcal{L}}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathcal{L}$ is over true statements x in a language L and $\bar{\mathcal{L}}$ is over false statements $\bar{x}$ from $\bar{L} = \{0,1\}^* \setminus L$. They require that $\mathcal{L}_\lambda$ is samplable by an efficient algorithm $\mathsf{Sam}(1^\lambda)$ that additionally outputs valid witnesses w.

In our new notion formalized below, we syntactically replace $\bar{\mathcal{L}}$ with a different sequence of distributions $\mathcal{L}^\star$, which differs from $\bar{\mathcal{L}}$ in two key ways: (1) we require that $\mathcal{L}^\star$ is efficiently samplable, yet (2) we only require $\mathcal{L}^\star$ to output false statements up to some negligible error probability. We note that all candidate instantiations from [GW11] satisfy our new notion with slight modifications. For example, for a subexponentially secure PRG G, we can define $\mathcal{L}_\lambda$ to be the distribution over random PRG images $G(U_\lambda)$ and $\mathcal{L}^\star_\lambda$ to be the uniform distribution over the co-domain of G on inputs of length λ. It is clear that $\mathcal{L}^\star_\lambda$ is efficiently samplable and yields enough false statements for any sufficiently length-expanding function G.

Definition 2.7 (Efficiently Samplable Hard Subset Membership Problems)**.** Let L be an NP language with relation R and $c > 0$.We say that a tuple $(\mathcal{L}, \mathcal{L}^\star, \mathsf{Sam})$ is an *efficiently samplable c-subexponentially hard subset membership problem for* L if the following hold:

- **Efficient Sampling:** $\mathcal{L} = \{\mathcal{L}_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathcal{L}^\star = \{\mathcal{L}^\star_\lambda\}_{\lambda\in\mathbb{N}}$, and $\{\mathsf{Sam}(1^\lambda)\}_{\lambda\in\mathbb{N}}$ are efficiently samplable sequences of distributions.
- **Correctness of $\mathcal{L}$ and Sam:** For all $\lambda \in \mathbb{N}$, it holds that

$$\{x \leftarrow \mathcal{L}_\lambda\} \equiv \{x : (x,w) \leftarrow \mathsf{Sam}(1^\lambda)\},$$

  and for any $(x,w) \in \mathrm{Supp}\left(\mathsf{Sam}(1^\lambda)\right)$, it holds that $(x,w) \in R$.
- **Correctness of $\mathcal{L}^\star$:** There exists a negligible function ϵ such that for all $\lambda \in \mathbb{N}$, it holds that

$$\Pr[x \leftarrow \mathcal{L}^\star_\lambda : x \notin L] > 1 - \epsilon(\lambda).$$

- **Hardness:** For any non-uniform probabilistic $2^{O(\lambda^c)}$-size algorithm D, there exists a negligible function ϵ such that for all $\lambda \in \mathbb{N}$, it holds that

$$\left|\Pr\left[x \leftarrow \mathcal{L}_\lambda : D(x) = 1\right] - \Pr\left[x^\star \leftarrow \mathcal{L}^\star_\lambda : D(x^\star) = 1\right]\right| \leq \epsilon(2^{\lambda^c}).$$

We say that $(\mathcal{L}, \mathcal{L}^\star, \mathsf{Sam})$ has input-length $m(\lambda)$ if it holds for any $(x,w) \in \mathrm{Supp}\left((\mathsf{Sam}(1^\lambda)\right)$ that $(|x| + |w|) \leq m(\lambda)$.

3 Black-Box Separation for Unique SNARGs

In this section, we prove our lower bound separating unique SNARGs for NP from any falsifiable assumption, assuming the existence of an efficiently samplable subexponentially hard subset membership problem as per Definition 2.7.

Theorem 3.1 (Black-Box Separation)**.** *Let L be an NP language with relation R and an efficiently samplable subset membership problem $(\mathcal{L}, \mathcal{L}^\star, \mathsf{Sam})$ with input-length $m(\lambda)$. Let $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a SNARG for R with proof length $p(\lambda)$ on inputs of length $m(\lambda)$. Let $c > 0$ such that $p(\lambda) \in O(\lambda^c)$ and q_{uniq} be any polynomial. Assuming $(\mathcal{L}, \mathcal{L}^\star, \mathsf{Sam})$ is c-subexponentially hard and Π has $2^{-p(\lambda)}$-completeness error and $1/q_{\mathsf{uniq}}(\lambda)$-uniqueness, then for any d-subexponential time falsifiable cryptographic assumption (C, α) for constant $d > 0$, one of the following must hold:*

- *The assumption (C, α) is false.*
- *There is no d-subexponential time black-box reduction from the soundness of Π to (C, α).*

To prove the theorem, we first construct a basic adversary $P^\star$ in Lemma 3.2 below against the soundness of Π that is a *semi-efficient simulatable* attacker, meaning that it can be efficiently simulated for essentially any algorithm that can't already compute $P^\star$. We then finish the proof of the theorem by showing that this immediately gives a way to efficiently simulate any reduction to a falsifiable assumption that only makes black-box use of $P^\star$.

Lemma 3.2 (Simulatable Attacker)**.** *For the same values of p and c from Theorem 3.1, there exists an algorithm $P^\star$, called a semi-efficient simulatable Π-adversary, such that the following hold:*

- ***Efficiency:*** *For any crs corresponding to security parameter 1^λ, $P^\star(\mathsf{crs})$ runs in time $2^{2 \cdot p(\lambda)} \cdot \mathsf{poly}(\lambda)$.*
- ***Correctness:*** *$P^\star$ is a valid Π-adversary for adaptive soundness.*
- ***Simulation:*** *There exists an efficient simulator $P^\star_{\mathsf{sim}}$ such that for any non-uniform $2^{O(\lambda^c)}$-size distinguisher D and input crs corresponding to sufficiently large security parameter $\lambda \in \mathbb{N}$, it holds that*
$$|\Pr[D(P^\star(\mathsf{crs})) = 1] - \Pr[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1]| \leq 2^{-p(\lambda)}.$$

Proof. We define the algorithm $P^\star$ and then prove the necessary properties for the lemma. Throughout the proof, we assume that $P^\star$ receives as input some (not necessarily honestly generated) CRS crs corresponding to security parameter 1^λ and input-length $m(\lambda) \in \mathsf{poly}(\lambda)$.

At a high level, the attacker $P^\star$ samples a random statement $x^\star \leftarrow \mathcal{L}^\star_\lambda$ and outputs a unique verifying proof $\pi^\star$, if one exists. Our goal is to show that this can be simulated by sampling a random true statement $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$ and outputting the honestly generated proof π for x. It turns out this is true if perfect completeness and uniqueness holds for the input crs.

To handle (potentially malformed) crs where perfect completeness and uniqueness may not hold, we first run a statistical test to check for completeness, outputting an honest (x, π) if the test fails. Otherwise, we attempt to output $(x^\star, \pi^\star)$ as above. If $x^\star$ doesn't have a unique accepting proof, we instead output an honest (x, π) *conditioned on not having a unique accepting proof.*

To formalize the above, we first introduce helper functions Comp and Uniq that test for completeness and uniqueness, respectively, for a given crs and input statements. Then we give a formal description of $P^\star$ on input crs corresponding to security parameter λ.

$\mathsf{Comp}(\mathsf{crs}, x, w)$: Output 1 iff $\mathsf{Ver}(\mathsf{crs}, x, \mathsf{Prove}(\mathsf{crs}, x, w)) = 1$.

$\mathsf{Uniq}(\mathsf{crs}, x)$: Output 1 iff there exists a unique $\pi \in \{0,1\}^{p(\lambda)}$ such that $\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1$.

$P^\star(\mathsf{crs})$:

1. Set $b_{\mathsf{crs}} = 1$ if for $2^{p(\lambda)}$ samples $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$, it holds that $\mathsf{Comp}(\mathsf{crs}, x, w) = 1$.
2. If $b_{\mathsf{crs}} = 1$:
   (a) Sample $x^\star \leftarrow \mathcal{L}^\star_\lambda$. If $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$, output $(x^\star, \pi^\star)$ where $\pi^\star$ is the unique proof such that $\mathsf{Ver}(\mathsf{crs}, x^\star, \pi^\star) = 1$.
   (b) Otherwise, sample $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$ at most $2^{p(\lambda)}$ times until $\mathsf{Uniq}(\mathsf{crs}, x) \neq 1$. If such an x is found, output (x, π) where $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$, or output $\perp$ otherwise.
3. Otherwise, if $b_{\mathsf{crs}} \neq 1$:
   (a) Sample $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$ and output (x, π) where $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$.

In the subsequent claims, we prove that the required efficiency, correctness, and simulation properties for $P^\star$ hold.

Claim 3.3 (Efficiency)**.** *For any string* crs *corresponding to security parameter* 1^λ*,* $P^\star(\mathsf{crs})$ *runs in time* $2^{2 \cdot p(\lambda)} \cdot \mathsf{poly}(\lambda)$*.*

Proof. First, note that $\mathsf{Comp}(\mathsf{crs}, x, w)$ runs in time $\mathsf{poly}(\lambda)$ by the efficiency of Prove and Ver, and $\mathsf{Uniq}(\mathsf{crs}, x)$ runs in time $2^{p(\lambda)} \cdot \mathsf{poly}(\lambda)$ by running Ver on every possible string π.

It follows that Step 1 runs in time $2^{p(\lambda)} \cdot \mathsf{poly}(\lambda)$ by efficiency of Sam and Comp, Step 2a runs in time $2^{p(\lambda)} \cdot \mathsf{poly}(\lambda)$ by efficiency of Uniq, Ver, and $\mathcal{L}^\star_\lambda$, Step 2b runs in time $2^{2p(\lambda)} \cdot \mathsf{poly}(\lambda)$ by the efficiency of Sam, Uniq, and Prove, and Step 3a runs in time $\mathsf{poly}(\lambda)$ by the efficiency of Sam and Prove. Thus, the overall running time of $P^\star$ on any input crs is at most $2^{2p(\lambda)} \cdot \mathsf{poly}(\lambda)$. □

Claim 3.4 (Correctness)**.** *There exists a polynomial* q *such that for for sufficiently large* $\lambda \in \mathbb{N}$*, it holds that*

$$\Pr\left[\begin{array}{l|l} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)}) & \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \\ (x, \pi) \leftarrow P^\star(\mathsf{crs}) & \wedge\ x \notin L \end{array}\right] > 1/q(\lambda).$$

Proof. Let $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)})$ be a randomly sampled CRS for security parameter λ. It suffices to show that $P^\star(\mathsf{crs})$ outputs $(x^\star, \pi^\star)$ in Step 2a such that $x^\star \notin L$ with inverse polynomial probability. To do so, we argue that (1) $b_{\mathsf{crs}} = 1$, (2) $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$, and (3) $x^\star \notin L$ all hold with sufficient probability.

To help analyze events (1) and (2), we define quantities α_{crs} and β_{crs} to denote the probability that completeness and uniqueness hold for a single sample $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$:

$$\alpha_{\mathsf{crs}} := \Pr\left[(x, w) \leftarrow \mathsf{Sam}(1^\lambda) : \mathsf{Comp}(\mathsf{crs}, x, w) = 1\right],$$
$$\beta_{\mathsf{crs}} := \Pr\left[(x, w) \leftarrow \mathsf{Sam}(1^\lambda) : \mathsf{Uniq}(\mathsf{crs}, x) = 1\right].$$

We want to argue for honestly generated crs that both α_{crs} and β_{crs} are sufficiently large with noticeable probability. Specifically, we say that a value crs is "good" if it is in the set G defined as follows:

$$G = \left\{ \mathsf{crs} : \begin{array}{l} \alpha_{\mathsf{crs}} \geq 1 - 4q_{\mathsf{uniq}}(\lambda) \cdot 2^{-p(\lambda)} \\ \wedge\ \beta_{\mathsf{crs}} \geq 1/2q_{\mathsf{uniq}}(\lambda) \end{array} \right\}.$$

To make sense of the set G, we assume that $p(\lambda) > 2 \log q_{\mathsf{uniq}}(\lambda)$ so $\alpha_{\mathsf{crs}} > 0$. This is without loss of generality since if the proofs are very small and $p(\lambda) \leq 2 \log q_{\mathsf{uniq}}(\lambda)$, then we could define a new prover that ignores its witness and does brute force search to find the first accepting proof. This would take time $2^{p(\lambda)} \leq q^2_{\mathsf{uniq}}(\lambda) \in \mathsf{poly}(\lambda)$, so the verifier could also check this, ensuring uniqueness always holds while maintaining at least the same level of completeness.

Now, to analyze the size of G, we first note from our assumptions that Π has $2^{-p(\lambda)}$ completeness error and $1/q_{\mathsf{uniq}}(\lambda)$-uniqueness, that $\alpha_{\mathsf{crs}}, \beta_{\mathsf{crs}}$ are lower bounded in expectation: $\mathbb{E}_{\mathsf{crs}}[\alpha_{\mathsf{crs}}] \geq 1 - 2^{-p(\lambda)}$, $\mathbb{E}_{\mathsf{crs}}[\beta_{\mathsf{crs}}] \geq 1/q_{\mathsf{uniq}}(\lambda)$. For α_{crs}, we can apply Markov's inequality to $1 - \alpha_{\mathsf{crs}}$ to conclude that

$$\Pr_{\mathsf{crs}}[\alpha_{\mathsf{crs}} \geq 1 - 4q_{\mathsf{uniq}}(\lambda) \cdot 2^{-p(\lambda)}] \geq 1 - 1/4q_{\mathsf{uniq}}(\lambda).$$

By an averaging argument on β_{crs}, it follows that

$$\Pr_{\mathsf{crs}}[\beta_{\mathsf{crs}} \geq 1/2q_{\mathsf{uniq}}(\lambda)] \geq 1/2q_{\mathsf{uniq}}(\lambda).$$

This implies that

$$\Pr_{\mathsf{crs}}[\mathsf{crs} \in G] \geq 1/4q_{\mathsf{uniq}}(\lambda).$$

Now, observe that for each fixing of crs in G, the probability that $b_{\mathsf{crs}} = 1$ and $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$ in the execution of $P^\star(\mathsf{crs})$ is independent. It follows that the probability that $P^\star(\mathsf{crs})$ succeeds is at least

$$\begin{aligned} &\Pr_{\mathsf{crs}}[\mathsf{crs} \in G] \cdot \Pr_{\mathsf{crs}}\left[\begin{array}{l|l} b_{\mathsf{crs}} = 1 & \\ \wedge\ \mathsf{Uniq}(\mathsf{crs}, x^\star) = 1 & \mathsf{crs} \in G \\ \wedge\ x^\star \notin L & \end{array} \right] \\ &\geq (1/4q_{\mathsf{uniq}}(\lambda)) \cdot \Pr[b_{\mathsf{crs}} = 1 \mid \mathsf{crs} \in G] \cdot \Pr[\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1 \mid \mathsf{crs} \in G] - \Pr[x^\star \in L]. \end{aligned}$$

It remains to lower bound each of the terms above.

- We lower bound $\Pr[b_{\mathsf{crs}} = 1 \mid \mathsf{crs} \in G]$ by observing that b_{crs} simply runs $2^{p(\lambda)}$ independent checks, each of which succeeds with probability α_{crs} by assumption. This gives the following bound:

$$\begin{aligned} \Pr[b_{\mathsf{crs}} = 1 \mid \mathsf{crs} \in G] &\geq \Pr\left[b_{\mathsf{crs}} = 1 \mid \alpha_{\mathsf{crs}} \geq 1 - 4q_{\mathsf{uniq}}(\lambda) \cdot 2^{-p(\lambda)}\right] \\ &\geq (1 - 4q_{\mathsf{uniq}}(\lambda) \cdot 2^{-p(\lambda)})^{2^{p(\lambda)}} \\ &\geq (1 - 2^{-(p(\lambda) - 2\log q_{\mathsf{uniq}}(\lambda))})^{2^{p(\lambda) - 2\log q_{\mathsf{uniq}}(\lambda)}} \cdot (1/2)^{2 \log q_{\mathsf{uniq}}(\lambda)} \\ &\geq 1/16q_{\mathsf{uniq}}(\lambda), \end{aligned}$$

  where the last line follows since the sequence $\{(1 - 1/n)^n\}_{n \in \mathbb{N}}$ is increasing and by our assumption that $p(\lambda) > 2 \log q_{\mathsf{uniq}}(\lambda)$.

- Next, to analyze the probability that $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$ for a random $x^\star \leftarrow \mathcal{L}^\star_\lambda$, we observe that this must be close to β_{crs} which is defined the same except over a random $x \leftarrow \mathcal{L}_\lambda$. Namely, for any fixed crs, the process of checking for a unique accepting π requires time $2^{p(\lambda)} \cdot \mathsf{poly}(\lambda) \in 2^{O(\lambda^c)}$. So, it follows from the c-subexponential hardness of the subset membership problem that there exists a negligible function ϵ_1 such that for all $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1 \mid \mathsf{crs} \in G\right] \geq \beta_{\mathsf{crs}} - \epsilon'(\lambda) \geq 1/2q_{\mathsf{uniq}}(\lambda) - \epsilon_1(\lambda).$$

  This is at least $1/3q_{\mathsf{uniq}}(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$.

- Finally, by the correctness of $\mathcal{L}^\star$ for the hard subset membership problem, it follows that there exists a negligible function ϵ_2 such that

$$\Pr[x^\star \in L] \leq \epsilon_2(\lambda).$$

  This is at most $1/384q^3_{\mathsf{uniq}}(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$.

Plugging in the above, we conclude that for sufficiently large $\lambda \in \mathbb{N}$, $P^\star(\mathsf{crs})$ breaks soundness with probability at least

$$(1/4q_{\mathsf{uniq}}(\lambda)) \cdot (1/16q_{\mathsf{uniq}}(\lambda)) \cdot (1/3q_{\mathsf{uniq}}(\lambda)) - 1/384q^3_{\mathsf{uniq}}(\lambda) \geq 1/192q^3_{\mathsf{uniq}}(\lambda).$$

The claim then follows by choosing the polynomial $q(\lambda) = 192q^3_{\mathsf{uniq}}(\lambda)$. □

Claim 3.5 (Simulation)**.** *There exists an efficient algorithm $P^\star_{\mathsf{sim}}$ such that for any non-uniform $2^{O(\lambda^c)}$-size distinguisher D and input crs corresponding to sufficiently large security parameter $\lambda \in \mathbb{N}$, it holds that*

$$\left|\Pr[D(P^\star(\mathsf{crs})) = 1] - \Pr[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1]\right| \leq 2^{-p(\lambda)}.$$

Proof. On input any string crs, the algorithm $P^\star_{\mathsf{sim}}(\mathsf{crs})$ samples $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$ and outputs (x, π) where $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$. Note that $P^\star_{\mathsf{sim}}(\mathsf{crs})$ is runs in time $\mathsf{poly}(\lambda)$ by the efficiency of Sam and Prove, so it is efficient, as required.

We prove the required indistinguishability by constructing a sequence of hybrid experiments as follows:

- $H_0(\mathsf{crs}) := D(P^\star(\mathsf{crs}))$: The first hybrid simply corresponds to the running the distinguisher D on the output of $P^\star(\mathsf{crs})$.

- $H_1(\mathsf{crs})$: In this hybrid, we modify Step 2a in the code of $P^\star$ as follows:

  – 2(a) Sample $(x^\star, w) \leftarrow \mathsf{Sam}(1^\lambda)$. If $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$, output $(x^\star, \pi^\star)$ where $\pi^\star$ is the unique proof such that $\mathsf{Ver}(\mathsf{crs}, x^\star, \pi^\star) = 1$.

  The only change between $H_1(\mathsf{crs})$ and $H_0(\mathsf{crs})$ is whether $x^\star$ is sampled from $\mathcal{L}_\lambda$ or $\mathcal{L}^\star_\lambda$. As the rest of the experiment runs D and otherwise runs in time $2^{2p(\lambda)} \cdot \mathsf{poly}(\lambda) \in 2^{O(\lambda^c)}$, it follows directly from the c-subexponential hardness of the subset membership problem that there exists a negligible function ϵ such that for all $\lambda \in \mathbb{N}$, it holds that

$$\left|\Pr[H_0(\mathsf{crs}) = 1] - \Pr[H_1(\mathsf{crs}) = 1]\right| \leq \epsilon(2^{\lambda^c}).$$

- $H_2(\mathsf{crs})$: We next modify Step 2b in the code of $P^\star$ as follows:

  – 2(b) Perfectly sample $(x, w) \leftarrow \mathsf{Sam}(1^\lambda)$ conditioned on $\mathsf{Uniq}(\mathsf{crs}, x) \neq 1$. Output (x, π) where $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$.

  Note that $H_1(\mathsf{crs})$ and $H_2(\mathsf{crs})$ are identically distributed unless $H_1(\mathsf{crs})$ aborts and outputs $\perp$ in Step 2b, so it suffices to bound the probability of this event occurring.

  Define β_{crs} to be the probability that $\mathsf{Uniq}(\mathsf{crs}, x) = 1$ for crs over the randomness of sampling $(x, w) \leftarrow \mathcal{L}_\lambda$. Then, $H_1(\mathsf{crs})$ aborts only if $\mathsf{Uniq}(\mathsf{crs}, x) = 1$ in Step 2a yet for $2^{p(\lambda)}$ consecutive trials in Step 2b, $\mathsf{Uniq}(\mathsf{crs}, x) \neq 1$. It follows for any possible value of $\beta_{\mathsf{crs}} \in [0, 1]$ that $H_1(\mathsf{crs})$ outputs $\perp$ with probability at most

  $$\beta_{\mathsf{crs}} \cdot (1 - \beta_{\mathsf{crs}})^{2^{p(\lambda)}} < (1/e) \cdot 2^{-p(\lambda)},$$

  where we prove this bound in Lemma B.1, so the experiments differ with probability at most

  $$|\Pr[H_1(\mathsf{crs}) = 1] - \Pr[H_2(\mathsf{crs}) = 1]| < (1/e) \cdot 2^{-p(\lambda)}.$$

- $H_3(\mathsf{crs}) := D(P^\star_{\mathsf{sim}}(\mathsf{crs}))$: In the final hybrid, we fully replace the code of $P^\star$ with the simulator $P^\star_{\mathsf{sim}}$.

  We observe that $H_2(\mathsf{crs})$ and $H_3(\mathsf{crs})$ are identically distributed unless the proof $\pi^\star$ output in Step 2a of $H_2(\mathsf{crs})$ is not equal to $\pi = \mathsf{Prove}(\mathsf{crs}, x^\star, w)$. However, as $\mathsf{Uniq}(\mathsf{crs}, x^\star) = 1$ in this case, there is exactly one accepting proof for $x^\star$, so this can only occur if π isn't accepting and $\mathsf{Comp}(\mathsf{crs}, x^\star, w) \neq 1$.

  So, let α_{crs} denote the probability that $\mathsf{Comp}(\mathsf{crs}, x, w) = 1$ for fixed crs and over the randomness of $(x, w) \leftarrow \mathcal{L}_\lambda$. Then, the probability that π is not accepting in Step 2a is if $b_{\mathsf{crs}} = 1$, meaning $2^{p(\lambda)}$ trials passed, yet $\mathsf{Comp}(\mathsf{crs}, x^\star, w) \neq 1$ on the next trial. This occurs with probability at most

  $$(\alpha_{\mathsf{crs}})^{2^{p(\lambda)}} \cdot (1 - \alpha_{\mathsf{crs}}) < (1/e) \cdot 2^{-p(\lambda)}$$

  again by Lemma B.1 since $\alpha_{\mathsf{crs}} \in [0, 1]$. It follows that the experiments differ with probability at most

  $$|\Pr[H_2(\mathsf{crs}) = 1] - \Pr[H_3(\mathsf{crs}) = 1]| < (1/e) \cdot 2^{-p(\lambda)}.$$

Putting it altogether, it follows that

$$|Pr[D(P^\star(\mathsf{crs})) = 1] - \Pr[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1]| < (2/e) \cdot 2^{-p(\lambda)} + \epsilon(2^{\lambda^c}),$$

where ϵ is a negligible function. As $p(\lambda) \in O(\lambda^c)$ by assumption, it follows that $\epsilon(2^{\lambda^c}) \leq (1 - 2/e) \cdot 2^{-p(\lambda)}$ for sufficiently large $\lambda \in \mathbb{N}$, so

$$|Pr[D(P^\star(\mathsf{crs})) = 1] - \Pr[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1]| \leq 2^{-p(\lambda)},$$

as required for the claim. □

The proof of Lemma 3.2 follows from Claims 3.3, 3.4, and 3.5. □

We are now ready to finish the proof of our main theorem.

Proof of Theorem 3.1. Suppose there exists a probabilistic $2^{O(\lambda^d)}$-time black-box reduction $R^{(\cdot)}$ from the soundness of Π to (C, α). It suffices to prove that the d-subexponential time assumption (C, α) must be false.

Let $P^\star$ be the semi-efficient simulatable Π-adversary guaranteed to exist by Lemma 3.2. Since R is a valid reduction and $P^\star$ a valid Π-adversary for input length $m(\lambda)$, there exists a polynomial q' such that for infinitely many $\lambda \in \mathbb{N}$, it holds that

$$\Pr\left[1 = \langle C, R^{P^\star}\rangle(1^\lambda)\right] > \alpha + 1/q'(2^{\lambda^d}).$$

For simplicity of notation, let $t(\lambda)$ be the combined running time of C and R in the interaction $\langle C, R^{(\cdot)}\rangle(1^\lambda)$ and $\delta(\lambda) = 1/q'(2^{\lambda^d})$, and note that $t(\lambda), 1/\delta(\lambda) \in 2^{O(\lambda^d)}$. We construct an attacker A that depends on t and δ and breaks the assumption (C, α).

> $A(1^\lambda)$ emulates the oracle algorithm $R^{(\cdot)}(1^\lambda)$ in the interaction $\langle C, R^{(\cdot)}\rangle(1^\lambda)$ and responds to oracle queries to $P^\star$ on input crs corresponding to security parameter 1^n as follows:
>
> - If $2^{p(n)} \leq 2t(\lambda)/\delta(\lambda)$, respond with $P^\star(\mathsf{crs})$.
> - Otherwise, when $2^{p(n)} > 2t(\lambda)/\delta(\lambda)$, respond with $P^\star_{\mathsf{sim}}(\mathsf{crs})$.

In terms of running time, observe that $A(1^\lambda)$ runs in time $t(\lambda)$ plus the time to respond to each of at most $t(\lambda)$ queries. We show that responding to each query always takes time at most $2^{O(\lambda^d)}$ by analyzing each case separately:

- When $2^{p(n)} \leq 2t(\lambda)/\delta(\lambda)$, it follows from the efficiency of $P^\star$ that directly computing $P^\star(\mathsf{crs})$ takes time $2^{2p(n)} \cdot \mathsf{poly}(n)$. However, observe that $2^{2p(n)} \leq (2t(\lambda)/\delta(\lambda))^2 \in 2^{O(\lambda^d)}$, and $n \leq t(\lambda) \in 2^{O(\lambda^d)}$ since crs contains 1^n in unary and is computed by R. Thus, the time to respond to queries in this case is at most $2^{O(\lambda^d)}$.
- Whenever $2^{p(n)} > 2t(\lambda)/\delta(\lambda)$, it takes time $\mathsf{poly}(n)$ to respond to the query by the efficiency of $P^\star_{\mathsf{sim}}$. As discussed above, $n \leq t(\lambda) \in 2^{O(\lambda^d)}$, so it follows that responding to such queries also takes time at most $2^{O(\lambda^d)}$.

Thus, the overall running time of $A(1^\lambda)$ is in $2^{O(\lambda^d)}$, so A is has the required efficiency to violate the assumption (C, α).

We next argue correctness of the attacker A via a sequence of hybrid experiments. Specifically, for each $\lambda \in \mathbb{N}$ and $i \in [0, t(\lambda)]$, we define a hybrid experiment H_i as follows:

- $H_i(1^\lambda)$ emulates $\langle C, R^{(\cdot)}\rangle(1^\lambda)$ and answers the first $t(\lambda) - i$ queries by R using $P^\star$ and the last i queries using the strategy of A.

Note that by construction it holds that $H_0(1^\lambda)$ corresponds to the interaction $\langle C, R^{P^\star}\rangle(1^\lambda)$ and $H_{t(\lambda)}(1^\lambda)$ corresponds to $\langle C, A\rangle(1^\lambda)$. It follows by assumption that, for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr\left[H_0(1^\lambda) = 1\right] > \alpha + \delta(\lambda).$$

In the following claim, we argue that, for sufficiently large $\lambda \in \mathbb{N}$, it holds that each pair of hybrids outputs 1 with similar probability.

Claim 3.6. *Let $\lambda \in \mathbb{N}$ be sufficiently large, then for each $i \in [t(\lambda)]$, it holds that*

$$\left|\Pr\left[H_{i-1}(1^\lambda) = 1\right] - \Pr\left[H_i(1^\lambda) = 1\right]\right| \leq \delta(\lambda)/(2t(\lambda)).$$

Proof. The experiments H_{i-1} and H_i are identically distributed except potentially for the response that R receives to its $(t(\lambda)-i+1)$th query. As such, consider fixing the randomness and transcript in the experiment up until this query, and let crs be the subsequent query. The next query response is always generated as $P^\star(\mathsf{crs})$ in H_{i-1} but may be generated as $P^\star_{\mathsf{sim}}(\mathsf{crs})$ in H_i when $2^{p(n)} > 2t(\lambda)/\delta(\lambda)$. When $2^{p(n)} \leq 2t(\lambda)/\delta(\lambda)$, the experiments are identically distributed since H_i also receives $P^\star(\mathsf{crs})$ as a response.

Let D be an algorithm that hardcodes any such transcript and randomness, receives as input the response to the next query, and then continues the emulation of $H_i(1^\lambda)$ until it outputs a bit. It follows by construction that

$$\left|\Pr\left[H_{i-1}(1^\lambda) = 1\right] - \Pr\left[H_i(1^\lambda) = 1\right]\right| \leq \left|\Pr\left[D(P^\star(\mathsf{crs})) = 1\right] - \Pr\left[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1\right]\right|,$$

where crs corresponds to a security parameter n such that $2^{p(n)} > 2t(\lambda)/\delta(\lambda)$. Note, however, that D runs in time $2^{O(\lambda^d)} \in 2^{O(p(n))}$ as all remaining queries are answered using the strategy of A. By Lemma 3.2, it then holds, for sufficiently large $n \in \mathbb{N}$, that

$$\left|\Pr\left[D(P^\star(\mathsf{crs})) = 1\right] - \Pr\left[D(P^\star_{\mathsf{sim}}(\mathsf{crs})) = 1\right]\right| \leq 2^{-p(n)} \leq \delta(\lambda)/(2t(\lambda)).$$

The claim follows for sufficiently large $\lambda \in \mathbb{N}$ that ensure that n is large enough given $2^{p(n)} > 2t(\lambda)/\delta(\lambda)$. □

Given Claim 3.6, it follows for infinitely many $\lambda \in \mathbb{N}$ that

$$\begin{aligned}
\Pr\left[1 = \langle C, A\rangle(1^\lambda)\right] &= \Pr\left[H_{t(\lambda)}(1^\lambda) = 1\right] \\
&\geq \Pr\left[H_0(1^\lambda) = 1\right] - \sum_{i=1}^{t(\lambda)} \left|\Pr\left[H_{i-1}(1^\lambda) = 1\right] - \Pr\left[H_i(1^\lambda) = 1\right]\right| \\
&> \alpha + \delta(\lambda) - t(\lambda) \cdot \delta(\lambda)/(2t(\lambda)) \\
&\geq \alpha + \delta(\lambda)/2.
\end{aligned}$$

As $2/\delta(\lambda) \in 2^{O(\lambda^d)}$, it follows that A violates the d-subexponential time assumption (C, α), as required. □

4 Constructing Unique SNARGs with Adaptive Security

In this section, we show how to circumvent the lower bound of Section 3 by relaxing completeness, resulting in the following theorem.

Theorem 4.1 (Adaptively Secure Unique SNARG)**.** *Let L be any NP language with witness relation R. Assuming the existence of sub-exponentially secure iO for circuits and OWFs, there exists a SNARG for R that on security parameter λ and inputs of length m has CRS length $\mathsf{poly}(\lambda + m)$, proof length $\mathsf{poly}(\lambda)$, 1-uniqueness, and $2^{-\lambda}$-completeness error.*

To prove the theorem, we first define the necessary ingredients in Section 4.1, then give the construction in Section 4.2, and prove the necessary properties in Section 4.3.

4.1 Ingredients

Our construction relies on injective OWFs with inefficient samplers, puncturable PRFs, and iO for circuits, all of which are implied by only iO and OWF as required for Theorem 4.1.

For simplicity in this section, we adopt the notion of c-subexponential security from [WW25], but we note that it is implied by the notion of c'-subexponential security from Definition 2.4 for any $c' > c$. As such, these assumptions are also ruled out by our black-box separation from Theorem 3.1.

Injective one-way functions with inefficient samplers. An (injective) OWF is an efficiently computable (injective) function that is hard to invert for a uniform input. As in [WW25], we consider a modified notion that allows for inefficient sampling, captured by the following definition.

Definition 4.2 (Injective One-Way Functions with Inefficient Samplers)**.** Let $c > 0$ be a constant. We say that a pair of algorithms $(\mathsf{GenChal}, \mathsf{VerChal})$ is a *c-subexponentially secure injective one-way function with an inefficient sampler* with the following syntax and efficiency:

- $(y, x) \leftarrow \mathsf{GenChal}(1^\lambda)$: A randomized (potentially inefficient) algorithm that on input the security parameter 1^λ outputs a challenge $y \in \{0,1\}^{\mathsf{poly}(\lambda)}$ with corresponding solution $x \in \{0,1\}^{\mathsf{poly}(\lambda)}$.
- $b = \mathsf{VerChal}(y, x)$: A deterministic polynomial-time algorithm that on input a challenge y with potential solution x outputs a bit b.

We require that $(\mathsf{GenChal}, \mathsf{VerChal})$ satisfy the following properties:

- **Correctness:** For all $\lambda \in \mathbb{N}$, it holds that
$$\Pr\left[(y, x) \leftarrow \mathsf{GenChal}(1^\lambda) : \mathsf{VerChal}(y, x) = 1\right] = 1.$$
- **Injectivity:** For all $\lambda \in \mathbb{N}$, $(y, x) \in \mathrm{Supp}\left(\mathsf{GenChal}(1^\lambda)\right)$, and $x' \neq x$, it holds that
$$\mathsf{VerChal}(y, x') = 0.$$
- **One-wayness:** For all non-uniform PPT attackers A, there exists $\lambda_A \in \mathbb{N}$ such that for all $\lambda \geq \lambda_A$, it holds that
$$\Pr\left[\begin{array}{l}(y, x) \leftarrow \mathsf{GenChal}(1^\lambda) \\ x' \leftarrow A(1^\lambda, y)\end{array} : \mathsf{VerChal}(y, x') = 1\right] \leq 2^{-\lambda^c}.$$

Waters and Wu [WW25] show how to construct injective OWFs with inefficient samplers from plain OWFs. Furthermore, their construction is sub-exponentially secure if the underlying OWF is.

Theorem 4.3 ([WW25])**.** *Assuming the existence of subexponentially secure OWFs, there exist subexponentially secure injective OWFs with inefficient samplers*

We note that the construction of standard injective OWFs from [BPW16] based on iO and OWF does not suffice for our construction since their construction only achieves statistical rather than perfect injectivity. Adapting their construction to get perfect injectivity requires inefficient sampling for the same reason as in [WW25].

Puncturable pseudorandom functions. A PRF is an efficient keyed function that is indistinguishable from a random function without access to the key. A *puncturable* PRF allows a key to be punctured at a specific input $x^\star$ such that the function is pseudorandom at $x^\star$ even given the punctured key.

Definition 4.4 (Puncturable PRFs)**.** Let $c > 0$ be a constant. We say that a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Puncture})$ is a *c-subexponentially secure puncturable pseudorandom function* with the following syntax and efficiency:

- $k \leftarrow \mathsf{KeyGen}(1^\lambda, 1^{\ell_{\mathsf{in}}}, 1^{\ell_{\mathsf{out}}})$: A randomized polynomial-time algorithm that takes as input a security parameter 1^λ, input length $1^{\ell_{\mathsf{in}}}$, and output length $1^{\ell_{\mathsf{out}}}$, and outputs a key k. We assume that the key k implicitly specifies ℓ_{in} and ℓ_{out}.
- $k^{(x^\star)} = \mathsf{Puncture}(k, x^\star)$: A deterministic polynomial-time algorithm that takes as input a key k and an input $x^\star \in \{0,1\}^{\ell_{\mathsf{in}}}$ and outputs a punctured key $k^{(x^\star)}$ for the same input and output length.
- $y = \mathsf{Eval}(k, x)$: A deterministic polynomial-time algorithm that takes as input a key k and an input $x \in \{0,1\}^{\ell_{\mathsf{in}}}$ and outputs a value $y \in \{0,1\}^{\ell_{\mathsf{out}}}$.

We require that $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Puncture})$ satisfy the following properties:

- **Functionality-Preserving:** For all $\lambda, \ell_{\mathsf{in}}, \ell_{\mathsf{out}} \in \mathbb{N}$, punctured inputs $x^\star \in \{0,1\}^{\ell_{\mathsf{in}}}$, and other inputs $x \in \{0,1\}^{\ell_{\mathsf{in}}}$ such that $x \neq x^\star$, it holds that

$$\Pr\left[\begin{array}{l} k \leftarrow \mathsf{KeyGen}(1^\lambda, 1^{\ell_{\mathsf{in}}}, 1^{\ell_{\mathsf{out}}}) \\ k^{(x^\star)} = \mathsf{Puncture}(k, x^\star) \end{array} : \mathsf{Eval}(k, x) = \mathsf{Eval}(k^{(x^\star)}, x)\right] = 1.$$

- **Punctured Pseudorandomness:** For all stateful non-uniform PPT attackers A, there exists $\lambda_A \in \mathbb{N}$ such that for all $\lambda \geq \lambda_A$, it holds that

$$\Pr\left[\begin{array}{l} (1^{\ell_{\mathsf{in}}}, 1^{\ell_{\mathsf{out}}}, x^\star) \leftarrow A(1^\lambda) \\ k \leftarrow \mathsf{KeyGen}(1^\lambda, 1^{\ell_{\mathsf{in}}}, 1^{\ell_{\mathsf{out}}}) \\ k^{(x^\star)} = \mathsf{Puncture}(k, x^\star) \\ b \leftarrow \{0,1\} \\ y_0 = \mathsf{Eval}(k, x^\star) \\ y_1 \leftarrow \{0,1\}^{\ell_{\mathsf{out}}} \\ b' \leftarrow A(k^{(x^\star)}, y_b) \end{array} : b' = b\right] \leq 1/2 + 2^{-\lambda^c}.$$

We say that the puncturable PRF is simply subexponentially secure if there exists a constant $c > 0$ such that it is c-subexponentially secure.

The standard GGM PRF construction [GGM84] from OWFs can be adapted to give a puncturable PRF as observed by [BW13, BGI14, KPTZ13, SW14].

Theorem 4.5 ([GGM84, SW14])**.** *Assuming the existence of sub-exponentially secure one-way functions, there exist sub-exponentially secure puncturable PRFs.*

Indistinguishability obfuscation. An indistinguishability obfuscator (iO) for circuits is an efficient algorithm that produces an obfuscated circuit C' of some circuit C, which is indistinguishable from any obfuscation of another functionally equivalent circuit of the same size as C.

Definition 4.6 (Indistinguishability Obfuscation)**.** Let $c > 0$ be a constant. We say that an algorithm Obf is a *c-subexponentially secure indistinguishability obfuscator (iO) for circuits* if it has the following syntax and efficiency:

- $C' \leftarrow \mathsf{Obf}(1^\lambda, 1^s, C)$: A randomized polynomial-time algorithm that takes as input a security parameter 1^λ, circuit size 1^s, and circuit C, and outputs a new circuit C'.

We require that Obf satisfies the following properties:

- **Correctness:** For all $\lambda, s \in \mathbb{N}$ and circuits C of size at most s, it holds that

$$\Pr\left[C' \leftarrow \mathsf{Obf}(1^\lambda, 1^s, C) : C' \equiv C\right] = 1.$$

- **Indistinguishability:** For all stateful non-uniform PPT attackers A, there exists $\lambda_A \in \mathbb{N}$ such that for all $\lambda \geq \lambda_A$, it holds that

$$\Pr\left[\begin{array}{l} b \leftarrow \{0,1\} \\ (1^s, C_0, C_1) \leftarrow A(1^\lambda) \\ C' \leftarrow \mathsf{Obf}(1^\lambda, 1^s, C_b) \\ b' \leftarrow A(C') \end{array} : \begin{array}{l} |C_0| = |C_1| \leq s \\ \wedge\ C_0 \equiv C_1 \\ \wedge\ b' = b \end{array}\right] \leq 2^{-\lambda^c}.$$

We say that Obf is simply subexponentially secure if there exists a constant $c > 0$ such that it is c-subexponentially secure.

Jain, Lin, and Sahai [JLS21, JLS22] show how to construct iO for circuits from the following falsifiable subexponential assumptions. We defer to [JLS22] for precise definitions and parameter ranges for these assumptions.

Theorem 4.7 ([JLS22])**.** *Assuming subexponential security of the LPN and DLIN assumptions and the existence of subexponentially secure boolean PRGs in NC^0, there exist a subexponentially secure iO for circuits.*

4.2 Construction

Let L be the NP language with witness relation R corresponding to input-length $|x| = n$ and $|(x, w)| = m$. Our construction relies on the following ingredients:

- Let Obf be an iO for circuits with c_{obf}-subexponential security for some constant $c_{\mathsf{obf}} > 0$. Let $\lambda_{\mathsf{obf}} = (2\lambda + n)^{1/c_{\mathsf{obf}}}$ be the iO security parameter.
- Let $(\mathsf{KeyGen}, \mathsf{Puncture}, \mathsf{Eval})$ be a puncturable PRF with c_{prf}-subexponential security for constant $c_{\mathsf{prf}} > 0$. Let $\lambda_{\mathsf{prf}} = (2\lambda + n)^{1/c_{\mathsf{prf}}}$ be the PRF security parameter.
- Let $(\mathsf{GenChal}, \mathsf{VerChal})$ be an injective OWF with an inefficient sampler that has input length $\ell(\lambda) \in \mathsf{poly}(\lambda)$ with c_{owf}-subexponential security for constant $c_{\mathsf{owf}} > 0$. Let $\lambda_{\mathsf{owf}} = (2\lambda)^{1/c_{\mathsf{owf}}}$.

Our construction $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ closely follows the blueprint of [WW25]. The setup algorithm Gen outputs two obfuscated circuits $\mathsf{GenProof}'$ and $\mathsf{VerProof}'$. At a high level, the honest prover computes $\mathsf{GenProof}$ on input (x, w) to get a PRF signature on x, and $\mathsf{VerProof}$ simply checks

the signature is valid. More concretely, $\mathsf{GenProof}$ only outputs a signature for x if $(x, w) \in R$ *and* a different "selector" PRF on x doesn't hit a random hardcoded value $z \in \{0,1\}^\lambda$. Otherwise, the program aborts and outputs $\perp$ (meaning the honest prover won't find a proof for x). We formalize this below.

$\mathsf{Gen}(1^\lambda, 1^m)$:

1. Sample PRF keys $k_{\mathsf{sel}} \leftarrow \mathsf{KeyGen}(1^{\lambda_{\mathsf{prf}}}, 1^n, 1^\lambda)$ and $k \leftarrow \mathsf{KeyGen}(1^{\lambda_{\mathsf{prf}}}, 1^n, 1^{\ell(\lambda_{\mathsf{owf}})})$.
2. Sample a random value $z \leftarrow \{0,1\}^\lambda$.
3. Define the program $\mathsf{GenProof}[k, k_{\mathsf{sel}}, z](x, w)$ as follows:
   - If $(x, w) \notin R$, output $\perp$.
   - If $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z$, output $\perp$.
   - Otherwise, output $\mathsf{Eval}(k, x)$.
4. Define the program $\mathsf{VerProof}[k](x, \pi)$ as follows:
   - Output 1 iff $\mathsf{Eval}(k, x) = \pi$.
5. Let $s \in \mathsf{poly}(\lambda + m)$ denote the maximum size of the programs $\mathsf{GenProof}$ and $\mathsf{VerProof}$ in the hybrids from the security analysis below.
6. Compute the obfuscated circuits $\mathsf{GenProof}' \leftarrow \mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{GenProof}[k, k_{\mathsf{sel}}, z])$ and $\mathsf{VerProof}' \leftarrow \mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{VerProof}[k])$.
7. Output $(\mathsf{GenProof}', \mathsf{VerProof}')$.

$\mathsf{Prove}(\mathsf{crs}, x, w)$: Parse $\mathsf{crs} = (\mathsf{GenProof}', \mathsf{VerProof}')$ and output $\pi = \mathsf{GenProof}'(x, w)$.

$\mathsf{Ver}(\mathsf{crs}, x, \pi)$: Parse $\mathsf{crs} = (\mathsf{GenProof}', \mathsf{VerProof}')$ and output $\mathsf{VerProof}'(x, \pi)$.

4.3 Analysis

It remains to argue that the construction above satisfies the required efficiency, uniqueness, completeness, and soundness properties for Theorem 4.1. We prove these properties in the following lemmas.

Lemma 4.8 (Efficiency)**.** *The algorithms* Gen, Prove, *and* Ver *run in polynomial-time and for any* $\lambda, m \in \mathbb{N}$, $(x, w) \in R \cap \{0,1\}^m$, *and* $\mathsf{crs} \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda, 1^m)\right)$, *it holds that* $|\mathsf{crs}| \in \mathsf{poly}(\lambda + m)$ *and* $|\pi| \in \mathsf{poly}(\lambda)$.

Proof. Efficiency of Gen follows from the efficiency of the relation R, KeyGen, Eval, and Obf, and the bounds on λ_{obf}, λ_{prf}, and s. Furthermore, s is polynomially bounded also due to the efficiency of $\mathsf{Puncture}$ and $\mathsf{VerChal}$ (see Lemma 4.11). This immediately implies the existence of a polynomial q such that $|\mathsf{crs}|$, $|\mathsf{GenProof}'|$, $|\mathsf{VerProof}'| \leq q(\lambda + m) \in \mathsf{poly}(\lambda + m)$, which also gives the required efficiency for Prove and Ver.

For succinctness, we note that for any $\lambda, m \in \mathbb{N}$, statement $(x, w) \in R \cap \{0,1\}^m$, and $\mathsf{crs} \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda, 1^m)\right)$, the honest proof $\pi = \mathsf{Prove}(\mathsf{crs}, x, w)$ is either $\perp$ or equal to $\mathsf{Eval}(k, x)$ where $k \in \mathrm{Supp}\left(\mathsf{KeyGen}(\cdot, \cdot, 1^{\ell(\lambda_{\mathsf{owf}})})\right)$. It follows that $|\pi| \leq \ell(\lambda_{\mathsf{owf}}) = \ell((2\lambda)^{1/c_{\mathsf{owf}}}) \in \mathsf{poly}(\lambda)$ by the syntax of KeyGen and Eval and since ℓ is polynomially bounded. □

Lemma 4.9 (Uniqueness)**.** *For all $\lambda, m \in \mathbb{N}$, $\mathsf{crs} \in \mathrm{Supp}\left(\mathsf{Gen}(1^\lambda, 1^m)\right)$, and statement x corresponding to inputs of R of length m, it holds that*

$$|\{\pi : \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1\}| = 1.$$

Proof. We observe that $\mathsf{VerProof}[k](x, \pi)$ only outputs 1 for the unique value $\pi = \mathsf{Eval}(k, x)$ by construction. It follows that the same holds for $\mathsf{VerProof}'$ by the correctness of Obf, so this also holds for Ver since $\mathsf{Ver}(\mathsf{crs}, x, \pi)$ simply outputs $\mathsf{VerProof}'(x, \pi)$. We note that this holds for false statements as well as true statements. □

Lemma 4.10 (Completeness)**.** *For all $\lambda, m \in \mathbb{N}$ and $(x, w) \in R \cap \{0,1\}^m$, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^m) \\ \pi = \mathsf{Prove}(\mathsf{crs}, x, w)\end{array} : \mathsf{Ver}(\mathsf{crs}, x, \pi) \neq 1\right] = 2^{-\lambda}.$$

Proof. Let $\pi = \mathsf{Prove}(\mathsf{crs}, x, \pi)$ in the experiment above. By correctness of Obf, it follows that π is computed according to $\mathsf{GenProof}[k, k_{\mathsf{sel}}, z](x, w)$ for a random value $z \leftarrow \{0,1\}^\lambda$. For any fixed key k_{sel} and input x, it holds that $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z'$ for some value $z' \in \{0,1\}^\lambda$, so the probability that $z' = z$ is exactly $2^{-\lambda}$ since z and z' are independent. This implies that $\pi = \perp$ with probability $2^{-\lambda}$ and is equal to $\mathsf{Eval}(k, x)$ otherwise. By construction of $\mathsf{VerProof}$ and correctness of Obf, it holds that $\mathsf{Ver}(\mathsf{crs}, x, \pi) = \mathsf{VerProof}'(x, \pi) = 1$ iff $\mathsf{Eval}(k, x) = \pi$. It follows that $\mathsf{Ver}(\mathsf{crs}, x, \pi) \neq 1$ with probability exactly $2^{-\lambda}$, as required. □

Lemma 4.11 (Adaptive Soundness)**.** *The construction $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ satisfy adaptive soundness.*

The proof of soundness follows essentially the same structure as in [WW24, WW25]. The main difference is we lose a factor of $2^{-\lambda}$ in the advantage by checking that the adversary outputs a statement x such that $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z$. The rest of the hybrids and reductions are essentially identical, so we defer the proof details to Appendix A.

References

[ADSS24] Gorjan Alagic, Dana Dachman-Soled, Manasi Shingane, and Patrick Struck. Quantum black-box separations: Succinct non-interactive arguments from falsifiable assumptions. *IACR Cryptol. ePrint Arch.*, page 1763, 2024.

[BBK+23] Zvika Brakerski, Maya Farber Brodsky, Yael Tauman Kalai, Alex Lombardi, and Omer Paneth. Snargs for monotone policy batch NP. In *CRYPTO (2)*, volume 14082 of *Lecture Notes in Computer Science*, pages 252–283. Springer, 2023.

[BCH+22] Nir Bitansky, Arka Rai Choudhuri, Justin Holmgren, Chethan Kamath, Alex Lombardi, Omer Paneth, and Ron D. Rothblum. Ppad is as hard as lwe and iterated squaring. In *TCC*, 2022.

[BD25] Marshall Ball and Dana Dachman-Soled. Uniform black-box separations via non-malleable extractors. In *CRYPTO (1)*, volume 16000 of *Lecture Notes in Computer Science*, pages 585–616. Springer, 2025.

[BDG+13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "fiat-shamir for proofs" lacks a proof. In *TCC*, volume 7785 of *Lecture Notes in Computer Science*, pages 182–201. Springer, 2013.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519. Springer, 2014.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. On the security of one-witness blind signature schemes. In *ASIACRYPT (2)*, volume 8270 of *Lecture Notes in Computer Science*, pages 82–99. Springer, 2013.

[BP04] Boaz Barak and Rafael Pass. On the possibility of one-message weak zero-knowledge. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 121–132, 2004.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In *TCC (A1)*, volume 9562 of *Lecture Notes in Computer Science*, pages 474–502. Springer, 2016.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71. Springer, 1998.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *ASIACRYPT (2)*, volume 8270 of *Lecture Notes in Computer Science*, pages 280–300. Springer, 2013.

[CFJW25] Liyan Chen, Cody Freitag, Zhengzhong Jin, and Daniel Wichs. Unambiguous snargs for P from LWE with applications to PPAD hardness. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 944–954. ACM, 2025.

[CGJ+23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and snargs from sub-exponential DDH. In *CRYPTO (4)*, volume 14084 of *Lecture Notes in Computer Science*, pages 635–668. Springer, 2023.

[CGKS23] Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. Impossibilities in succinct arguments: Black-box extraction and more. In *AFRICACRYPT*, volume 14064 of *Lecture Notes in Computer Science*, pages 465–489. Springer, 2023.

[CHK+19a] Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. Finding a Nash equilibrium is no easier than breaking

Fiat-Shamir. In *51st ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1103–1114, 2019.

[CHK+19b] Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. PPAD-hardness via iterated squaring modulo a composite. *IACR Cryptology ePrint Archive*, 2019:667, 2019.

[CJ25] Liyan Chen and Zhengzhong Jin. On the impossibility of snargs with short crs (or: Revisiting gentry-wichs barrier in the non-adaptive setting). In *FOCS*, 2025.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for P from LWE. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 68–79, 2021.

[CLMP13] Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In *ITCS*, pages 389–400. ACM, 2013.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from P-certificates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 50–59, 2013.

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *CRYPTO (1)*, volume 9215 of *Lecture Notes in Computer Science*, pages 287–307. Springer, 2015.

[CPW20] Suvradip Chakraborty, Manoj Prabhakaran, and Daniel Wichs. Witness maps and applications. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 220–246. Springer, 2020.

[EFKP20] Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. Continuous verifiable delay functions. In *EUROCRYPT (3)*, volume 12107 of *Lecture Notes in Computer Science*, pages 125–154. Springer, 2020.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479. IEEE Computer Society, 1984.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, pages 99–108. ACM, 2011.

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. Snargs for P from sub-exponential DDH and QR. In *EUROCRYPT (2)*, volume 13276 of *Lecture Notes in Computer Science*, pages 520–549. Springer, 2022.

[IR88] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 8–26. Springer, 1988.

[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 *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 708–721, 2021.

[JKLM25] Zhengzhong Jin, Yael Tauman Kalai, Alex Lombardi, and Surya Mathialagan. Universal snargs for NP from proofs of correctness. In *STOC*, pages 933–943. ACM, 2025.

[JKLV24] Zhengzhong Jin, Yael Kalai, Alex Lombardi, and Vinod Vaikuntanathan. Snargs under LWE via propositional proofs. In *STOC*, pages 1750–1757. ACM, 2024.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, pages 60–73. ACM, 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb {F}_p$, dlin, and prgs in nc^0. In *EUROCRYPT (1)*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2022.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, STOC*, pages 723–732, 1992.

[Kiy21] Susumu Kiyoshima. Black-box impossibilities of obtaining 2-round weak ZK and strong WI from polynomial hardness. In *TCC (1)*, volume 13042 of *Lecture Notes in Computer Science*, pages 369–400. Springer, 2021.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *CCS*, pages 669–684. ACM, 2013.

[KPY19] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In *STOC*, pages 1115–1124. ACM, 2019.

[KPY20] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. Delegation with updatable unambiguous proofs and ppad-hardness. In *CRYPTO (3)*, volume 12172 of *Lecture Notes in Computer Science*, pages 652–673. Springer, 2020.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *STOC*, pages 485–494. ACM, 2014.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and snargs. In *TCC (1)*, volume 13042 of *Lecture Notes in Computer Science*, pages 330–368. Springer, 2021.

[LV20] Alex Lombardi and Vinod Vaikuntanathan. Fiat-shamir for repeated squaring with applications to ppad-hardness and vdfs. In *Advances in Cryptology - CRYPTO*, pages 632–651, 2020.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science, FOCS*, pages 436–453, 1994.

[MW20] Tal Moran and Daniel Wichs. Incompressible encodings. In *CRYPTO (1)*, volume 12170 of *Lecture Notes in Computer Science*, pages 494–523. Springer, 2020.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology - CRYPTO*, pages 96–109, 2003.

[Pas06] Rafael Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on np-hardness. In *CCC*, pages 96–110. IEEE Computer Society, 2006.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In *STOC*, pages 109–118. ACM, 2011.

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In *TCC*, volume 7785 of *Lecture Notes in Computer Science*, pages 334–354. Springer, 2013.

[PTV11] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Towards non-black-box lower bounds in cryptography. In *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 579–596. Springer, 2011.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484. ACM, 2014.

[Wee05] Hoeteck Wee. On round-efficient argument systems. In *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 140–152. Springer, 2005.

[WW22] Brent Waters and David J. Wu. Batch arguments for np and more from standard bilinear group assumptions. In *CRYPTO (2)*, volume 13508 of *Lecture Notes in Computer Science*, pages 433–463. Springer, 2022.

[WW24] Brent Waters and David J. Wu. Adaptively-sound succinct arguments for NP from indistinguishability obfuscation. In *STOC*, pages 387–398. ACM, 2024.

[WW25] Brent Waters and David J. Wu. A pure indistinguishability obfuscation approach to adaptively-sound snargs for NP. In *CRYPTO*, 2025.

[WZ24] Brent Waters and Mark Zhandry. Adaptive security in snargs via io and lossy functions. In *CRYPTO (10)*, volume 14929 of *Lecture Notes in Computer Science*, pages 72–104. Springer, 2024.

A Proof of Lemma 4.11

We proceed to give the main ideas for the proof of soundness for our unique SNARG construction in Section 4.

Lemma A.1 (Adaptive Soundness)**.** *The construction* $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ *of Section 4.2 satisfies adaptive soundness.*

Proof. Our proof follows essentially the same structure as in [WW24, WW25]. For simplicity, we refer to the proof in [WW25] when relevant and omit the details that follow immediately from their analysis.

Suppose by way of contradiction that there exists a non-uniform PPT attacker A and polynomials m, q such that for infinitely many $\lambda \in \mathbb{N}$, it holds that

$$\Pr\left[\begin{array}{l|l} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)}) & \mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \\ (x, \pi) \leftarrow A(\mathsf{crs}) & \wedge\ x \notin L \end{array}\right] > 1/q(\lambda).$$

Towards a contradiction, we define a sequence of hybrid experiments as follows for each $\lambda \in \mathbb{N}$:

- $H_0(\lambda)$: The first experiment corresponds to the adaptive soundness game. Namely, the experiment samples $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda, 1^{m(\lambda)})$, computes $(x, \pi) \leftarrow A(\mathsf{crs})$, and outputs 1 iff
$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L.$$
- $H_1(\lambda)$: Let k_{sel} and z be the values sampled by Gen. This hybrid is the same as H_0 except it outputs 1 iff
$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L \wedge \mathsf{Eval}(k_{\mathsf{sel}}, x) = z.$$
- $H_2(\lambda)$: This hybrid stops checking that $x \notin L$. Namely, it outputs 1 iff
$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge \mathsf{Eval}(k_{\mathsf{sel}}, x) = z.$$
- $H_3(\lambda)$: This hybrid changes the way that $\mathsf{VerProof}$ is constructed in Gen. Specifically, $\mathsf{Gen}(1^\lambda, 1^m)$ is changed as follows:
  - Sample new independent values $(y^\star, x^\star) \leftarrow \mathsf{GenChal}(1^\lambda)$.
  - Redefine the program $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star](x, \pi)$ as follows:
    - If $\mathsf{Eval}(k_{\mathsf{sel}}, x) \neq z$, output 1 iff $\mathsf{Eval}(k, x) = \pi$.
    - Otherwise, output 1 iff $\mathsf{VerChal}(y^\star, \pi \oplus \mathsf{Eval}(k, x))$.
  - The program $\mathsf{VerProof}'$ is now computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star])$ using the program above.

Claim A.2. *For sufficiently large* $\lambda \in \mathbb{N}$*, it holds that*

$$\Pr[H_1(\lambda) = 1] \geq 2^{-\lambda} \cdot \Pr[H_0(\lambda) = 1] - 3 \cdot 2^{-2\lambda}.$$

Proof. This proof closely follows the proof of Lemma 4.4 in [WW25]. The main difference is that we lose a factor of 2^λ in the probability instead of 2 since our selector PRF outputs λ bits rather than 1. We give the structure of the main argument for completeness while omitting details that follow directly from [WW25].

For each $j \in \{0,1\}$ and $i \in \{0,1\}^n$, we define $H_j^{(i)}(\lambda) = 1$ iff $H_j(\lambda) = 1$ and the adversary A outputs the statement $x = i$. As such, we can rewrite the probability space for H_j for $j \in \{0,1\}$ as follows:

$$\Pr[H_j(\lambda) = 1] = \sum_{i \in \{0,1\}^n} \Pr[H_j^{(i)}(\lambda) = 1]$$

We will show for all $i \in \{0,1\}^n$ that

$$\Pr[H_1^{(i)}(\lambda) = 1] \geq 2^{-\lambda} \cdot \Pr[H_0^{(i)}(\lambda) = 1] - 3 \cdot 2^{-(2\lambda+n)}.$$

Note that this trivially holds when $x = i \in L$ as

$$\Pr[H_1^{(i)}(\lambda) = 1] = 0 = \Pr[H_0^{(i)}(\lambda) = 1].$$

As such, we only need to consider the case where $x = i \notin L$. To analyze this case, we introduce an additional sequence of sub-hybrids:

- $H_{0,0}^{(i)}(\lambda)$: This hybrid is equivalent to $H_0^{(i)}(\lambda)$. In particular, the experiment outputs 1 iff

$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L \wedge x = i.$$

- $H_{0,1}^{(i)}(\lambda)$: Same as $H_{0,0}^{(i)}(\lambda)$ except this hybrid changes the way that $\mathsf{GenProof}'$ is constructed in Gen as follows:
  - After sampling k_{sel}, compute the punctured key $k_{\mathsf{sel}}^{(i)} = \mathsf{Puncture}(k_{\mathsf{sel}}, i)$.
  - The program $\mathsf{GenProof}'$ is now computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{GenProof}[k, k_{\mathsf{sel}}^{(i)}, z])$ using the punctured key $k_{\mathsf{sel}}^{(i)}$ instead of k_{sel}.

- $H_{0,2}^{(i)}(\lambda)$: Same as $H_{0,1}^{(i)}(\lambda)$ except this experiment samples an independent random value $z' \leftarrow \{0,1\}^\lambda$ and outputs 1 iff

$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L \wedge x = i \wedge z' = z.$$

- $H_{0,3}^{(i)}(\lambda)$: Same as $H_{0,2}^{(i)}(\lambda)$ except this experiment outputs 1 iff

$$\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L \wedge x = i \wedge \mathsf{Eval}(k_{\mathsf{sel}}, x) = z.$$

- $H_{0,4}^{(i)}(\lambda)$: Same as $H_{0,3}^{(i)}(\lambda)$ except the experiment reverts to computing $\mathsf{GenProof}'$ as the output of $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{GenProof}[k, k_{\mathsf{sel}}, z])$ using k_{sel} again instead of $k_{\mathsf{sel}}^{(i)}$.

It is clear by construction that $H_0^{(i)}(\lambda) \equiv H_{0,0}^{(i)}(\lambda)$ and $H_{0,4}^{(i)}(\lambda) \equiv H_1^{(i)}(\lambda)$. In the following subclaims, we consider each pair of adjacent sub-hybrids.

Subclaim A.3. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H_{0,1}^{(i)}(\lambda) = 1] \geq \Pr[H_{0,0}^{(i)}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This proof follows identically to the proof of Claim A.1 in [WW25], so we omit the details. At a high level, since $x = i \notin L$, it holds that $\mathsf{GenProof}[k, k_{\mathsf{sel}}, z]$ and $\mathsf{GenProof}[k, k^{(i)}_{\mathsf{sel}}, z]$ are functionally equivalent, so the subclaim follows by appealing to the indistinguishability security of Obf. □

Subclaim A.4. *For all $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{0,2}(\lambda) = 1] = 2^{-\lambda} \cdot \Pr[H^{(i)}_{0,1}(\lambda) = 1].$$

Proof. This follows since the only difference between $H_{0,2}(\lambda)$ and $H_{0,1}(\lambda)$ is the extra condition that $z' = z$ for randomly chosen $z' \leftarrow \{0,1\}^\lambda$. As z' and z are independent, $z' = z$ with probability $2^{-\lambda}$. The subclaim follows as z' is also independent of the rest of the experiment. □

Subclaim A.5. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{0,3}(\lambda) = 1] \geq \Pr[H^{(i)}_{0,2}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This proof follows identically to the proof of Claim A.3 in [WW25], so we omit the details. At a high level, the subclaim follows by appealing to the punctured pseudorandomness of $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Puncture})$ to indistinguishably swap a random value $z' \leftarrow \{0,1\}^\lambda$ with $\mathsf{Eval}(k_{\mathsf{sel}}, i)$ even given the punctured key $k^{(i)}_{\mathsf{sel}}$. □

Subclaim A.6. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{0,4}(\lambda) = 1] \geq \Pr[H^{(i)}_{0,3}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This follows identically to Subclaim A.3 by appealing to the indistinguishability security of Obf. □

Combining the subclaims above, it holds for all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$ that

$$\Pr[H^{(i)}_1(\lambda) = 1] \geq 2^{-\lambda} \cdot \Pr[H^{(i)}_{0,0}(\lambda) = 1] - 3 \cdot 2^{-(2\lambda+n)}.$$

Summing over all values of $i \in \{0,1\}^n$, it follows that

$$\Pr[H_1(\lambda) = 1] \geq 2^{-\lambda} \cdot \Pr[H_{0,0}(\lambda) = 1] - 3 \cdot 2^{-2\lambda},$$

as required for Claim A.2. □

Claim A.7. *For all $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H_2(\lambda) = 1] \geq \Pr[H_1(\lambda) = 1].$$

Proof. Same as in Lemma 4.5 of [WW25], this holds immediately since the conditions for outputting 1 in H_2 are a strict subset of those in H_1. □

Claim A.8. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H_3(\lambda) = 1] \geq \Pr[H_2(\lambda) = 1] - 12 \cdot 2^{-2\lambda}.$$

Proof. This proof closely follows the proof of Lemma 4.6 from [WW25]. We give the structure of the main argument for completeness while omitting details that follow directly from [WW25].

For each $i \in \{0,1\}^n$, we define a sequence of sub-hybrids. For simplicity, we correspond each index $i \in \{0,1\}^n$ with its associated integer value in $[0, 2^n - 1]$ when comparing different values.

- $H^{(i)}_{2,0}(\lambda)$: Compared to $H_2(\lambda)$, this hybrid changes the way the $\mathsf{VerProof}'$ is constructed in Gen as follows:
  - Sample new independent values $(y^\star, x^\star) \leftarrow \mathsf{GenChal}(1^\lambda)$.
  - Redefine the program $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i](x, \pi)$ as follows:
    - If $\mathsf{Eval}(k_{\mathsf{sel}}, x) \neq z$ or $x \geq i$, output 1 iff $\mathsf{Eval}(k, x) = \pi$.
    - Otherwise, output 1 iff $\mathsf{VerChal}(y^\star, \pi \oplus \mathsf{Eval}(k, x))$.
  - The program $\mathsf{VerProof}'$ is now computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i])$ using the program above.

  As in $H_2(\lambda)$, the experiment outputs 1 iff $\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1$ and $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z$.

  We also define this experiment for the special case of $i = 2^n$ where note that $x < 2^n$ for all $x \in \{0,1\}^n$.

- $H^{(i)}_{2,1}(\lambda)$: Same as $H^{(i)}_{2,0}(\lambda)$ except this experiment again changes the way $\mathsf{VerProof}'$ is constructed:
  - Set $r^\star = x^\star \oplus \mathsf{Eval}(k, i)$.
  - Redefine the program $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star](x, \pi)$ as follows:
    - If $\mathsf{Eval}(k_{\mathsf{sel}}, x) \neq z$ or $x > i$, output 1 iff $\mathsf{Eval}(k, x) = \pi$.
    - If $x = i$, output 1 iff $\pi \oplus r^\star = x^\star$.
    - Otherwise, output 1 iff $\mathsf{VerChal}(y^\star, \pi \oplus \mathsf{Eval}(k, x))$.
  - The program $\mathsf{VerProof}'$ is now computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star])$ using the program above.

- $H^{(i)}_{2,2}(\lambda)$: Same as $H^{(i)}_{2,1}(\lambda)$ except the experiment punctures k at i. Specifically, $\mathsf{GenProof}'$ and $\mathsf{VerProof}'$ are constructed as follows:
  - After sampling k, compute the punctured key $k^{(i)} = \mathsf{Puncture}(k, i)$.
  - The program $\mathsf{GenProof}'$ is now computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{GenProof}[k^{(i)}, k_{\mathsf{sel}}, z])$ and $\mathsf{VerProof}'$ is computed as $\mathsf{Obf}(1^{\lambda_{\mathsf{obf}}}, 1^s, \mathsf{VerProof}[k^{(i)}, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star])$, both using the punctured key $k^{(i)}$ instead of k.

- $H^{(i)}_{2,3}(\lambda)$: Same as $H^{(i)}_{2,2}(\lambda)$ except we sample $r^\star \leftarrow \{0,1\}^{\ell(\lambda_{\mathsf{owf}})}$.

- $H^{(i)}_{2,4}(\lambda)$: Same as $H^{(i)}_{2,3}(\lambda)$ except we set $r^\star = \mathsf{Eval}(k, i)$.

In the subsequent subclaims, we relate the success probability between each pair of sub-hybrids.

Subclaim A.9. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(0)}_{2,0}(\lambda) = 1] \geq \Pr[H_2(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This proof follows identically to the proof of Claim 4.7 in [WW25], so we omit the details. At a high level, the subclaim follows by appealing to the indistinguishability security of Obf since the programs $\mathsf{VerProof}[k]$ and $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, 0]$ are functionally equivalent. □

Subclaim A.10. *For all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{2,1}(\lambda) = 1] \geq \Pr[H^{(i)}_{2,0}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This proof follows from Claim 4.8 in [WW25]. At a high level, the subclaim follows from the indistinguishability security of Obf since the programs $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i]$ and $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star]$ are functionally equivalent. Specifically, they are equivalent since when $x = i$ and $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z$, it holds that $\pi \oplus r^\star = x^\star$ iff $\mathsf{Eval}(k, x) = \pi$ as $r^\star = x^\star \oplus \mathsf{Eval}(k, i)$ by construction. □

Subclaim A.11. *For all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{2,2}(\lambda) = 1] \geq \Pr[H^{(i)}_{2,1}(\lambda) = 1] - 2 \cdot 2^{-(2\lambda+n)}.$$

Proof. This proof follows from Claim 4.9 in [WW25]. At a high level, the subclaim again follows from the indistinguishability security of Obf. Specifically, we note that $\mathsf{GenProof}[k, k_{\mathsf{sel}}, z]$ and $\mathsf{GenProof}[k^{(i)}, k_{\mathsf{sel}}, z]$ are functionally equivalent by the punctured correctness of $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Puncture})$ and neither program ever queries $\mathsf{Eval}(k, i)$. Similarly, the programs $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star]$ and $\mathsf{VerProof}[k^{(i)}, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star]$ are functionally equivalent. We lose an additive $2^{-(2\lambda+n)}$ term for each program, giving the bound in the subclaim above. □

Subclaim A.12. *For all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{2,3}(\lambda) = 1] \geq \Pr[H^{(i)}_{2,2}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. This proof follows from Claim 4.10 of [WW25]. At a high level, the only difference between hybrids $H^{(i)}_{2,3}$ and $H^{(i)}_{2,2}$ is that $r^\star$ is uniformly sampled as $r^\star \leftarrow \{0,1\}^{\ell(\lambda_{\mathsf{owf}})}$ in the former and set to be $x^\star \oplus \mathsf{Eval}(k, i)$ in the latter. So, since neither $\mathsf{GenProof}'$ nor $\mathsf{VerProof}'$ need to query $\mathsf{Eval}(k, i)$, indistinguishability of the hybrids follows from puncturable PRF security since $\mathsf{Eval}(k, i)$ is indistinguishable from random even given the punctured key $k^{(i)}$. □

Subclaim A.13. *For all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(i)}_{2,4}(\lambda) = 1] \geq \Pr[H^{(i)}_{2,3}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. Similar to Subclaim A.12 and Claim 4.11 of [WW25], this follows from puncturable PRF security. □

Subclaim A.14. *For all $i \in \{0,1\}^n$ and sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H^{(0)}_{2,i+1}(\lambda) = 1] \geq \Pr[H^{(i)}_{2,4}(\lambda) = 1] - 2 \cdot 2^{-(2\lambda+n)}.$$

Proof. This proof follows from Claim 4.12 of [WW25]. At a high level, this follows from two applications of indistinguishability security of Obf. First, by reverting $\mathsf{GenProof}'$ to using the non-punctured key k instead of $k^{(i)}$. These programs are functionally equivalent by punctured correctness of the PRF. Second, $\mathsf{VerProof}[k^{(i)}, k_{\mathsf{sel}}, z, y^\star, i, x^\star, r^\star]$ and $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, i+1]$ are functionally equivalent by punctured correctness and the fact that $(\mathsf{GenChal}, \mathsf{VerChal})$ is perfectly injective. Injectivity ensures that $\pi \oplus \mathsf{Eval}(k, i) = x^\star$ iff $\mathsf{VerChal}(y^\star, \pi \oplus \mathsf{Eval}(k, x))$. □

Subclaim A.15. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H_3(\lambda) = 1] \geq \Pr[H_{2,2^n,0}(\lambda) = 1] - 2^{-(2\lambda+n)}.$$

Proof. Similar to Subclaim A.9 and Claim 4.13 of [WW25], this follows from an application of indistinguishability security of Obf since $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star]$ and $\mathsf{VerProof}[k, k_{\mathsf{sel}}, z, y^\star, 2^n, x^\star, r^\star]$ are functionally equivalent given $x < 2^n$ for all $x \in \{0,1\}^n$. □

Combining the subclaims above consisting of $5 \cdot 2^n + 2$ pairs of adjacent hybirds of distance at most $2 \cdot 2^{-(2\lambda+n)}$, it holds for sufficiently large $\lambda \in \mathbb{N}$ that

$$\Pr[H_3(\lambda) = 1] \geq \Pr[H_2(\lambda) = 1] - (5 \cdot 2^n + 2) \cdot 2 \cdot 2^{-(2\lambda+n)} \geq \Pr[H_2(\lambda) = 1] - 12 \cdot 2^{-2\lambda},$$

as required for Claim A.8. □

By Claims A.2, A.7, and A.8, we get the following lower bound on the probability that the experiement $H_3(\lambda)$ outputs 1 for sufficiently large $\lambda \in \mathbb{N}$:

$$\Pr[H_3(\lambda) = 1] \geq 2^{-\lambda} \cdot \Pr[H_0(\lambda) = 1] - 15 \cdot 2^{-2\lambda}.$$

However, we also initially assumed that for infinitely many $\lambda \in \mathbb{N}$ that $\Pr[H_0(\lambda) = 1] > 1/q(\lambda)$ for some polynomial q. As $15 \cdot 2^{-2\lambda} < 2^{-\lambda}$ for large enough λ, we conclude for infinitely many $\lambda \in \mathbb{N}$ that

$$\Pr[H_3(\lambda) = 1] > 2^{-\lambda}/(q(\lambda) + 1). \tag{A.1}$$

In the following claim, we give an upper bound on the probability that $H_3(\lambda)$ outputs 1 for sufficiently large λ, leading to a contradiction.

Claim A.16. *For sufficiently large $\lambda \in \mathbb{N}$, it holds that*

$$\Pr[H_3(\lambda) = 1] \leq 2^{-2\lambda}.$$

Proof. This proof closely follows the proof of Lemma 4.14 in [WW25]. At a high level, it follows from the c_{owf}-subexponential security of the injective OWF with an inefficient sampler $(\mathsf{GenChal}, \mathsf{VerChal})$. More specifically, a reduction can invert the challenge $y^\star$ by sampling crs according to $H_3(\lambda)$, computing $(x, \pi) \leftarrow A(\mathsf{crs})$, and outputting $x^\star = \pi \oplus \mathsf{Eval}(k, x)$. The reduction succeeds whenever $H_3(\lambda)$ outputs 1 since this implies that $\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1$ and $\mathsf{Eval}(k_{\mathsf{sel}}, x) = z$, so $\mathsf{VerChal}(y^\star, \pi \oplus \mathsf{Eval}(k, x))$, as required. □

As $2^{-\lambda} \leq 1/(q(\lambda) + 1)$ for any polynomial q and sufficiently large $\lambda \in \mathbb{N}$, Claim A.16 is in contradiction with Equation (A.1). This establishes that our construction $\Pi = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ is adaptively sound, as required for Lemma 4.11. □

B Helpful Lemma for Theorem 3.1

In the following lemma, we give a simple bound on the function $f(\delta, t) = \delta \cdot (1-\delta)^t$ used in the proof of Theorem 3.1.

Lemma B.1. *For any $\delta \in [0,1]$ and $t > 0$, it holds that*

$$f(\delta, t) = \delta \cdot (1-\delta)^t < \frac{1}{e \cdot t}.$$

Proof. Note that when $\delta = 0$ or 1, $f(\delta, t) = 0$ for all $t > 0$. Furthermore, for any $\delta \in (0,1)$, it hods that $(1-\delta) \in (0,1)$, so $f(\delta, t) > 0$ is strictly positive. Taking the derivative of $f(\delta, t)$ with respect to δ, we see that f is maximized when

$$\frac{\partial}{\partial \delta} f(\delta, t) = (1-\delta)^{t-1} \cdot (1 - \delta \cdot (t+1)) = 0,$$

meaning $\delta = 1/(t+1)$. Plugging this back into f, it follows that

$$f(\delta, t) \leq \frac{1}{t+1} \cdot \left(1 - \frac{1}{t+1}\right)^t < \frac{1}{t+1} \cdot \frac{t+1}{t} \cdot \frac{1}{e} = \frac{1}{e \cdot t},$$

as required. □