Witness Semantic Security

Paul Lou[*] Nathan Manohar[†] Amit Sahai[‡]

Abstract

To date, the strongest notions of security achievable for two-round *publicly-verifiable* cryptographic proofs for NP are witness indistinguishability (Dwork-Naor 2000, Groth-Ostrovsky-Sahai 2006), witness hiding (Bitansky-Khurana-Paneth 2019, Kuykendall-Zhandry 2020), and super-polynomial simulation (Pass 2003, Khurana-Sahai 2017). On the other hand, zero-knowledge and even weak zero-knowledge (Dwork-Naor-Reingold-Stockmeyer 1999) are impossible in the two-round publicly-verifiable setting (Goldreich-Oren 1994). This leaves an enormous gap in our theoretical understanding of known achievable security and the impossibility results for two-round publicly-verifiable cryptographic proofs for NP.

Towards filling this gap, we propose a new and natural notion of security, called *witness semantic security*, that captures the natural and strong notion that an adversary should not be able to learn any partial information about the prover's witness beyond what it could learn given only the statement x. Not only does our notion of witness semantic security subsume both witness indistinguishability and witness hiding, but it also has an easily appreciable interpretation.

Moreover, we show that assuming the subexponential hardness of LWE, there exists a two-round public-coin publicly-verifiable witness semantic secure argument. To our knowledge, this is the strongest form of security known for this setting.

As a key application of our work, we show that non-interactive zero-knowledge (NIZK) arguments in the common reference string (CRS) model can additionally maintain witness semantic security even when the CRS is maliciously generated. Our work gives the first construction from (subexponential) standard assumptions that achieves a notion stronger than witness-indistinguishability against a malicious CRS authority.

In order to achieve our results, we give the first construction of a ZAP from subexponential LWE that is adaptively sound. Additionally, we propose a notion of simulation using non-uniform advice about a malicious CRS, which we also believe will be of independent interest.

[*] UCLA. Email: pslou@cs.ucla.edu.

[†] IBM T.J. Watson Research Center. Email: nmanohar@ibm.com.

[‡] UCLA. Email: sahai@cs.ucla.edu.

1 Introduction

Cryptographic proofs for languages in NP, first studied by Goldwasser et. al [GMR85], are fundamental and powerful primitives. The strongest security guarantee, zero-knowledge (ZK), allows a prover to convince a verifier that a statement is true without revealing anything beyond its validity. This seemingly magical capability, however, has a price. In the plain model, a ZK protocol for any language outside of BPP requires at least three rounds of communication between the prover and the verifier as shown by Goldreich and Oren [GO94]. In fact, as observed by Bitansky et al. [BKP19], this impossibility result rules out even publicly-verifiable, weak ZK [DNRS99] in two rounds.

This barrier is rather unfortunate as two-round public-coin publicly-verifiable protocols are enormously useful because publicly-verifiable, or better yet public-coin, protocols allow for a single back-and-forth between parties, and typically allow the first round message to be re-used[1].

Our collective understanding of what qualitative level of security is achievable in the two-round publicly-verifiable setting for NP is quite limited. To our knowledge, witness indistinguishability (WI) [DN00, GOS06], witness hiding (WH) [FS90, BKP19, KZ20], and super-polynomial simulation (SPS) [KS17, Pas03] are the only known achievable notions in this setting[2]. Worse yet, with all these notions, it's not clear in general how to intuitively interpret a limit on the information leaked by the proof. This leaves an enormous gap in our qualitative understanding of what level of security is possible for this setting.

What do we want from a two-round publicly-verifiable cryptographic proof? Suppose that Alice is an auditor, and a prover would like to send to Alice an encrypted signed financial document and a proof that attests the underlying signed financial document would pass an audit. What level of security is sufficient to preserve the privacy of this financial document? If the encryption scheme has perfect decryption, then there is a unique witness given by a unique message and unique encryption randomness, so WI gives no meaningful security at all as there is a unique witness. Witness hiding only prevents Alice from recovering the entire signed document and randomness of the prover; WH does not prevent Alice from extracting out critical partial information, such as a bank account number or a partial transaction history. This state of affairs begs the question,

Can we achieve a qualitatively stronger security notion than witness indistinguishability and witness hiding for two-round publicly-verifiable proofs?

Our contribution: Witness Semantic Security. In this work, we propose a naturally motivated notion of security, *witness semantic security* (WSS). Informally speaking, witness semantic security guarantees that any partial information about the witness[3] that an adversary can learn from seeing the proof, the adversary could have also learned from seeing only the statement being proven! This definition subsumes both WI and WH. Furthermore, *witness semantic security gives a qualitative security guarantee that is easy to understand*, and nevertheless, we show how to construct a two-round public-coin publicly-verifiable protocol achieving witness semantic security from the subexponential hardness of learning with errors. Our construction effectively addresses a significant gap in our understanding of what qualitative security guarantees are theoretically possible in the two-round publicly-verifiable setting.

1.1 Application: Malicious-CRS Security for Non-Interactive Zero-Knowledge

As a key application of our ideas, we also construct a non-interactive zero-knowledge (NIZK) argument for NP in the CRS model that additionally achieves a new simulation-based definition of security even when the CRS is maliciously generated. We show that any NIZK protocol in the CRS model that satisfies this notion, which we term malicious CRS non-uniform zero-knowledge (malicious CRS NUZK), immediately translates into a plain model two-round publicly-verifiable protocol achieving witness semantic security. Our

[1] All our protocols will have this reusability property!

[2] A work of Khurana [Khu21] achieves a different kind of cryptographic proof, but one that does not allow the prover to prove a fixed NP statement of its choosing. This is done via a new security notion termed non-interactive distributional indistinguishability (NIDI). In NIDI protocols, the statement is sampled from a distribution simultaneously with the proof.

[3] This partial information is modeled as a predicate (or indeed more generally, a function) of the witness, exactly as partial information is modeled in the definition of semantic security for encryption schemes [GM84].

construction of this NIZK argument, in of itself, advances our collective knowledge of the qualitative security guarantees attainable in what has been called the subverted CRS setting [BFS16]. We now motivate this setting and explain the significance of this contribution.

Non-interactive Zero-knowledge. Non-interactive zero-knowledge (NIZK) protocols involve only a single message to the verifier. It is well-known that non-interactive zero-knowledge protocols for all of NP cannot be constructed in the plain model [BFM88]. However, they can be constructed in the common reference/random string (CRS) model, where a trusted party (the CRS authority) publishes a (possibly uniformly random) public string that is used during the protocol execution. While the CRS authority is modeled as a completely trusted entity, in the real world, unconditionally trusted parties do not exist. Indeed, misplaced trust in an entity has led to real world consequences: It is widely believed that the Dual EC deterministic random bit generators had intentional backdoors built in by the NSA [CFN^{+}14]. Returning our attention to NIZKs, the standard definition of NIZKs does not — indeed, it cannot! — guarantee zero-knowledge when a CRS authority behaves maliciously. In fact, the malicious CRS authority may be capable of recovering the prover's entire witness! This naturally leads to the following fundamental question, which is also *highly motivated* by recent and ongoing examples of central authority misbehavior and subversion by authorities and governments.

What privacy guarantees for the prover's witness can be achieved for NIZKs in the CRS model if the CRS is generated maliciously?

There have been several previous approaches to tackling this issue, which we summarize next. However, as we describe below, these previous works have all suffered from various drawbacks. In our work, we will address these drawbacks through our concept of witness semantic security.

Prior Work: Subversion in the CRS Model. The study of NIZKs with a malicious CRS was initiated by Bellare et al. [BFS16], who study various notions of security for NIZKs under a subverted CRS, showing both impossibility and feasibility results. For example, they show that a natural extension of soundness to the subverted CRS setting (subversion soundness) is incompatible with zero-knowledge even when zero knowledge is only required with an honestly generated CRS. Since we demand zero-knowledge to hold under an honestly generated CRS, we therefore cannot hope to achieve subversion soundness; thus our focus will be to achieve the strongest form of security for the witness possible under a malicious CRS.

Bellare et al. [BFS16] also propose a notion of zero-knowledge against a maliciously generated CRS, which they term subversion zero-knowledge. However, this definition of subversion zero-knowledge has two drawbacks: Firstly, it inherently only addresses security with respect to uniform adversaries and is (trivially) unachievable against non-uniform adversaries, which are the standard model for adversaries in cryptography. Moreover, all known constructions of NIZKs achieving the notion of subversion zero-knowledge rely on knowledge-extraction assumptions that are neither falsifiable nor standard assumptions.

In particular, [BFS16] show that under the Diffie-Hellman knowledge-of-exponent assumption (DH-KEA), there exists a NIZK protocol that also satisfies subversion zero-knowledge. In general, these knowledge extraction assumptions require the existence of a type of extractor that is related to the notion of extractable one-way functions whose existence is incompatible with the existence of indistinguishability obfuscation [BCPR14], a primitive we now have from well-founded assumptions [JLS21]. While the DH-KEA assumption is not known to be false as stated in [BFS16], in this work, we show that it is false in a world with uniform auxiliary input. Indeed, we show that if there were a world where some external party instantiates a selectively secure one-time universal sampler [HJK^{+}16], then the DH-KEA assumption is false. Such selectively secure one-time universal samplers exist assuming indistinguishability obfuscation and one-way functions. The DH-KEA assumption is also quantum broken. Other constructions of proof systems with subversion ZK [Fuc18, ABLZ17, Bag19] rely either on similar knowledge-extraction assumptions or even stronger Generic/Algebraic Group Models.

To the best of our knowledge, without using a knowledge-of-exponent assumption, the strongest form of subversion zero-knowledge that we know how to obtain for a NIZK in the standard CRS model is subversion witness indistinguishability [BFS16]. Unfortunately, witness indistinguishability (WI) provides no security guarantees when a statement has a unique witness (resp. structured witness) and could reveal the entire witness (resp. structured part of the witness[4]).

Prior Work: Accountability in the CRS Model. Recently, Ananth et al. [AADG21] began a line of work on notions of accountability in the CRS model. A malicious authority may use a maliciously generated CRS to create proofs of false statements that pass the verifier's validity check. Indeed, for any NIZK system, such power to prove false statements is inevitable (as argued by [BFS16]). However, a malicious authority can be "caught red-handed" if it is convinced/bribed to produce a proof for a statement that is known to be false (e.g. a statement in coNP with a witness of falsity). Our systems will retain this basic form of accountability for authorities that misbehave to prove false statements. We will not address accountability for soundness further, however, as our focus is on privacy.

Ananth et al. propose a method to hold the CRS authority accountable in such instances. Informally, they construct a NIZK proof from SXDH that satisfies a notion of accountability: if a malicious CRS authority sells an *entire witness* from a proof to a buyer who then turns this over to an investigator, then there is an extractor that can produce a piece of evidence τ that convinces an algorithm Judge that the CRS authority has misbehaved.

Unfortunately, the construction of [AADG21] comes with two drawbacks: (1) Accountability against a malicious CRS authority is only possible if the CRS authority can learn the prover's *entire* witness. In particular, their construction provides no guarantees against a malicious CRS authority that only learns partial information about the prover's witness (see the beginning of the introduction, and below, for an example). (2) Their construction relies on the quantum-broken assumption of SXDH.

Our Contributions to Malicious CRS NIZKs. As our main application of witness semantic security, we achieve two crucial objectives simultaneously:

- Security from a **standard assumption**. Ideally, this assumption should also be post-quantum secure.
- A security notion that **hides partial information** about the prover's witness.

The Importance of Hiding Partial Information about Witnesses. We have already discussed why it is crucial to hide partial information earlier in this introduction. As another example, consider a scenario in which a naïve prover has a document digitally signed by a credit score company that contains not only the prover's credit score but also other personal data (e.g. date of birth, home address, social security number, etc.). The prover can use this document as a witness for the statement, "The prover's credit score is greater than 750." If the prover produces a NIZK argument/proof π using a maliciously generated CRS and sends π to the verifier—a crooked loanshark—then the verifier can ask the malicious CRS authority to extract only part of the witness, such as the prover's social security number, from the proof π. Such a situation is clearly undesirable for the prover, yet the notion of accountability proposed in [AADG21] does not protect against this. Observe that this part of the witness (the prover's social security number) is otherwise hard to predict from only knowing whether the prover's credit score is greater than 750. Therefore, our notion of witness semantic security will provide strong protection in this setting!

1.2 Our Results

In this work, we introduce a new notion that we call *witness semantic security* for cryptographic proofs.

Witness Semantic Security. Witness semantic security generalizes witness hiding to hiding even partial information about the witness. We use the terminology "semantic security" in the original sense proposed by Goldwasser and Micali [GM82] for probabilistic encryption (not to be confused with the notion of ciphertext indistinguishability [GM84], which was shown to be equivalent in the context of encryption). That is, our definitions capture the guarantee that any hard-to-compute function of the witness, when given only the statement, remains hard to compute when also given a proof of the statement. Witness hiding, first defined

[4] For example, suppose for a particular instance x, all witnesses are of the form $w \circ w'$, for a fixed string w, but where w' can be any string. Then a WI protocol can reveal all of w.

in the seminal work of Feige and Shamir [FS90], can be seen as a (much) weaker version of our definition, which only guarantees that the entire witness cannot be extracted from the proof. Yet, to our knowledge, surprisingly this natural extension of hiding partial information has not been explored other than through the standard definition of zero-knowledge or weak zero-knowledge. However, as we show in this work, it is possible to achieve witness semantic security by a two-round publicly-verifiable plain model argument system and by NIZK arguments in the CRS model where witness semantic security additionally holds even with a malicious CRS.

For the definition of witness semantic security to be meaningful, it is defined with respect to some distribution over instances, witnesses, auxiliary information, and a family of deterministic functions. This is necessary because there are certainly easy-to-learn deterministic functions f of a witness w for which the adversary has non-negligible advantage of obtaining $f(w)$ when given only x and f (for example, the constant function f that is always 0), so witness semantic security is defined with respect to sensible distributions where $f(w)$ is hard to learn given only x and f.

Informally, witness semantic security states that (with respect to a suitable distribution), an adversary $\mathcal{A}$, on input a statement x, a function f, auxiliary input Aux, and a proof π generated using a witness w, is unable to output $f(w)$ with non-negligible advantage.

Extending Witness Semantic Security. Our definition above is most meaningful in the case where, for each input x, there is a unique witness for x. This setting immediately separates the notion of WSS from WI. It is also meaningful if all valid witnesses for x share a very long common substring – this is indeed very common in cryptography, where for example the witness includes the entire plaintext for some ciphertext that is part of the statement x. Nevertheless, a natural question is whether witness semantic security can be meaningfully extended to setting of multiple witnesses. We define (and achieve) what we believe to be important progress towards this, for functions f with long verifiable outputs.

More precisely, we define an extension of witness semantic security where we additionally require that there exist a polynomial-time verification algorithm V_f, associated with the function f, such that $V_f(x, y) = 1$ if and only if there exists some witness w' for x such that $y = f(w')$. We then require, roughly speaking, that all such verifiable functions that are hard to predict given only x, are still hard to predict when given the proof as well. We term this extension of witness semantic security as verifiable witness semantic security (VWSS).

We observe that verifiable witness semantic security implies witness hiding for NP languages. This follows by setting f as the identity function and letting V_f be the NP verification algorithm. More broadly, verifiable witness semantic security generalizes witness hiding to hiding any verifiable function of any valid witness. In fact, the notion of VWSS is easily separated from WH by considering the language $L_{\mathsf{AND}} = \{(x, x') : x, x' \in \mathsf{SAT}\}$ and the function f that on witnesses of the form (w, w') outputs w, the witness for x. Since SAT is in NP, there exists an NP verifier V_f that serves as the verification function in the definition of VWSS. Observe that a proof system that produces proofs (w, π'), where π' is a WH proof for x', is WH for L_{AND}, yet this proof system is not VWSS since one can trivially recover $w = f((w, w'))$.

Whether our notion of verifiable witness semantic security can be meaningfully strengthened even further is an important open question. We give some evidence that this is a nontrivial question in Section 6.

Reusable Witness Semantic Security. The notion of witness semantic security only deals with hiding partial information about a single witness w used to generate a proof π for a single statement x. We additionally define a strengthening of witness semantic security, called *reusable* witness semantic security, that captures the notion that an adversary should be unable to recover partial information $f(w_1, \ldots, w_n)$ about many witnesses $\{w_i\}$ used to generate proofs $\{\pi_i\}$ for statements $\{x_i\}$ even against an adversary that is allowed to adaptively query statements x_i and receive proofs π_i. Our constructions achieve this stronger definition, without needing to make any additional assumptions beyond subexponential security of LWE.

Achieving Witness Semantic Security. Assuming subexponentially secure LWE, we construct both (1) two-round public-coin publicly verifiable (reusable) WSS and VWSS cryptographic proofs, and (2) NIZKs that additionally satisfies (reusable) WSS and VWSS even in the presence of a malicious CRS.

Witness Hiding Trivially Implies Accountability. We observe that the accountability and defamation-free notions in [AADG21] are trivially satisfiable if it is possible to prevent the malicious CRS authority from ever extracting a witness from a proof, regardless of whether or not the CRS was honestly generated (e.g. if the NIZK is witness-hiding [FS90] even in the presence of a maliciously generated CRS). In such a situation, the algorithm Judge would simply never accept any evidence τ, and the accountability notion is vacuous since a malicious CRS authority is never capable of obtaining the witness from a proof. We observe that witness-hiding [FS90] is a stronger form of accountability since instead of proving to a judge that the CRS authority was capable of learning a prover's witness, we simply prevent the CRS authority from learning the prover's witness in the first place! Our notion of verifiable witness semantic security implies witness hiding, and, therefore, our NIZK construction achieves a much stronger security guarantee than accountability if the CRS authority is malicious.

Additional Contributions. In order to construct our NIZK from subexponential LWE satisfying witness semantic security with a malicious CRS, we construct a ZAP from subexponential LWE that is adaptively sound. Previous constructions of ZAPs from LWE [BFJ$^+$20] were not adaptively sound. Along the way to our construction, we construct what we call *super-dense public-key encryption* from LWE, where super-dense refers to the property that *every* possible public key string has a working decryption key. We believe this tool may prove to be useful in other contexts.

Additionally, we show that the Diffie-Hellman knowledge-of-exponent assumption [BFS16] is false in a world with uniform auxiliary input, giving further evidence that knowledge-of-exponent assumptions are significantly shakier than standard assumptions.

Finally, we propose the notion of simulation using *non-uniform advice about a malicious CRS* – which we also believe will be of independent interest.

1.3 Other Related Works

Besides the aforementioned results about the round complexity of the interactive cryptographic proofs and the results for the single CRS model, there are other models which address the issue of trusting a CRS authority such as the multi-string model [GO07, BCG$^+$15, BGG18, BJMS20] and the updatable CRS model [GKM$^+$18].

1. **The multi-string model**: This model generates a single CRS from multiple CRS's generated by multiple parties with the guarantee that the overall CRS is secure as long as at least one party in the fixed set of participants behaves honestly. Thus, trust is distributed across multiple parties instead of relying on a single trusted authority. This trust model, however, raises the issue of how this fixed set of participants is chosen.

2. **The updatable CRS model**: This model addresses the above concern of a fixed set of parties by allowing the CRS to be updated by an arbitrary party. Once an honest party updates the CRS, soundness is preserved for subsequent computations based on the updated CRS even after subsequent (possibly malicious) updates. This feature allows an honest verifier to update the CRS, thereby circumventing the impossibility result of [BFS16]. We emphasize that all known constructions in this line of work use either knowledge extraction assumptions [GKM$^+$18, MBKM19] or the use of idealized models like the Random Oracle Model or Algebraic Group Model to prove security [DRZ20, CHM$^+$20].

Concurrent Work. The concurrent and independent work of [AAG$^+$24] introduce a notion of zero-knowledge called subversion advice-ZK, which is nearly identical to the notion of zero-knowledge, malicious CRS non-uniform zero-knowledge (Def. 4.12), introduced in this work. Moreover, their construction of a NIZK protocol that additionally achieves subversion advice-ZK follows the same template as our construction of a NIZK protocol that achieves malicious CRS non-uniform zero-knowledge. They additionally introduce a notion of accountable soundness, in the same style of the accountability definition introduced in [AADG21], that we do not consider in this work. This notion addresses the impossibility result of [BFS16] that states subversion soundness and honest CRS zero-knowledge cannot be simultaneously achieved.

[AAG$^+$24] similarly point out that subversion advice-ZK implies that hard-to-compute deterministic predicates and functions of the witness are not leaked by the proof, a notion they call subversion function hiding. This notion is introduced in our work as *witness semantic security* and is explored more fully in our paper and is the central focus of this work. They state in their paper: "While an extension to hide functions of all witnesses is interesting, it is not clear how such a definition should look like. We leave it as an open question for the future." We address this question by introducing several natural variants of witness semantic security, namely verifiable witness semantic security, and show lower bounds on natural extensions of witness semantic security, i.e. a version of witness semantic security which even distributional zero-knowledge does not imply.

1.4 Organizational Overview

We introduce our primary security notions—witness semantic security, malicious CRS witness semantic security, verifiable witness semantic security, and malicious CRS verifiable witness semantic security—in Section 4. These definitions are all shown to be implied by simulation-based definition also found in Section 4. An extended definition capturing reusability, or concurrent and sequential composition, for witness semantic security is found in Section 4.5. Our NIZK construction from subexponential LWE that satisfies both malicious CRS witness semantic security and malicious CRS verifiable witness semantic security is found in Section 5. Finally, we explore fundamental barriers to strengthening our definitions of witness semantic security in Section 6.

2 Technical Overview

2.1 Defining Witness Semantic Security

We desire a notion of witness hiding that stipulates that an adversary cannot learn any hard-to-learn function of the witness used by the honest prover. Intuitively, this notion captures the idea that for some instance-witness pair $(x, w) \in R_L$ for an NP relation R_L, if it is hard to learn $f(w)$ for some function f given only x, then it should also be hard to learn $f(w)$ when interacting with the honest prover. In the malicious CRS setting, it should be that for any adversary who maliciously chooses a CRS, if it is hard to learn $f(w)$ for some function f given only x, then it should also be hard to learn $f(w)$ given a proof π that $x \in L$ with respect to a maliciously chosen CRS. This leads to the following two definitions.

Definition 2.1 (Witness Semantic Security). *An interactive argument system $(\mathcal{P}, \mathcal{V})$ for a language $L \in \mathsf{NP}$ is witness semantic secure if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x, w, \mathsf{Aux}, f, y) \mid y = f(w), (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$, where $\mathscr{F}$ is a set of deterministic functions, for all polynomial sized $\mathcal{A}_1$ and polynomial sized $\mathcal{A}_2$ which additionally takes as input a state τ generated by $\mathcal{A}_1$, there exists a polynomial sized $\mathcal{B}$ and there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$*

$$\Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)} \left[\mathcal{A}_2\left(1^\lambda, \tau, \langle \mathcal{P}(1^\lambda, x, w), \mathcal{A}_1(1^\lambda)\rangle, x, \mathsf{Aux}, f\right) = y \right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)} \left[\mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) = y\right] + \mu(\lambda).$$

Remark 2.2. In this work, all interactive protocols we consider are two rounds. Note that in the definition above, the verifier does not see any outputs of D when generating its first message. The typical situation we consider is one where the verifier's first message is sent much earlier than the generation of any statement or proof. Our construction satisfies the property that the verifier's first message can be reused across multiple proofs.

Definition 2.3 (Malicious CRS Witness Semantic Security). *A non-interactive argument system $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ for a language $L \in \mathsf{NP}$ is malicious CRS witness semantic security if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x, w, \mathsf{Aux}, f, y) \mid y = f(w), (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions, for all unbounded $\mathcal{A}_1$ and polynomial-sized $\mathcal{A}_2$, there exists a polynomial sized $\mathcal{B}$ and a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$*

$$\Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\begin{array}{c}(\mathsf{CRS}^*,\tau)\leftarrow\mathcal{A}_1(1^\lambda)\\ \pi\leftarrow\mathsf{Prove}(\mathsf{CRS}^*,x,w)\\ y\leftarrow\mathcal{A}_2(1^\lambda,\tau,x,\pi,\mathsf{Aux},f)\end{array}\right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\mathcal{B}(1^\lambda,x,\mathsf{Aux},f)=y\right]+\mu(\lambda).$$

At this point, we make two quick observations. The first is that WSS implies WI and the second is that any NIZK argument for $L\in\mathsf{NP}$ that satisfies malicious CRS WSS can be immediately converted to a two-round publicly-verifiable WSS argument in the plain model, allowing us to focus only on constructing a NIZK satisfying malicious CRS WSS.

2.1.1 Verifiable Witness Semantic Security

The above definition of witness semantic security only addresses preventing an adversary from learning $f(w)$, where w is the witness used in constructing a proof π. It, however, does not provide any guarantee about the adversary learning $f(w')$ for a different witness w' that was not used in generating π. As discussed previously, we can define an extension of witness semantic security, called verifiable witness semantic security, that states that the adversary should not be able to learn $f(w')$ for any valid witness w' if it is possible to efficiently verify that an output y is indeed $f(w')$ for some w'.

Definition 2.4 (Verifiable Witness Semantic Security)**.** *An interactive argument system $(\mathcal{P},\mathcal{V})$ for a language $L\in\mathsf{NP}$ is verifiable witness semantic secure (VWSS) if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x,w,\mathsf{Aux},f)\mid (x,w)\in R_L,\mathsf{Aux}\in\{0,1\}^*,f\in\mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions such that for all $f\in\mathscr{F}$, there exists a polynomial-time verification algorithm V_f such that $V_f(x,y)=1$ if and only if $y=f(w')$ for some w' such that $(x,w')\in R_L$, for all polynomial sized $\mathcal{A}_1$ and polynomial sized $\mathcal{A}_2$ which additionally takes as input a state τ generated by $\mathcal{A}_1$, there exists a polynomial sized $\mathcal{B}$ and there exists a negligible function $\mu(\cdot)$ such that*

$$\Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[\begin{array}{c}y\leftarrow\mathcal{A}_2\left(1^\lambda,\tau,\langle\mathcal{P}(1^\lambda,x,w),\mathcal{A}_1(1^\lambda)\rangle,x,\mathsf{Aux},f\right)\\ s.t.\ \exists w',y=f(w')\wedge(x,w')\in R_L\end{array}\right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[y\leftarrow\mathcal{B}(1^\lambda,x,\mathsf{Aux},f):\exists w',y=f(w')\wedge(x,w')\in R_L\right]+\mu(\lambda),$$

where WLOG Aux contains a description of V_f.

Definition 2.5 (Malicious CRS Verifiable Witness Semantic Security)**.** *A non-interactive argument system $(\mathsf{GenCRS},\mathsf{Prove},\mathsf{Verify})$ for a language $L\in\mathsf{NP}$ is malicious CRS verifiable witness semantic security if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x,w,\mathsf{Aux},f)\mid (x,w)\in R_L,\mathsf{Aux}\in\{0,1\}^*,f\in\mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions such that for all $f\in\mathscr{F}$, there exists a polynomial-time verification algorithm V_f such that $V_f(x,y)=1$ if and only if $y=f(w')$ for some w' such that $(x,w')\in R_L$, if for all unbounded $\mathcal{A}_1$ and polynomial-sized $\mathcal{A}_2$, there exists a polynomial-sized $\mathcal{B}$ and a negligible function $\mu(\cdot)$ such that*

$$\Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[\begin{array}{c}(\mathsf{CRS}^*,\tau)\leftarrow\mathcal{A}_1(1^\lambda)\\ \pi\leftarrow\mathsf{Prove}(\mathsf{CRS}^*,x,w)\\ y\leftarrow\mathcal{A}_2(1^\lambda,\tau,x,\pi,\mathsf{Aux},f)\\ y=f(w')\wedge(x,w')\in R_L\end{array}\right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[y\leftarrow\mathcal{B}(1^\lambda,x,\mathsf{Aux},f):\exists w',y=f(w')\wedge(x,w')\in R_L\right]+\mu(\lambda).$$

Without loss of generality, assume that V_f is given in the auxiliary input Aux.

We now make several observations regarding verifiable witness semantic security.

1. Any NIZK protocol satisfying malicious CRS VWSS immediately gives a two-round publicly-verifiable VWSS protocol.

2. Witness hiding is a special case of verifiable witness semantic security. This is seen by considering a singleton set $\mathscr{F}$ that contains only the identity function. Conceptually, this is because NP relations are efficiently verifiable, therefore an adversary cannot extract any witness for x unless it was already easy to do so from just the statement x alone. Thus, malicious CRS verifiable witness semantic security is a stronger security notion than malicious CRS witness hiding.

3. Malicious CRS witness hiding trivially implies accountability (Def. 3.33) and makes the defamation-free guarantee meaningless by rendering the Judge algorithm useless. Therefore, any protocol that satisfies malicious CRS verifiable witness semantic security trivially satisfies accountability in that a malicious CRS authority *cannot* extract a witness except with negligible probability. This is formally shown in Section 4.

4. Observe that WSS (Def. 2.1) is incomparable to VWSS (Def. 2.4). Definition 2.1 does not impose a condition on efficient verifiability and applies to functions whose output is only a single bit but only prevents the adversary from recovering the $f(w)$ for the *specific* witness w used in a proof π. On the other hand, Definition 2.4 captures a more general security property, but only for a smaller family of functions with long outputs for which the verifiability property holds and the function is hard-to-learn on any witness.

2.1.2 Barriers to Strengthening Verifiable Witness Semantic Security

One can consider a natural strengthening of Def. 2.4 where we remove the requirement of the existence of a verification algorithm V. In Section 6, we give a reasonable conjecture (one provable in the Random Oracle Model) and an interactive protocol that under the conjecture satisfies distributional zero-knowledge yet does not satisfy this strengthening of verifiable witness semantic security. We leave further investigation of this strengthening to future work.

2.1.3 Reusable Witness Semantic Security

We also define a strengthening of Def. 2.3 called malicious reusable CRS witness semantic security that captures the notion that an adversary should be unable to recover partial information $f(w_1, \ldots, w_n)$ about many witnesses $\{w_i\}$ used to generate proofs $\{\pi_i\}$ for statements $\{x_i\}$ even against an adversary that is allowed to adaptively query statements x_i and receive proofs π_i. This is formally defined in Section 4.5.

2.2 NIZK Satisfying Witness Semantic Security with a Malicious CRS

Our main result is the following informal theorem and its informal corllary:

Theorem 2.6 (Informal Main Result)**.** *Assuming the subexponential hardness of* LWE*, there exists a NIZK argument system that also satisfies malicious CRS witness semantic security, malicious CRS verifiable witness semantic security, and malicious reusable CRS witness semantic security..*

Corollary 2.7 (Informal Main Corollary)**.** *Assuming the subexponential hardness of* LWE*, there exists a two-round public-coin publicly-verifiable argument system that satisfies witness semantic security,and verifiable witness semantic security, where the first message is reusable.*

We note that our NIZK argument has a CRS generator that, when behaving honestly, generates *uniformly* random CRS's. To achieve such a construction, we introduce and build what we call super-dense public-key encryption.

Super-Dense PKE A critical building block towards building our NIZK arguments with the desired properties is a "super-dense" PKE scheme, which was not known from the hardness of LWE prior to this work[5]. Denseness refers to the public-key space and dense PKE schemes in the existing literature have the property that with an overwhelming probability, a randomly chosen public key has a decryption key for

[5] In fact, the work of Badrinarayan et al. [BFJ+20] writes that such a scheme is "unfortunately not known to exist based on LWE".

which correctness holds. This property is not good enough for our purposes, and we need "super dense" PKE schemes in which *every* possible string of the appropriate length has a corresponding decryption key. Our starting point is the dual Regev encryption scheme [GPV08].

Dual Regev Encryption Scheme [GPV08]:

Let $n = n(\lambda), m = m(\lambda), q = q(\lambda)$ be polynomials in λ where $m > 2n \log q$. Let χ denote the LWE error distribution.

- $\mathsf{Keygen}(1^\lambda)$: Sample uniform randomly $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$ and $\mathbf{s} \in \{-1, 0, 1\}^m$. Let $\mathsf{pk} = \mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \mathbf{s}^\top \mathbf{A} \end{bmatrix}$ and $\mathsf{sk} = (\mathbf{s}^\top, -1)$. Output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(1^\lambda, \mathsf{pk}, b \in \{0,1\})$: Parse pk as $\mathbf{A}'$. Sample a random error vector $\mathbf{e}'$ from χ^{m+1}. Sample a random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Let $\mathbf{b} \in \mathbb{Z}_q^{m+1}$ be a vector with $b \cdot \lfloor q/2 \rfloor$ in $(m+1)$th index and 0 elsewhere. Output $\mathsf{ct} = \mathbf{A}'\mathbf{u} + \mathbf{e} + \mathbf{b}$.
- $\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{ct})$: Parse sk as $\mathbf{s}'^\top = (\mathbf{s}^\top, -1) \in \mathbb{Z}_q^{m+1}$. Parse ct as $\mathbf{a}'' \in \mathbb{Z}_q^{m+1}$. Compute $r \leftarrow \mathbf{s}'^\top \cdot \mathbf{a}'' \in \mathbb{Z}_q$ and round r to closer of 0 and $\lfloor q/2 \rfloor$. If r rounds to 0, then output 0. Otherwise output 1.

Observe that the dual Regev encryption scheme is dense but not super dense. Ideally, the totality of the SIS problem, which guarantees a short solution for every matrix $\mathbf{A} \in \mathbb{Z}_q^{(m+1) \times n}$ for $m > 2n \log q$, gives us a corresponding decryption key given by this short solution. A quick observation is that any short solution with a zero in the last coordinate would fail to be a good decryption key because the decryption algorithm simply computes the inner product between the decryption key and the ciphertext, in which case the message bit would be multiplied by 0 and the message bit is lost. Unfortunately, there are many matrices $\mathbf{A}$ for which the only short solutions must have a 0 in the last entry, as evidenced by the all zeroes matrix whose last column's entries are all $\lfloor q/2 \rfloor$.

To prevent this correctness violation and achieve a super dense PKE scheme, we introduce a "super dense" dual Regev encryption scheme, in which the encryption algorithm outputs $n+1$ many dual Regev encryptions where the ith encryption of the bit b places the message bit in the ith entry instead of the last entry. By doing so, any non-zero short solution to any matrix $\mathbf{A}$ is a valid decryption key, thus achieving the super dense property. Moreover, the security remains due to the hardness of the decisional LWE problem with polynomially many samples. This super dense PKE scheme allows us to give the first construction of a ZAP from subexponential LWE with adaptive soundness. This ZAP will play a critical role in our construction of the NIZK as we now explain.

Adapting Pass' Two-Round Witness Hiding Protocol As a simpler goal, we will first construct a NIZK that satisfies witness hiding even with a malicious CRS. While it remains an open problem to construct non-interactive witness hiding in the plain model, there exist two-round witness hiding schemes. Such schemes involve the verifier first sending a message and then the prover responding to the verifier's message with a proof. Moreover, there exist constructions where the proof is publicly verifiable given the protocol transcript.

Since we are focused on non-interactive protocols, we must compress the verifier's message. In the CRS model for NIZKs, we can move the first round message of the two-round witness hiding protocols into the CRS. Since the two-round witness hiding protocol is publicly verifiable, it remains verifiable even if the verifier's first round message is instead viewed as the CRS. Our construction of the NIZK, therefore, is based on compressing the Pass construction [Pas03, KZ20] for two-round witness hiding in the plain model into the CRS model.

We review the two-round witness hiding Pass construction: Let x be a statement for language $L \in \mathsf{NP}$ and let w be a witness for x. Let R_L denote the NP relation for L. Let $f : \{0,1\}^n \to \{0,1\}^k$ be any surjective one-way function, let Com be a perfectly binding commitment scheme, and let Π_{NIWI} be a non-interactive witness indistinguishable argument system for the language of statements $\{S_{b,c_w,c_r}\}$ where S_{b,c_w,c_r} is defined over arbitrary strings b, c_w, c_r as

$$\begin{aligned} S_{b,c_w,c_r} \triangleq & (\exists w', R' \text{ such that } c_w = \mathsf{Com}(w'; R') \wedge ((x, w') \in R_L)) \\ & \vee (\exists r', R' \text{ such that } c_r = \mathsf{Com}(r'; R') \wedge (b = f(r'))) \end{aligned}$$

in the two-round Pass construction:

Two-round witness hiding Pass construction:

1. The Verifier samples a random $r \leftarrow \{0,1\}^k$ from the codomain of f.
2. Let $c_w = \mathsf{Com}(w; R)$, $c_r = \mathsf{Com}(0; R')$ for randomness R, R'. The Prover outputs (c_w, c_r, π) where π is a NIWI proof for S_{b,c_w,c_r} where the witness the Prover uses is (w, R).
3. The Verifier runs the NIWI verification process on (c_w, c_r, π).

Using Surjectivity Against Malicious CRS Authorities Suppose, as discussed above, that the verifier's message was moved into the CRS. Our main observation is that if a (possibly maliciously chosen) CRS is from the codomain of a surjective one-way function, then no matter the choice of randomness r' used in the commitment c, there exists a preimage of r' under f. This fact guarantees a second witness for a fixed NIWI proof therefore enabling a hybrid argument using the witness-indistinguishability property to argue witness hiding. In fact, we obtain zero-knowledge in the case of an honestly generated CRS by observing the fact that for every CRS, the simulator can use the second witness to produce a proof. The surjectivity of the one-way function guarantees witness hiding even if the CRS is maliciously chosen. However, surjectivity certainly removes the possibility of obtaining a *proof* (statistical soundness) and arguing computational soundness must be carefully done.

Instantiating the Cryptographic Primitives A few objectives naturally arise from attempting to compress the two-round Pass construction into a NIZK in the CRS model where we insist that the one-way function f is surjective:

1. To instantiate the surjective one-way function, we observe that the short integer solution (SIS) problem gives a suitable instantiation for the surjective one-way function. The SIS problem takes a uniform random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ where $m = \Omega(n \log q)$ and asks whether it is difficult to find a nonzero vector $\mathbf{x} \in \mathbb{Z}^m$ with ℓ_2-norm bounded by some real number β such that $\mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q$. Straightforward counting shows that any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for $m = 2n \log q$ has a short vector solution whose ℓ_2-norm is bounded by $\sqrt{m}$. This fact guarantees a second witness for a suitable statement chosen with respect to $\mathbf{A}$, analogously to how the statements in Pass' construction are defined with respect to f. Then we proceed by a careful complexity leveraging argument with respect to the SIS problem to obtain computational soundness of our NIZK.
2. We'd like to replace the NIWI protocol with a primitive based on well-founded post-quantum secure assumptions. Currently, NIWIs are only known from bilinear pairing assumptions [GOS06] and indistinguishability obfuscation [BP15]. However, we can relax the non-interactive requirement on the NIWIs to a two-round protocol and place the first round message of the two-round protocol in the CRS. Two-round public coin witness indistinguishable protocols, known as ZAPs [DN00], are known from LWE [BFJ+20, GJJM20]. These constructions, however, have only non-adaptive soundness. Crucially, we need adaptively sound witness-indistinguishable argument systems. This adaptive soundness requirement is necessary because the suitable statement we define is chosen in response to the matrix $\mathbf{A}$ placed in the CRS. At the cost of downgrading from public-coin to public-verifiability, [LVW19] obtains computational adaptive soundness in a construction from subexponential LWE. We observe, however, that a simple construction along the ideas proposed in the introduction of [BFJ+20], and similar to that of [LVW19], achieves both public-coin and computational adaptive soundness. This construction idea requires the existence of super dense PKE from LWE, which was not known to exist. In Appendix A, we give the first such construction and use it to construct a computational adaptively sound ZAP in Appendix B.

3. So far, we have only discussed how to achieve witness hiding in the presence of a malicious CRS. However, we would like to strengthen this and achieve the notions of security discussed previously, malicious CRS witness semantic security and malicious CRS verifiable witness semantic security. It turns out that our NIZK protocol satisfies a notion of simulation, where the simulator is allowed to non-uniformly depend on the malicious CRS, that implies malicious CRS witness semantic security, malicious CRS verifiable witness semantic security, and malicious reusable CRS witness semantic security. Essentially, the CRS consists of a string s in the codomain of a surjective one-way function f, and the non-uniform advice dependent on the CRS that the simulator requires is a preimage of s under f.

3 Preliminaries

Notation We use λ to denote the security parameter, $\mathsf{negl}(\lambda)$ to denote any function asymptotically smaller than $\frac{1}{p(\lambda)}$ for any polynomial $p(\cdot)$.

For a language $L \in \mathsf{NP}$, let V be a verifier for L, then define the relation R_L as the corresponding set of instance-witnesses, $R_L \triangleq \{(x,w) : V(x,w) = 1\}$. Define $R_L(x) \triangleq \{w : V(x,w) = 1\}$ to be the set of all witnesses for instance x.

If S is a set, then $x \xleftarrow{\$} S$ denotes sampling element x from S uniformly at random. If D is a distribution, then $x \leftarrow D$ denotes sampling element x according to D.

Definition 3.1 (Probability Ensemble). *A probability ensemble D over a support S is map from $\mathbb{N}$ to distributions of strings over S.*

Definition 3.2 (Polynomially-bounded Probability Ensemble). *Probability ensemble D is polynomially-bounded if there exists a polynomial $p(\cdot)$ such that for all $\lambda \in \mathbb{N}$ and for all x in the support of $D(\lambda)$, $|x| \leq p(\lambda)$.*

Definition 3.3 (Efficiently Samplable Probability Ensemble). *Probability ensemble D is efficiently samplable if for every $\lambda \in \mathbb{N}$ there exists a PPT algorithm that samples from $D(\lambda)$.*

Definition 3.4 (Computationally Indistinguishability). *Two ensembles of distributions D_1 and D_2 are computationally indistinguishable if for all polynomial sized distinguishers $\mathcal{A}$,*

$$\left| \Pr_{x \leftarrow D_1(\lambda)}[\mathcal{A}(1^\lambda, x) = 1] - \Pr_{x \leftarrow D_2(\lambda)}[\mathcal{A}(1^\lambda, x) = 1] \right| \leq \mathsf{negl}(\lambda).$$

We use the notation $D_1 \approx_c D_2$ to denote that two ensembles of distributions D_1 and D_2 are computationally indistinguishable.

Remark 3.5. We use the terminology sub-exponential hardness of a decisional problem P to denote that for any adversary of size $2^{O(\lambda^c)}$ for constant $c \in (0,1)$, the distinguishing advantage is bounded by $2^{-\Omega(\lambda^c)}$.

We use the terminology polynomial hardness of a decisional problem P to denote that for any adversary of size $n^{O(1)}$, the distinguishing advantage is bounded by $n^{-\omega(1)}$. We analogously use the same terminology for search problems.

Definition 3.6 (Non-interactive Commitment Scheme Syntax). *A commitment scheme is defined by a single PPT algorithm Com with the following syntax:*

> $c \leftarrow \mathsf{Com}(m; r)$*: The PPT algorithm Com receives a message m from message space $\mathcal{M}$ and uses randomness (the opening) $r \in \{0,1\}^{\mathsf{poly}(|m|)}$ and outputs a commitment $c \in \mathcal{C}$ for some commitment space $\mathcal{C}$.*

Definition 3.7 (Commitment Scheme Properties). *We assume a non-interactive commitment scheme that satisfies the following two properties*

> ***Perfectly Binding****: For all $m_0, m_1 \in \mathcal{M}$ such that $m_0 \neq m_1$ and for all $r_0, r_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$,*
>
> $$\Pr[\mathsf{Com}(m_0; r_0) = \mathsf{Com}(m_1; r_1)] = 0.$$
>
> ***Computationally Hiding****: For all $m_0, m_1 \in \mathcal{M}$, the following two probability ensembles are computationally indistinguishable,*
>
> $$\{\mathsf{Com}(m_0; r)\}_r \approx_c \{\mathsf{Com}(m_1; r)\}_r.$$

3.1 Lattices

Notation We use bold font upper case characters for matrices $\mathbf{A}$ and bold font lower case characters for vectors $\mathbf{v}$.

An m-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. Let $k \leq m$ be the rank of lattice so that Λ is generated as all integer linear combinations of some k linearly independent basis vectors $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$. For modulus q and ring $\mathbb{Z}_q$, we consider elements in the balanced representation $[-q/2, q/2)$. In particular, we consider the ℓ_2-norm for vectors in $\mathbb{Z}_q$ in the balanced representation.

Definition 3.8 (Decisional Learning with Errors Problem)**.** *Let $n \geq 1$ be a parameter for dimension, and let $q = q(n) \geq 2$ be a modulus. Let $m \geq 1$ be a parameter for number of samples. Let $\chi = \chi(n)$ be an error distribution over $\mathbb{Z}_q$. The decisional learning with errors problem $\mathsf{LWE}_{n,m,q,\chi}$ is to distinguish between the following two distributions:*

$$\left\{(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \xleftarrow{\$} \chi^m\right\}$$

and

$$\left\{(\mathbf{A}, \mathbf{u}) \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m\right\}$$

Definition 3.9 (Bounded Error Distributions)**.** *Let $B = B(\lambda)$ such that $B(\lambda) \in \mathbb{N}$. We say that a family of distributions $\chi = \{\chi_\lambda\}_{\lambda \in \mathbb{N}}$ over the integers is B-bounded if for all $\lambda \in \mathbb{N}$,*

$$\Pr\left[|x| \leq B(\lambda) \mid x \leftarrow \chi_\lambda\right] = 1.$$

Definition 3.10 (Short Integer Solution Problem (SIS) [Ajt96])**.** *Let $n, m, q \in \mathbb{Z}$ such that $m = \Omega(n \log q) \subseteq \mathsf{poly}(n)$. Let $\beta > 0$ be a real number such that $\beta < q$. Let $\mathbf{A}$ be a uniformly random matrix over $\mathbb{Z}_q^{n \times m}$. The short integer solution search problem $\mathsf{SIS}_{n,m,q,\beta}$ is solved by finding a nonzero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $|\mathbf{x}|_2 \leq \beta$ and $\mathbf{A}\mathbf{x} \equiv 0 \bmod q$.*

Theorem 3.11 (Hardness of SIS, Imported from [MP13])**.** *Let n and $m = \mathsf{poly}(n)$ be integers, let $\beta \geq \beta_\infty \geq 1$ be reals, let $Z = \{\mathbf{z} \in \mathbb{Z}^m : \|\mathbf{z}\|_2 \leq \beta$ and $\|\mathbf{z}\|_\infty \leq \beta_\infty\}$ and let $q \geq \beta \cdot n^\delta$ for some constant $\delta > 0$. The solving (on the average, with non-negligible probability) SIS with parameters n, m, q and solution set $\mathbb{Z} \setminus \{0\}$ is at least as hard as approximating lattice problems in the worst case on n-dimensional lattices to within $\gamma = \max\{1, \beta \cdot \beta_\infty / q\} \cdot \tilde{O}(\beta\sqrt{n})$ factors.*

Lemma 3.12 (Existence of a Short Solution)**.** *For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and such that $m > 2n \log q$, there exists a short vector $\mathbf{x} \in \{-1, 0, 1\}^m$ such that $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.*

Proof. We count and apply the pigeonhole principle. The number of vectors in $\{0, 1\}^m$ is 2^m. The total number of vectors in the codomain is q^n. Plugging for n, m, q, we see that $2^m > q^n$. Therefore, there exists $\mathbf{r}, \mathbf{r}' \in \{0, 1\}^m$ such that $\mathbf{r} \neq \mathbf{r}'$ and $\mathbf{A}\mathbf{r} = \mathbf{A}\mathbf{r}' \bmod q$. Then observe that $\mathbf{r} - \mathbf{r}' \in \{-1, 0, 1\}^m$ and $\mathbf{r} - \mathbf{r}' \neq \mathbf{0}$. □

Corollary 3.13 (Existence of a Short Solution, Alternate)**.** *For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and such that $m > 2n \log q$, there exists a short vector $\mathbf{x} \in \mathbb{Z}_q^m$ such that $\mathbf{x} \neq \mathbf{0}$, $\|\mathbf{x}\|_2 \leq \sqrt{m}$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.*

Proof. Immediate by Lemma 3.12. □

Remark 3.14 (Non-interactive Perfectly Binding Commitment Schemes from LWE-based PKEs)**.** Any PKE with perfect decryption correctness gives a non-interactive commitment. As observed previously [LS19], this perfect decryption correctness implies perfect binding even though the committer is allowed to choose the public key maliciously. Since LWE with polynomial modulus-to-noise ratio under a bounded error distribution gives Regev encryption with perfect decryption error [AEKP19], it also gives non-interactive perfectly binding, computationally hiding non-interactive commitments. These facts are used in our main construction.

Remark 3.15 (Commitments to Strings)**.** Throughout this paper, we use the convention that a commitment to a string is done bit-by-bit.

Definition 3.16 (Function class $\mathcal{F}_{\lambda,k(\lambda),s(\lambda)}$ of log-depth circuits)**.** *For any polynomials $k(\cdot)$, $s(\cdot) = \omega(k(\cdot))$ and any $\lambda \in \mathbb{N}$, let $\mathcal{F}_{\lambda,k(\lambda),s(\lambda)}$ (shorthand notation: $\mathcal{F}_{\lambda,k,s}$) denote the class of NC^1 circuits of size $s(\lambda)$ that on input $k(\lambda)$ bits output λ bits. Namely, $f : \{0,1\}^{k(\lambda)} \to \{0,1\}^{\lambda}$ is in $\mathcal{F}_{\lambda,k,s}$ if it has size $s(\lambda)$ and depth bounded by $O(\log \lambda)$.*

Definition 3.17 (μ_{CI}-Correlation Intractable Hash Function Family)**.** *A hash function family $\mathcal{H} = (\mathsf{Setup}, \mathsf{Eval})$ is μ_{CI}-correlation intractable (CI) with respect to $\mathcal{F} = \{\mathcal{F}_{\lambda,s(\lambda)}\}_{\lambda\in\mathbb{N}}$ as defined in Definition 3.16, if the following two properties hold:*

- ***Correlation Intractability**: For every large enough $\lambda \in \mathbb{N}$, for every non-uniform polynomial size adversary $\mathcal{A}$, every polynomial s, every $f \in \mathcal{F}_{\lambda,s(\lambda)}$,*

$$\Pr_{K\leftarrow\mathcal{H}.\mathsf{Setup}(1^\lambda,f)}[\mathcal{A}(K) \to x : (x, \mathcal{H}.\mathsf{Eval}(K,x)) = (x, f(x))] \leq \mu_{\mathsf{CI}}(\lambda)$$

- ***Statistical Indistinguishability of Hash Keys**: Moreover, for every $f \in \mathcal{F}_{\lambda,s(\lambda)}$, for every unbounded adversary $\mathcal{A}$, and every large enough $\lambda \in \mathbb{N}$,*

$$\left|\Pr_{K\leftarrow\mathcal{H}.\mathsf{Setup}(1^\lambda,f)}[\mathcal{A}(K) = 1] - \Pr_{K\leftarrow\{0,1\}^\ell}[\mathcal{A}(K) = 1]\right| \leq 2^{-\lambda^{\Omega(1)}}$$

Theorem 3.18 (Implied by Theorem 3.6 [PS19])**.** *Let μ_{LWE} be an explicit upper bound on distinguishing advantage for the decisional LWE problem $\mathsf{LWE}_{n,m,q,\chi}$ for a $\mathsf{poly}(n)$-bounded χ and sufficiently large $q = \mathsf{poly}(m)$. Assuming the μ_{LWE}-hardness of LWE, for all polynomials $k(\cdot), s(\cdot)$ where $s = \omega(k(\cdot)))$, there exists a $(\mu_{\mathsf{LWE}} + \mathsf{negl}(\lambda))$-public coin correlation intractable hash family with respect to $\mathcal{F}_{\lambda,k,s}$.*

Remark 3.19. To understand the public-coin property, observe that if the null circuit is given as input to the setup algorithm for the CI hash family for log-depth circuits [PS19], then this hash key is in fact uniformly random. Moreover, for an arbitrary non-null circuit, this hash key is statistically close to uniform random by the leftover hash lemma. In more detail, in the construction of the CI hash family for log-depth circuits, the hash key consists of a random vector $\mathbf{a}$ and a commitment $\mathbf{C}$ of the form $\mathbf{AR} + \mathbf{x}^\top \otimes \mathbf{G}$ for uniform random matrices $\mathbf{A}, \mathbf{R}$, message $\mathbf{x}$, and gadget matrix $\mathbf{G}$.

3.2 ZAPs

Remark 3.20. All parties in interactive protocols take the security parameters 1^λ as input. However, we notationally omit this input for the sake of brevity.

Notation Let $\langle P(x,w), V(x)\rangle$ denote an execution, or transcript, of a protocol between $P(x,w)$ and $V(x)$ with instance x and witness w. Let $\mathsf{Output}_V(\langle P(x,w), V(x)\rangle)$ denote the verifier's final output.

In the case of a two-message interactive proof/argument, we use α to denote the first message (verifier to prover) and π to denote the second message (prover to verifier).

Definition 3.21 (Two-Message Interactive Proofs)**.** *A two-message interactive protocol (P, V) for deciding a language L is an interactive argument for L if it satisfies the following properties:*

- ***Completeness**: For every $(x,w) \in R_L$,*

$$\Pr[\mathsf{Output}_V(\langle P(x,w), V(x)\rangle) = 1] = 1 - \mathsf{negl}(\lambda)$$

where the probability is over the random coins of P and V.

- ***Adaptive Soundness**: For every polynomial sized adversary P^*, there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$,*

$$\Pr_{\substack{\alpha\leftarrow V(1^\lambda,x)\\(x^*,\pi^*)\leftarrow P^*(1^\lambda,\alpha)}}[x^* \notin L \wedge V(\alpha, x^*, \pi^*) = 1] \leq \mu(\lambda)$$

Remark 3.22. If soundness additionally holds against unbounded provers, the protocol is said to be an interactive *proof.*

Definition 3.23 (Public-coin Two-Message Interactive Proof)**.** *A two-message interactive proof is said to be public coin if there exists a polynomial k such that the first round messages form a distribution on strings of length $k(n)$ that depends only on n, which is the length of the problem instance x. Moreover, the verifier's output is a polynomial time computable function of x, the verifier's first round message α, and π only.*

Definition 3.24 (Computational Witness Indistinguishability [DN00])**.** *A two-message interactive proof satisfies computational witness indistinguishability if the following condition holds. Let $w_1, w_2 \in R_L(x)$ be witnesses to $x \in L$. Then for all first-round messages α, and for all polynomial-sized adversaries $\mathcal{A}$ whom are given w_1, w_2, the distribution on π when the prover has input (x, w_1) and the distribution on π when the prover has input (x, w_2) are indistinguishable when additionally given the statement x and any polynomially bounded auxiliary input z; that is, $\mathcal{A}$ has negligible distinguishing advantage.*

Definition 3.25 (Witness Hiding [FS90, KZ20])**.** *Let $L \in \mathsf{NP}$ and fix some polynomial-time verifier V_L for L defining a relation $R_L = \{(x,w) : V_L(x,w) = 1\}$. An interactive proof (P, V) satisfies witness hiding if for all probability ensembles D with support over R_L, for all polynomial-sized $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a polynomial sized $\mathcal{B}$ and a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for all $\lambda \in \mathbb{N}$*

$$\Pr_{(x,w)\sim D(\lambda)}\left[(x, \mathcal{A}_2(1^\lambda, x, \langle P(1^\lambda, x, w), \mathcal{A}_1(1^\lambda, x)\rangle)) \in R_L\right] \leq \Pr_{(x,w)\sim D(\lambda)}\left[w' \leftarrow \mathcal{B}(1^\lambda, x) : (x, w') \in R_L\right] + \mu(\lambda).$$

Definition 3.26 (ZAP)**.** *A ZAP for a language L is a two-message public coin computational witness indistinguishable argument system for a language L given by three PPT algorithms $(\mathsf{ZAP}_1, \mathsf{ZAP}_2, \mathsf{ZAP}_V)$:*

- $z \leftarrow \mathsf{ZAP}_1(1^\lambda)$*: takes as input a security parameter 1^λ and outputs some first message z.*
- $\pi_z \leftarrow \mathsf{ZAP}_2(z, x, w)$*: takes as input a first message z, an instance x, and a witness w. It outputs an argument π_z.*
- $0/1 \leftarrow \mathsf{ZAP}_V(z, x, \pi_z)$*: takes as input a first message z, an instance x, and an argument π_z. It outputs either* 0 *(reject) or* 1 *(accept).*

Theorem 3.27 (ZAPs from Subexponential LWE)**.** *Assuming the subexponential hardness of* LWE*, there exist two-message public coin WI arguments for* NP.

Proof. We adapt the construction of [BFJ+20] by noting there indeed exists super-dense public key cryptosystems from LWE. See Section A for a construction of the super dense PKE from LWE and Section B for a construction of the ZAP from LWE. □

Remark 3.28. We note that the construction given in [LVW19] applies to a wider class of Σ-protocols. In particular, if the bad challenge function is not in NC^1, then a bootstrapping step from [PS19] is required and the public coin property is lost. Our construction focuses specifically on the Blum protocol for Graph Hamiltonicity where the bad challenge function is in NC^1.

Remark 3.29. We note that the verifier's message is reusable across multiple executions with no loss in soundness.

3.3 NIZK

Definition 3.30 (Non-interactive Argument System Syntax)**.** *A non-interactive argument system* $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language $L \in \mathsf{NP}$ is defined by a triple of PPT algorithms:*

- $\mathsf{CRS} \leftarrow \mathsf{GenCRS}(1^\lambda)$*: takes as input a security parameter 1^λ and outputs a common reference string* CRS.
- $\pi \leftarrow \mathsf{Prove}(\mathsf{CRS}, x, w)$*: takes as input a common reference string* CRS*, an instance x, and a witness w for x. It outputs a message π.*

- $0/1 \leftarrow \mathsf{Verify}(\mathsf{CRS}, x, \pi)$*: takes as input a common reference string* CRS*, an instance* x*, and a message* π*. It outputs either* 0 *(reject) or* 1 *(accept).*

Definition 3.31 (Non-interactive Argument System Properties)**.** *A non-interactive zero-knowledge (NIZK) argument system* $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $L \in \mathsf{NP}$ *satisfies the following three properties:*

1. ***Perfect Completeness****: for all* $x \in L$ *and* $w \in R_L(x)$*,*

$$\Pr\left[\begin{array}{c} \mathsf{CRS} \leftarrow \mathsf{GenCRS}(1^\lambda), \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}, x, w), \\ \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1 \end{array}\right] = 1$$

2. ***Computational Soundness****: for every polynomial-size prover* P^**, there exists a negligible function* $\mu(\lambda)$ *such that for all* $\lambda \in \mathbb{N}$ *and for all* $x \notin L$

$$\Pr_{\substack{\mathsf{CRS} \leftarrow \mathsf{GenCRS}(1^\lambda) \\ \pi^* \leftarrow P^*(1^\lambda, \mathsf{CRS}, x)}} [\mathsf{Verify}(\mathsf{CRS}, x, \pi^*) = 1] \leq \mu(\lambda)$$

3. ***Adaptive Computational Zero-Knowledge****: there exists a PPT algorithm* Sim *split into two stages* Sim_1 *and* Sim_2 *such that for all PPT adversaries* $\mathcal{A}$*, the following two distributions* $\mathsf{Expt}_{Real,\mathcal{A}}$ *and* $\mathsf{Expt}_{Sim,\mathcal{A}}$ *are computationally indistinguishable.*

$\mathsf{Expt}_{Real,\mathcal{A}}(1^\lambda)$*:*

(i) $\mathsf{CRS} \leftarrow \mathsf{GenCRS}(1^\lambda)$*.*

(ii) $(x, w) \leftarrow \mathcal{A}(1^\lambda, \mathsf{CRS})$ *such that* $(x, w) \in R_L$*.*

(iii) $\pi \leftarrow \mathsf{Prove}(\mathsf{CRS}, x, w)$*.*

(iv) Output (CRS, x, π)*.*

$\mathsf{Expt}_{Sim,\mathcal{A}}(1^\lambda)$

(i) $\mathsf{CRS}, \tau \leftarrow \mathsf{Sim}_1(1^\lambda)$*.*

(ii) $(x, w) \leftarrow \mathcal{A}(1^\lambda, \mathsf{CRS})$ *such that* $(x, w) \in R_L$*.*

(iii) $\pi \leftarrow \mathsf{Sim}_2(x, \tau)$*.*

(iv) Output (CRS, x, π)*.*

3.3.1 Accountability [AADG21]

Definition 3.32 (Judge algorithm, Imported from [AADG21])**.** *We augment a NIZK system with another PPT algorithm* Judge *where the input and output is given by* $b \leftarrow \mathsf{Judge}(\mathsf{CRS}, \tau)$*. The algorithm takes as input* CRS *and some transcript* τ *and outputs a bit* b*. It outputs a bit* $b = 1$ *to indicate* τ *proves that the* CRS *is corrupted, and outputs* $b = 0$ *otherwise.*

Definition 3.33 (Accountability, Imported from [AADG21])**.** *Let* $L \in \mathsf{NP}$ *and let* R_L *be the corresponding* NP *relation. A NIZK system* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Judge})$ *satisfies accountability with respect to distribution* $\mathcal{D}$ *if for all sufficiently large security parameters* $\lambda \in \mathbb{N}$*, any PPT adversary* $\mathcal{A}$*, if there exists a non-negligible function* $\epsilon_1(\cdot)$ *such that*

$$\Pr[\mathsf{Acc.Real}_{\Pi,\mathcal{A},q}(\lambda) = 1] \geq \epsilon_1(\lambda)$$

then there exists PPT oracle-aided algorithm Ext_1 *making at most* q *queries and PPT algorithm* Ext_2*, and a non-negligible function* $\epsilon_2(\cdot)$ *such that*

$$\Pr[\mathsf{Acc.Ext}_{\Pi,\mathsf{Ext},q}(\lambda) = 1] \geq \epsilon_1(\lambda)$$

where the following two random variables $\mathsf{Acc.Real}_{\Pi,\mathcal{A},q}(\lambda)$ *and* $\mathsf{Acc.Ext}_{\Pi,\mathsf{Ext},q}(\lambda)$ *are defined as:*

$\mathsf{Acc.Real}_{\Pi,\mathcal{A},q}(\lambda)$*:*

(i) $\mathsf{CRS}^* \leftarrow \mathcal{A}(1^\lambda)$.

(ii) *For* $i \in [q]$, $(x_i, w_i) \leftarrow \mathcal{D}$,
$\pi_i \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x_i, w_i)$.
Abort and output 0 *if* $\mathsf{Verify}(\mathsf{CRS}^*, x_i, \pi_i) \neq 1$.

(iii) $(i, x'_i, w'_i) \leftarrow \mathcal{A}\left(\{(x_i, \pi_i)\}_{i\in[q]}\right)$.

(iv) *Output* 1 *if* $x_i = x'_i$ *and* $(x_i, w'_i) \in R_L$ *and* 0 *otherwise.*

$\mathsf{Acc.Ext}_{\Pi,\mathsf{Ext},q}(\lambda)$

(i) $\mathsf{CRS}^* \leftarrow \mathcal{A}(1^\lambda)$.

(ii) $\{(x_i, \pi_i)\}_{i\in[q]}, \mathsf{st} \leftarrow \mathsf{Ext}_1(\mathsf{CRS}^*)$.

(iii) $(i, x'_i, w'_i) \leftarrow \mathcal{A}\left(\{(x_i, \pi_i)\}_{i\in[q]}\right)$.

(iv) $\tau \leftarrow \mathsf{Ext}_2(\mathsf{st}, (i, x'_i, w'_i))$.

(v) *Output* 1 *if* $\mathsf{Judge}(\mathsf{CRS}^*, \tau) = 1$.

Definition 3.34 (Defamation-free, Imported from [AADG21])**.** *For every PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\cdot)$*, such that for all* $\lambda \in \mathbb{N}$*:*

$$\Pr[\mathsf{Judge}(\mathsf{CRS}, \mathcal{A}(\mathsf{CRS}) = 1] \leq \mu(\lambda)$$

where $\mathsf{CRS} \leftarrow \mathsf{GenCRS}(1^\lambda)$.

4 Witness Semantic Security

Recall that, informally, we would like to hide partial information about the prover's witness from an adversarial verifier interacting with the prover. We capture this intuition by formally defining *witness semantic security*, which states that interacting with an honest prover only gives negligibly more advantage in extracting out a function of the witness than attempting to extract out a function of the witness from the statement alone.

Definition 4.1 (Witness Semantic Security)**.** *An interactive argument system* $(\mathcal{P}, \mathcal{V})$ *for a language* $L \in \mathsf{NP}$ *is witness semantic secure if for all polynomially-bounded efficiently samplable probability ensembles* D *over* $\{(x, w, \mathsf{Aux}, f, y) \mid y = f(w), (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$*, where* $\mathscr{F}$ *is a set of deterministic functions, for all polynomial sized* $\mathcal{A}_1$ *and polynomial sized* $\mathcal{A}_2$ *which additionally takes as input a state* τ *generated by* $\mathcal{A}_1$*, there exists a polynomial sized* $\mathcal{B}$ *and there exists a negligible function* $\mu(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$

$$\begin{aligned} &\Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\ \mathcal{A}_2\left(1^\lambda, \tau, \langle \mathcal{P}(1^\lambda, x, w), \mathcal{A}_1(1^\lambda)\rangle, x, \mathsf{Aux}, f\right) = y\ \right] \\ &\leq \Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) = y\right] + \mu(\lambda). \end{aligned}$$

Remark 4.2. In this work, all interactive protocols we consider are two rounds. Note that in the definition above, the verifier does not see any outputs of D when generating its first message. The typical situation we consider is one where the verifier's first message is sent much earlier than the generation of any statement or proof. Our construction satisfies the property that the verifier's first message can be reused across multiple proofs.

Remark 4.3. Observe that this definition captures the case where $\mathcal{F}$ consists of a single function.

Lemma 4.4 (WSS implies Witness Indistinguishability)**.** *If an interactive argument system* $(\mathcal{P}, \mathcal{V})$ *for a language* $L \in \mathsf{NP}$ *is witness semantic secure (WSS), then it is witness indistinguishable (WI).*

Proof. We prove the contrapositive. Consider any language $L \in \mathsf{NP}$ and a corresponding relation R_L on statements and witnesses. Suppose that $(\mathcal{P}, \mathcal{V})$ is not witness indistinguishable. Then, there exists a sequence of statements and witnesses $\left(x_\lambda, w_\lambda^{(1)}, w_\lambda^{(2)}\right)_{\lambda\in\mathbb{N}}$ where $w_\lambda^{(1)}, w_\lambda^{(2)} \in R_L(x_\lambda)$, and $w_\lambda^{(1)} \neq w_\lambda^{(2)}$ (such a second witness can always exist WLOG by two different paddings), and a polynomial sized $\mathcal{A}, \mathcal{V}^*$ such that $\mathcal{V}^*$ distinguishes with non-negligible advantage between the probability ensembles $\{\langle \mathcal{P}(x_\lambda, w_\lambda^{(1)}), \mathcal{V}^*(x_\lambda)\rangle\}$ and $\{\langle \mathcal{P}(x_\lambda, w_\lambda^{(2)}), \mathcal{V}^*(x_\lambda)\rangle\}$. Then we construct a specific probability ensemble D for which WSS is violated. Let i_λ to be the smallest index on which the string $w_\lambda^{(1)}$ differs from the string $w_\lambda^{(2)}$. Then, let f_λ be a function that on input w outputs the i_λ-th bit. Now consider the distribution $D(\lambda)$ that samples $(x_\lambda, w_\lambda^{(1)}, \mathsf{Aux} = (w_\lambda^{(1)}, w_\lambda^{(2)}), f_\lambda, f_\lambda(w_\lambda^{(1)}))$ with probability 1/2 and $(x_\lambda, w_\lambda^{(2)}, \mathsf{Aux} = (w_\lambda^{(1)}, w_\lambda^{(2)}), f_\lambda, f_\lambda(w_\lambda^{(2)}))$ otherwise. Observe that *any* algorithm given $(1^\lambda, x_\lambda, \mathsf{Aux} = (w_\lambda^{(1)}, w_\lambda^{(2)}), f_\lambda)$ has at best 1/2 probability of guessing $f_\lambda(w_\lambda^{(b)})$ sampled from $D(\lambda)$. Yet, an algorithm additionally given the transcript $\langle \mathcal{P}(x_\lambda, w_\lambda^{(2)}), \mathcal{V}^*(x_\lambda)\rangle$ can run $\mathcal{A}$ to recover $f_\lambda(w_\lambda^{(b)})$ with non-negligible advantage over 1/2. □

Having defined witness semantic security, we now define malicious CRS witness semantic security. This definition captures the intuition that the adversary, in the CRS model and in the non-interactive setting, should not be capable of learning a function of the witness even given a proof with respect to a malicious CRS that the adversary has knowledge about.

Definition 4.5 (Malicious CRS Witness Semantic Security)**.** *A non-interactive argument system* $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $L \in \mathsf{NP}$ *is malicious CRS witness semantic security if for all polynomially-bounded efficiently samplable probability ensembles* D *over* $\{(x, w, \mathsf{Aux}, f, y) \mid y = f(w), (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$ *where* $\mathscr{F}$ *is a set of deterministic functions, for all unbounded* $\mathcal{A}_1$ *and polynomial-sized* $\mathcal{A}_2$*, there exists a polynomial sized* $\mathcal{B}$ *and a negligible function* $\mu(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$

$$\Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\begin{array}{c} (\mathsf{CRS}^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x, w) \\ y \leftarrow \mathcal{A}_2(1^\lambda, \tau, x, \pi, \mathsf{Aux}, f) \end{array}\right] \leq \Pr_{(x,w,\mathsf{Aux},f,y)\leftarrow D(\lambda)}\left[\mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) = y\right] + \mu(\lambda).$$

In the following Lemma, we make the simple observation that any interactive proof (argument) that satisfies malicious CRS witness semantic security implies a two-round publicly-verifiable, witness semantic secure proof (argument) in the plain model. Note that the converse is not true as the definition of malicious CRS witness semantic security considers unbounded adversaries $\mathcal{A}_1$ that generate a CRS. This observation justifies our focus on the malicious CRS setting for the main construction in the paper.

Lemma 4.6 (Malicious CRS WSS implies Two-Round WSS)**.** *If there exists a NIZK proof (argument)* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $L \in \mathsf{NP}$ *that satisfies malicious CRS witness semantic security (Def. 4.5), then there exists a two-round publicly-verifiable witness semantic secure proof (argument) (Def. 4.1) for* L *in the plain model. If the CRS generation in the NIZK proof is additionally uniformly random, then the two-round protocol is also public coin.*

Proof. Define V_1 as a polynomial-time algorithm that on input 1^λ outputs $\mathsf{msg}_1 \leftarrow \Pi.\mathsf{GenCRS}(1^\lambda)$. Define P as a polynomial-time algorithm that takes as input $(1^\lambda, x, w, \mathsf{msg}_1)$ and outputs $\pi \leftarrow \Pi.\mathsf{Prove}(1^\lambda, \mathsf{msg}_1, x, w)$. Define V_2 as a polynomial-time algorithm that on input $(1^\lambda, x, \mathsf{msg}_1, \pi)$ outputs $\Pi.\mathsf{Verify}(1^\lambda, \mathsf{msg}_1, x, \pi)$. The claim is that $(P, V = (V_1, V_2))$ is a two-round publicly-verifiable witness semantic secure proof (argument). Completeness, adaptive soundness, and witness semantic security follow immediately from the definitions. Public verifiability also immediately follows since $\Pi.\mathsf{Verify}$ has only public inputs given by the first and second round messages since no state information was passed from V_1 to V_2. With regards to being public coin, if the CRS is uniformly random, then the first round message is uniformly random. □

As previously stated, our construction will focus on the malicious CRS setting because all security results in the malicious CRS setting translate via Lemma 4.6 to the two-round setting.

4.1 Verifiable Witness Semantic Security

To motivate the following definitions, recall that the standard notion of witness hiding (Def. 3.25) with respect to some distribution D over statements asks that any adversary is unable to use the proof to produce *any* new witness it could not have produced by only seeing the statement. Therefore, witness hiding is *not* immediately implied by witness semantic security (Def. 4.1) because Def. 4.1 only prevents an adversary from learning information about a *specific* witness. A desirable objective is then to define a notion of witness semantic security that captures witness hiding.

Preventing an adversary from producing *any* witness, or even a fixed function f of *any* witness, via zero-knowledge implicitly requires an efficient verification procedure. For example, consider the task of proving that zero-knowledge implies witness hiding. Assuming we have an efficient algorithm that breaks witness hiding by extracting some non-trivial witness for a statement from an honestly generated proof, we can construct an efficient reduction that breaks zero-knowledge by distinguishing between an honest prover's output and a simulator's output. Crucially, this distinguisher needs to efficiently verify the validity of the witness, a task that is efficient because the language L is in NP.

Now consider the following issue: suppose we define a variant of witness semantic security to prevent an adversary, with some statement x, from learning any value y such that $y = f(w')$ for any witness w' for x. Then, the reduction that shows zero-knowledge implies this variant may be inefficient if verifying that y is $f(w')$ for some witness w' for x is inefficient. In fact, in Section 6 we show that without imposing an efficient verifiability property, such a privacy guarantee is unlikely to be provided by even distributional zero-knowledge. Therefore, we introduce the following notion of verifiable witness semantic security in a manner similar to the definitions introduced for witness semantic security (Def. 4.1). Note that we focus on functions f with long outputs instead of predicates since a predicate f would be trivially easy-to-learn if there ever existed distinct witnesses w and w' where $f(w) = 0$ and $f(w') = 1$.

Definition 4.7 (Verifiable Witness Semantic Security)**.** *An interactive argument system $(\mathcal{P}, \mathcal{V})$ for a language $L \in \mathsf{NP}$ is verifiable witness semantic secure (VWSS) if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x, w, \mathsf{Aux}, f) \mid (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions such that for all $f \in \mathscr{F}$, there exists a polynomial-time verification algorithm V_f such that $V_f(x, y) = 1$ if and only if $y = f(w')$ for some w' such that $(x, w') \in R_L$, for all polynomial sized $\mathcal{A}_1$ and polynomial sized $\mathcal{A}_2$ which additionally takes as input a state τ generated by $\mathcal{A}_1$, there exists a polynomial sized $\mathcal{B}$ and there exists a negligible function $\mu(\cdot)$ such that*

$$\Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[\begin{array}{c} y \leftarrow \mathcal{A}_2\left(1^\lambda, \tau, \langle \mathcal{P}(1^\lambda, x, w), \mathcal{A}_1(1^\lambda)\rangle, x, \mathsf{Aux}, f\right) \\ s.t.\ \exists w', y = f(w') \wedge (x, w') \in R_L \end{array}\right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[y \leftarrow \mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) : \exists w', y = f(w') \wedge (x, w') \in R_L\right] + \mu(\lambda),$$

where WLOG Aux *contains a description of V_f.*

Lemma 4.8 (VWSS implies Witness Hiding)**.** *If an interactive argument system $(\mathcal{P}, \mathcal{V})$ for a language $L \in \mathsf{NP}$ is verifiable witness semantic secure (VWSS), then it is witness hiding.*

Proof. The proof follows by viewing the definition of witness hiding as a special case of verifiable witness semantic security (Def. 4.7). The specific case is immediately seen by considering all distributions D where the distribution has support with syntax as given in Definition 4.7, in which $\mathcal{F}$ contains only the identity function. □ □

Definition 4.9 (Malicious CRS Verifiable Witness Semantic Security)**.** *A non-interactive argument system* $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language $L \in \mathsf{NP}$ is malicious CRS verifiable witness semantic security if for all polynomially-bounded efficiently samplable probability ensembles D over $\{(x, w, \mathsf{Aux}, f) \mid (x, w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions such that for all $f \in \mathscr{F}$, there exists a polynomial-time verification algorithm V_f such that $V_f(x, y) = 1$ if and only if $y = f(w')$ for some w' such that $(x, w') \in R_L$, if for all unbounded $\mathcal{A}_1$ and polynomial-sized $\mathcal{A}_2$, there exists a polynomial-sized $\mathcal{B}$ and a negligible function $\mu(\cdot)$ such that*

$$\Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[\begin{array}{c} (\mathsf{CRS}^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x, w) \\ y \leftarrow \mathcal{A}_2(1^\lambda, \tau, x, \pi, \mathsf{Aux}, f) \\ y = f(w') \wedge (x, w') \in R_L \end{array}\right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)}\left[y \leftarrow \mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) : \exists w', y = f(w') \wedge (x, w') \in R_L\right] + \mu(\lambda).$$

Without loss of generality, assume that V_f is given in the auxiliary input Aux*.*

Lemma 4.10 (Malicious CRS VWSS implies Two Round VWSS)**.** *If there exists a NIZK proof (argument)* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *that satisfies malicious CRS VWSS (Def. 4.9), then there exists a two-round* *publicly-verifiable witness semantic secure proof (argument) (Def. 4.7) in the plain model. If the CRS generation in the NIZK proof is additionally uniformly random, then the two-round protocol is also public coin.*

Proof. The proof is identical to that of Lemma 4.6. □

4.1.1 Malicious CRS Verifiable Witness Semantic Security Implies Accountability

We observe that the definition of malicious CRS verifiable witness semantic security (Def. 4.9) implies accountability (Def. 3.33). More specifically, the conditional statement in Definition 3.33 is rendered vacuously true: No adversary participating in the experiment $\mathsf{Acc.Real}_{\Pi,\mathcal{A},q}(\lambda)$ succeeds with non-negligible probability. Since no adversary can cheat with more than negligible probability, the concept of a Judge algorithm can be removed and the defamation-free property becomes meaningless.

Lemma 4.11 (Malicious CRS Verifiable Witness Semantic Security Implies Accountability)**.** *Let D be an efficiently samplable distribution defined over instance-witness pairs for an* NP *language L such that for $(x, w) \leftarrow D$ any polynomially-sized adversary given x has negligible probability of finding a witness w' such that $(x, w') \in R_L$. If $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ is a NIZK argument/proof that satisfies malicious CRS verifiable witness semantic security (Def. 4.9), then Π also satisfies accountability with respect to D (Def. 3.33).*

Proof. Here, the function f is implicitly the identity function. Observe that malicious CRS verifiable witness semantic security implies that for any polynomial-sized adversary $\mathcal{A}$, the probability that the real experiment in Definition 3.33, $\mathsf{Acc.Real}_{\Pi,\mathcal{A},q}(\lambda) = 1$, must be negligible. Therefore, accountability holds as the conditional statement is vacuously true. □

4.2 Malicious CRS Non-Uniform Zero-Knowledge with Auxiliary Information

In order to show that our NIZK construction satisfies Def. 4.5 and Def. 4.9, we will actually show that our NIZK construction satisfies a stronger simulation-based definition, given below, that we term *malicious CRS non-uniform zero-knowledge with auxiliary information.* This definition implies the two forms of witness semantic security above (Def. 4.5 and Def. 4.9). The term non-uniform refers to the fact that the simulator is a polynomial-sized circuit that *non-uniformly* depends on the CRS.

Definition 4.12 (Malicious CRS Non-Uniform Zero-Knowledge with Auxiliary Information (NUZK))**.** *A NIZK protocol for language $L \in \mathsf{NP}$ is malicious CRS non-uniform zero-knowledge with auxiliary information if for all constants $c_1, c_2, c_3, c_4, c_5 > 0$, there exists $\lambda^* \in \mathbb{N}$ such that for all $\lambda > \lambda^*$, for all common reference strings $\mathsf{CRS} \in \{0,1\}^{\lambda^{c_1}}$, there exists a circuit $\mathsf{Sim}_{\mathsf{CRS}}$ of size λ^{c_2}, such that for all (x, w, Aux) such that $|x| \leq \lambda^{c_3}$, $|w| \leq \lambda^{c_4}$, and $|\mathsf{Aux}| \leq \lambda^{c_5}$ and $(x, w) \in R_L$, the following holds:*

$$(x, \mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w), \mathsf{Aux}) \approx_c (x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}}(x), \mathsf{Aux})\,.$$

Observe that in Def. 4.12, the CRS is fixed and not output by the simulator; this is in contrast to the traditional definition of non-interactive zero-knowledge. The simulator $\mathsf{Sim}_{\mathsf{CRS}}$ crucially non-uniformly depends on this CRS, and there is not one simulator that works for every CRS. In the rest of this section, we show that a NIZK that satisfies Def. 4.12 satisfies both Def. 4.5 and Def. 4.9. We begin by defining an intermediate notion that we will use in the proof.

Definition 4.13 (Malicious CRS Non-Uniform *Distributional* Zero-Knowledge with Auxiliary Information (NUDZK))**.** *A NIZK protocol for language $L \in \mathsf{NP}$ is malicious CRS non-uniform distributional zero-knowledge with auxiliary information if for all constants $c_1, c_2, c_3 > 0$, there exists λ^* such that for all $\lambda > \lambda^*$, for all common reference strings CRS, there exists a polynomial sized $\mathsf{Sim}_{\mathsf{CRS}}$ such that for all probability ensembles D in which the distribution $D(\lambda)$ outputs (x, w, Aux) such that $|x| \leq \lambda^{c_1}$, $|w| \leq \lambda^{c_2}$, and $|\mathsf{Aux}| \leq \lambda^{c_3}$ and $(x, w) \in R_L$ the following holds for all polynomial-sized $\mathcal{A}$:*

$$\Bigg| \Pr_{(x,w,\mathsf{Aux}) \leftarrow D(\lambda)} \left[\mathcal{A}(1^\lambda, x, \mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w), \mathsf{Aux}) = 1\right] - \Pr_{(x,w,\mathsf{Aux}) \leftarrow D(\lambda)} \left[\mathcal{A}(1^\lambda, x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}}(x), \mathsf{Aux}) = 1\right] \Bigg| = \mathsf{negl}(\lambda).$$

We next give the following lemma, showing that Def. 4.12 implies Def. 4.13

Lemma 4.14 (Malicious CRS NUZK Implies Malicious CRS NUDZK)**.** *If a NIZK protocol* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $L \in \mathsf{NP}$ *is malicious CRS non-uniform zero-knowledge with auxiliary information, then* Π *is malicious CRS non-uniform distributional zero-knowledge with auxiliary information.*

Proof. We show the contrapositive. Suppose that NIZK Π is not malicious CRS non-uniform distributional zero-knowledge with auxiliary information. Then for infinitely many values of $\lambda \in \mathbb{N}$, there exists CRS_λ of length bounded by a polynomial in λ such that for all polynomial sized $\mathsf{Sim}_{\mathsf{CRS}_\lambda}$ there exists a probability ensemble D such that the distribution $D(\lambda)$ outputs (x, w, Aux) where x, w, Aux have length bounded by a polynomial in λ such that the distinguishing advantage

$$\left| \Pr_{(x,w,\mathsf{Aux}) \leftarrow D(\lambda)} [\mathcal{A}(x, \mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w), \mathsf{Aux}) = 1] - \Pr_{(x,w,\mathsf{Aux}) \leftarrow D(\lambda)} [\mathcal{A}(x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}}(x), \mathsf{Aux}) = 1] \right|$$

is non-negligible. By an averaging argument, there exists (x, w, Aux) from the support of $D(\lambda)$ on which $\mathcal{A}$'s distinguishing advantage is non-negligible. The existence of such (x, w, Aux) immediately implies that Π is not Malicious CRS NUZK. □

4.3 Malicious CRS NUZK Implies Malicious CRS Witness Semantic Security

We now show that malicious CRS NUZK (Def. 4.13) implies malicious CRS witness semantic security (Def. 4.5).

Lemma 4.15. *If a NIZK protocol* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $L \in \mathsf{NP}$ *is malicious CRS non-uniform distributional zero-knowledge with auxiliary information, then* Π *is malicious CRS witness semantic security.*

Proof. For syntax type checking between the distributions in the two definitions, first observe that we can equivalently express the support of distribution D as $\{(x, w, \mathsf{Aux}')\}$ where $\mathsf{Aux}' = (\mathsf{Aux}, f, y)$. Then by the definition of malicious CRS NUDZK, for every CRS and for all $\lambda \in \mathbb{N}$ sufficiently large, there exists a polynomial sized $\mathsf{Sim}_{\mathsf{CRS}}$ such that for all polynomial-sized $\mathcal{B}$:

$$\Bigg| \Pr_{(x,w,\mathsf{Aux}') \leftarrow D(\lambda)} \left[\mathcal{B}(1^\lambda, x, \mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w), \mathsf{Aux}') = 1\right] \\ - \Pr_{(x,w,\mathsf{Aux}') \leftarrow D(\lambda)} \left[\mathcal{B}(1^\lambda, x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}}(x), \mathsf{Aux}') = 1\right] \Bigg| = \mathsf{negl}(\lambda). \tag{1}$$

We will show that if we assume that Π is not malicious CRS witness semantically secure, then there exists a CRS for which we can construct a distinguisher between the distributions on $(x, \mathsf{CRS}, \mathsf{Prove}(\mathsf{CRS}, x, w), \mathsf{Aux}')$ and $(x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}}(x), \mathsf{Aux}')$, contradicting malicious CRS NUDZK. If Π is not malicious CRS witness semantically secure, then there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ where $\mathcal{A}_1$ is computationally unbounded and $\mathcal{A}_2$ is polynomial sized and such that for each polynomial sized $\mathcal{B}$ there exists a polynomial $p(\lambda)$ such that for every $N \in \mathbb{N}$ there exists a $\lambda > N$ such that

$$\Pr_{(x,w,\mathsf{Aux},f,y) \leftarrow D(\lambda)} \left[\begin{array}{c} (\mathsf{CRS}^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x, w) \\ y \leftarrow \mathcal{A}_2(1^\lambda, \tau, x, \pi, \mathsf{Aux}, f) \end{array} \right] \geq \Pr_{(x,w,\mathsf{Aux},f,y) \leftarrow D(\lambda)} \left[\mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) = y\right] + \frac{1}{p(\lambda)}. \tag{2}$$

Fix the CRS considered to be the CRS^* produced by $\mathcal{A}_1(1^\lambda)$. Then observe that the polynomial-sized $\mathcal{A}_2$ is exactly our desired distinguisher for this fixed value of CRS^*. More formally, we construct a polynomial sized distinguisher $\mathcal{B}_\tau$ where for each $\lambda \in \mathbb{N}$, it has an advice string given by τ_λ, where τ_λ is a polynomial-length state produced by $\mathcal{A}_1$ on input 1^λ that can contain the corresponding CRS^* and any possible trapdoors without loss of generality, such that the existence of $\mathcal{B}_\tau$ contradicts Equation 1.

$\mathcal{B}_\tau(1^\lambda, x, \mathsf{CRS}, \pi, (\mathsf{Aux}, f, y))$

1. If the output of $\mathcal{A}_2(1^\lambda, \tau, x, \pi, (\mathsf{Aux}, f))$ is equal to y, then output 0, where 0 represents a guess that π is an honest proof. Otherwise output 1, where 1 represents a guess that π is the simulated proof.

Observe that when π is produced by the simulator $\mathsf{Sim}_{\mathsf{CRS}^*}(x)$, running $\mathcal{A}_2(1^\lambda, \tau, x, \pi, (\mathsf{Aux}, f))$ is running a circuit of the form $\mathcal{B}(1^\lambda, x, \mathsf{Aux}, f)$, corresponding to the right hand side of Equation 2. On the other hand, when π is produced by the honest prover, running $\mathcal{A}_2(1^\lambda, \tau, x, \pi, (\mathsf{Aux}, f))$ corresponds exactly to the left hand side of Equation 2. Therefore, the statement on Equation 2 implies there exists a polynomial $p(\lambda)$ such that for every $N \in \mathbb{N}$ there exists a $\lambda > N$ such that

$$\Bigg| \Pr_{(x,w,\mathsf{Aux}')\leftarrow D(\lambda)} \left[\mathcal{B}_\tau(1^\lambda, x, \mathsf{CRS}^*, \mathsf{Prove}(\mathsf{CRS}^*, x, w), \mathsf{Aux}') = 1\right]$$

$$- \Pr_{(x,w,\mathsf{Aux}')\leftarrow D(\lambda)} \left[\mathcal{B}_\tau(1^\lambda, x, \mathsf{CRS}, \mathsf{Sim}_{\mathsf{CRS}^*}(x), \mathsf{Aux}') = 1\right] \Bigg| \geq \frac{1}{p(\lambda)}$$

Hence, we obtain our desired contradiction with Equation 1 and we conclude that Π must be malicious CRS witness semantically secure.

□

Corollary 4.16 (Malicious CRS NUZK Implies Malicious CRS Witness Semantic Security)**.** *If a NIZK protocol $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ for a language $L \in \mathsf{NP}$ is malicious CRS non-uniform zero-knowledge with auxiliary information, then Π satisfies malicious CRS witness semantic security.*

Proof. The statement follows directly from Lemma 4.14 and Lemma 4.15. □

4.4 Malicious CRS NUZK Implies Malicious CRS Verifiable Witness Semantic Security

Any NIZK protocol Π that satisfies malicious CRS NUZK also satisfies malicious CRS verifiable witness semantic security.

Lemma 4.17 (Malicious CRS NUDZK Implies Malicious CRS Verifiable Witness Semantic Security)**.** *If $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ is a NIZK for a language $L \in \mathsf{NP}$ that satisfies malicious CRS NUDZK, then Π also satisfies malicious CRS verifiable witness semantic security.*

Proof. The proof is completely analogous to the proof of Lemma 4.15 and remains nearly identical. We expound on the differences. In this setting, the distributions considered have support over, along with statements and witnesses, sets of deterministic functions $\mathcal{F}$ for which for every function $f \in \mathcal{F}$, there exists a polynomial-time verification algorithm $V_f(x, y)$ that outputs 1 if and only if $y = f(w)$ (and 0 otherwise) for some witness w for x. Again, for syntax type checking between the distributions in Def. 4.9 and the distributions in Def. 4.13, we can consider the support of the distribution in the form (x, w, Aux') where $\mathsf{Aux}' = (\mathsf{Aux}, f)$ where V_f is given in Aux. In the previous proof, we showed how an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ for the malicious CRS WSS property immediately implied a polynomial sized distinguisher $\mathcal{B}_\tau$ for the malicious CRS NUDZK property. That adversary $\mathcal{B}_\tau$ used a function of the witness denoted by y given to it in the auxiliary input Aux to check whether the output of $\mathcal{A}_2$ is indeed y which would indicate that $\mathcal{B}_\tau$ was non-negligibly more likely to have been given an honestly generated proof. In the VWSS setting, we will use an adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ for the malicious CRS VWSS property and in the same way, $\mathcal{A}'_1$ defines a CRS, CRS^*, for which $\mathcal{A}'_2$ immediately gives a distinguisher for the malicious CRS NUDZK property. This distinguisher certainly differs from that prior construction because no such y is given in Aux' to use for verifying the output y' of $\mathcal{A}_2$. Instead, the distinguisher uses V_f to verify the output y' of $\mathcal{A}'_2$ is a function f of some witness to x. If $V_f(x, y')$ is 1, then the proof given in the distinguishers input is non-negligibly more likely to have been honestly generated. Otherwise, $V_f(x, y') = 0$ and the proof is non-negligibly more likely to have been a simulated proof. The existence of such a distinguisher contradicts the malicious CRS NUDZK property. □

Corollary 4.18 (Malicious CRS NUZK Implies Malicious CRS Verifiable Witness Semantic Security)**.** *If* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *is a NIZK protocol for a language* $L \in \mathsf{NP}$ *that satisfies malicious CRS NUZK, then* Π *also satisfies malicious CRS verifiable witness semantic security.*

Proof. Lemma 4.14 implies the protocol Π is malicious CRS NUDZK, and Lemma 4.17 implies that Π is malicious CRS VWSS. □

4.5 Malicious Reusable CRS Witness Semantic Security

To address sequential and concurrent composition, we introduce the notion of malicious *reusable* CRS witness semantic security. That is, we extend the above security definition—which guarantees the inability of a malicious authority to recover partial information $f(w)$, for a single sampled $(x, w) \in R_L$ from a distribution—to the setting in which a malicious authority has the ability to adaptively query statements x_i and receive proofs π_i in order to learn $f'(w_1, \dots, w_n)$ for an arbitrary deterministic function of interest f'. The witnesses $w_1, \dots, w_n$ can even be correlated. Our security notion captures the hardness of guessing even a single bit of information on the joint distribution of the witnesses.

We will consider the following two experiments where $\mathcal{A}_1$ is computationally unbounded, $\mathcal{A}_2$ is a stateful PPT algorithm, and $\mathcal{B}$ is polynomial sized. The first experiment is as follows.

```
$\mathsf{ReWSS.NoProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}}(1^\lambda)$
  1. $\mathsf{CRS}^*, \mathsf{Aux}_1 \leftarrow \mathcal{A}_1(1^\lambda)$.
  2. $(x_1, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{Aux}_1)$.
  3. $\pi_1 \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_1, w_1)$.
  4. Query phase which repeats for $i \in [n-1]$ where $n$ depends on $\mathcal{A}_2$:
     (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \pi_i)$.
     (b) $\pi_{i+1} \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_{i+1}, w_{i+1})$.
  5. $(\mathsf{Aux}_2, f) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_n)$ where $n$ is the number of queries made.
  6. $y \leftarrow \mathcal{B}(1^\lambda, \mathsf{Aux}_1, \{x_i\}, \mathsf{Aux}_2, f)$.
  7. If $y = f(w_1, \dots, w_n)$, then the value of this experiment is 1. Otherwise, the value is 0.
```

The second experiment differs from the first only in that $\mathcal{B}$ also receives the generated proofs as input. This difference is marked in red below.

```
$\mathsf{ReWSS.WithProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}}(1^\lambda)$
  1. $\mathsf{CRS}^*, \mathsf{Aux}_1 \leftarrow \mathcal{A}_1(1^\lambda)$.
  2. $(x_1, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{Aux}_1)$.
  3. $\pi_1 \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_1, w_1)$.
  4. Query phase which repeats for $i \in [n-1]$ where $n$ depends on $\mathcal{A}_2$:
     (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \pi_i)$.
     (b) $\pi_{i+1} \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_{i+1}, w_{i+1})$.
  5. $(\mathsf{Aux}_2, f) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_n)$ where $n$ is the number of queries made.
  6. $y \leftarrow \mathcal{B}(1^\lambda, \mathsf{Aux}_1, \{x_i\}, \{\pi_i\}, \mathsf{Aux}_2, f)$.
  7. If $y = f(w_1, \dots, w_n)$, then the value of this experiment is 1. Otherwise, the value is 0.
```

Intuitively, the first scenario captures the hardness of learning $f(w_1, \dots, w_n)$ given only the statements $\{x_i\}$. The second scenario captures the hardness of learning $f(w_1, \dots, w_n)$ when given the proofs $\{\pi_i\}$ that were generated with CRS^*. Naturally, we desire that seeing the proofs $\{\pi_i\}$ only makes guessing $f(w_1, \dots, w_n)$ negligibly easier.

Remark 4.19. To be absolutely clear, in both scenarios, the circuit $\mathcal{B}$ aims to recover $f(w_1, \dots, w_n)$ and is not attempting to distinguish between the two experiments. The circuits $\mathcal{B}$ in the two games have different input domains, and are parameterized by different circuits $\mathcal{B}$ (with possibly the same functionality). Another trivial observation is that one can consider an adversary $\mathcal{A}_2$ that places all the witnesses in the auxiliary input Aux_2. In this case, the same such adversary $\mathcal{A}_2$ would behave identically in both games, trivially satisfying Definition 4.20.

Definition 4.20 (Malicious Reusable CRS WSS)**.** *A NIZK protocol* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *satisfies malicious reusable CRS witness semantic security if for all unbounded* $\mathcal{A}_1$*, all stateful polynomial sized* $\mathcal{A}_2$*, for all polynomial sized* $\mathcal{B}$*, there exists an unbounded* $\mathcal{A}_1'$*, a stateful polynomial sized* $\mathcal{A}_2'$*, a polynomial sized* $\mathcal{B}'$ *and a negligible function* $\mu : \mathbb{N} \to \mathbb{N}$ *such that*

$$\Pr\left[\mathsf{ReWSS.WithProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}}(1^\lambda) = 1\right] \leq \Pr\left[\mathsf{ReWSS.NoProofs}_{\mathcal{A}_1',\mathcal{A}_2',\mathcal{B}'}(1^\lambda) = 1\right] + \mu(\lambda).$$

As before, in order to show that our NIZK construction satisfies Definition 4.20, we will define notion of non-uniform simulation in which the simulator can depend on the CRS. We consider the following game-based definition for which we first define two experiments in which $\mathcal{A}_1$ is computationally unbounded and $\mathcal{A}_2$ is stateful and polynomial sized.

```
$\mathsf{ReNUZK.Real}_{\mathcal{A}_1,\mathcal{A}_2}(1^\lambda)$
  1. $\mathsf{CRS}^*, \mathsf{Aux}_1 \leftarrow \mathcal{A}_1(1^\lambda)$.
  2. $(x_1, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{Aux}_1)$.
  3. $\pi_1 \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_1, w_1)$.
  4. Query phase which repeats for $i \in [n-1]$ where $n$ is the number of queries $\mathcal{A}_2$ makes:
     (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \pi_i)$.
     (b) $\pi_{i+1} \leftarrow \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_{i+1}, w_{i+1})$.
  5. $\mathsf{Aux}_2 \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_n)$ where $n$ is the number of queries made.
  6. Output $(\mathsf{CRS}^*, \{x_i\}_{i\in[n]}, \{\pi_i\}_{i\in[n]}, \mathsf{Aux}_2)$.
```

The above experiment captures the real execution of the NIZK protocol in which an adversary can adaptively choose statements to obtain proofs for. The next experiment captures an ideal world in which all proofs are produced by a polynomial sized simulator Sim who depends on the CRS (this dependence is modeled by providing Sim with a trapdoor for the CRS computed by a computationally unbounded entity) and is given only the statements.

```
$\mathsf{ReNUZK.Ideal}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$
  1. $\mathsf{CRS}^*, \mathsf{Aux}_1 \leftarrow \mathcal{A}_1(1^\lambda)$.
  2. $\tau \leftarrow \mathsf{Ext}(\mathsf{CRS}^*)$.
  3. $(x_1, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{Aux}_1)$.
  4. $\tilde{\pi}_1 \leftarrow \mathsf{Sim}(1^\lambda, \tau, x_1)$.
  5. Query phase which repeats for $i \in [n-1]$ where $n$ depends on $\mathcal{A}_2$:
     (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \tilde{\pi}_i)$.
```

```
    (b) $\tilde{\pi}_{i+1} \leftarrow \mathsf{Sim}(1^\lambda, \tau, x_{i+1})$.
6. $\mathsf{Aux}_2 \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_n)$ where $n$ is the number of queries made.
7. Output $(\mathsf{CRS}^*, \{x_i\}_{i\in[n]}, \{\tilde{\pi}_i\}_{i\in[n]}, \mathsf{Aux}_2)$.
```

Definition 4.21 (Malicious Reusable CRS NUZK)**.** *A NIZK protocol* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *satisfies malicious reusable CRS non-uniform zero-knowledge with auxiliary information if there exists a computationally unbounded extractor* Ext *and there exists a polynomial sized* Sim *such that for all computationally unbounded* $\mathcal{A}_1$ *and for all stateful polynomial sized* $\mathcal{A}_2$*, the following holds:*

$$\mathsf{ReNUZK.Real}_{\mathcal{A}_1,\mathcal{A}_2}(1^\lambda) \approx_c \mathsf{ReNUZK.Ideal}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda).$$

Lemma 4.22 (Malicious CRS NUZK Implies Malicious Reusable CRS NUZK)**.** *If a NIZK protocol* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *satisfies malicious CRS non-uniform zero-knowledge with auxiliary information (Def. 4.12), then* Π *satisfies malicious reusable CRS non-uniform zero-knowledge with auxiliary information (Def. 4.21).*

Proof. We proceed by a proof by contradiction. Suppose for sake of contradiction that Π does not satisfy malicious reusable CRS non-uniform zero-knowledge with auxiliary information (Def. 4.21). Then we will show the existence of a string CRS^*, a string Aux, and a statement x^* with witness w^* such that for all polynomial sized $\mathsf{Sim}_{\mathsf{CRS}^*}$, the tuple $(x^*, \mathsf{CRS}^*, \mathsf{Prove}(\mathsf{CRS}^*, x^*, w^*), \mathsf{Aux})$ is computationally distinguishable from $(x^*, \mathsf{CRS}^*, \mathsf{Sim}_{\mathsf{CRS}^*}(x^*), \mathsf{Aux})$, contradicting Def. 4.12.

In particular, if $n = n(\lambda)$ is the number of queries made by $\mathcal{A}_2$, we consider the series of hybrids for $j \in \{0, \ldots, n\}$ where $\mathsf{ReNUZK}.\mathcal{H}^{(0)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda) \equiv \mathsf{ReNUZK.Real}_{\mathcal{A}_1,\mathcal{A}_2}(1^\lambda)$ and $\mathsf{ReNUZK}.\mathcal{H}^{(n)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda) \equiv \mathsf{ReNUZK.Ideal}_{\mathcal{A}_1,\mathcal{A}_2}(1^\lambda)$ and where $\mathsf{ReNUZK}.\mathcal{H}^{(j)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)$ is such that the last j proofs are produced by the simulator and the first $n - j$ proofs are produced by $\mathsf{Prove}(\cdot,\cdot,\cdot)$. More formally,

```
$\mathsf{ReNUZK}.\mathcal{H}^{(j)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$
1. $\mathsf{CRS}^*, \mathsf{Aux}_1 \leftarrow \mathcal{A}_1(1^\lambda)$.
2. $\tau \leftarrow \mathsf{Ext}(\mathsf{CRS}^*)$.
3. $(x_1, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{Aux}_1)$.
4. $\pi_1 \leftarrow \begin{cases} \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_1, w_1) & \text{if } j < n \\ \mathsf{Sim}(1^\lambda, \tau, x_1) & \text{if } j = n \end{cases}$.
5. Query phase which repeats for $i \in [n-1]$ where $n$ depends on $\mathcal{A}_2$:
    (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \pi_i)$.
    (b) $\pi_{i+1} \leftarrow \begin{cases} \mathsf{Prove}(1^\lambda, \mathsf{CRS}^*, x_{i+1}, w_{i+1}) & \text{if } i \le n - j + 1 \\ \mathsf{Sim}(1^\lambda, \tau, x_{i+1}) & \text{o.w.} \end{cases}$.
6. $\mathsf{Aux}_2 \leftarrow \mathcal{A}(1^\lambda, \mathsf{st}_n)$ where $n$ is the number of queries made.
7. Output $(\mathsf{CRS}^*, \{x_i\}_{i\in[n]}, \{\pi_i\}_{i\in[n]}, \mathsf{Aux}_2)$.
```

Observe that the presence of the extractor in the experiment $\mathsf{ReNUZK}.\mathcal{H}^{(0)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)$ does not affect the distribution at all, existing only for syntactical reasons, so $\mathsf{ReNUZK}.\mathcal{H}^{(0)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda) \equiv \mathsf{ReNUZK.Real}_{\mathcal{A}_1,\mathcal{A}_2}(1^\lambda)$. By our assumption made for the sake of contradiction, for any choice of polynomial sized simulator, Sim, there exists some index $j^* \in \{0, \ldots, n-1\}$ for which there exists a constant c and a distinguisher $\mathcal{D}_{j^*,\mathsf{Sim}}$

such that

$$\left| \Pr_{\mathcal{D}_{j^*,\mathsf{Sim}},\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}}\left[\mathcal{D}_{j^*,\mathsf{Sim}}\left(\mathsf{ReNUZK}.\mathcal{H}^{(j^*)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)\right)=1\right] - \Pr_{\mathcal{D}_{j^*,\mathsf{Sim}},\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}}\left[\mathcal{D}_{j^*,\mathsf{Sim}}\left(\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)\right)=1\right] \right| \geq \frac{1}{\lambda^c}.$$

Then by an averaging argument, there must exist some partial transcript

$$\mathsf{trans} := \left(\overline{\mathsf{CRS}^*,\mathsf{Aux}_1},\overline{\tau},r_{\mathcal{A}_2},\overline{(x_1,w_1,\mathsf{st}_1)},\overline{\pi_1},\dots,\overline{(x_{j^*},w_{j^*},\mathsf{st}_{j^*})},\overline{\pi_{j^*}}\right),$$

for which the following two experiments $\mathsf{ReNUZK}.\mathcal{H}^{(j^*),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$ and $\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$ are distinguishable with non-negligible advantage, where $r_{\mathcal{A}_2}$ denotes some fixed random coins of $\mathcal{A}_2$. The first experiment is one that fixes the random coins of $\mathcal{A}_2$ to be $r_{\mathcal{A}_2}$ and replaces the few rounds of messages in $\mathsf{ReNUZK}.\mathcal{H}^{(j^*)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$ with the fixed partial transcript which we highlight in red.

```
$\mathsf{ReNUZK}.\mathcal{H}^{(j^*),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$

  1. $\overline{\mathsf{CRS}^*,\mathsf{Aux}_1}$.
  2. $\overline{\tau}$.
  3. $\overline{(x_1,w_1,\mathsf{st}_1)}$.
  4. $\overline{\pi_1}$.
  5. For $1 \leq i \leq (n-j^*)$: $\overline{(x_i,w_i,\mathsf{st}_i)},\overline{\pi_i}$.
  6. $(x_{n-j^*+1},w_{n-j^*+1},\mathsf{st}_{n-j^*+1}) \leftarrow \mathcal{A}_2(1^\lambda,\overline{\mathsf{st}_{n-j^*}},\overline{\pi_{n-j^*}})$.
  7. $\pi_{n-j^*+1} \leftarrow \mathsf{Sim}(1^\lambda,\overline{\tau},\overline{x_{n-j^*+1}})$.
  8. The remaining query phase which repeats for $i \in \{n-j^*+2,\dots,n-1\}$ where $n$ depends on $\mathcal{A}_2$:
       (a) $(x_{i+1},w_{i+1},\mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda,\mathsf{st}_i,\pi_i)$.
       (b) $\pi_{i+1} \leftarrow \mathsf{Sim}(1^\lambda,\overline{\tau},x_{i+1})$.
  9. $\mathsf{Aux}_2 \leftarrow \mathcal{A}_2(1^\lambda,\mathsf{st}_n)$ where $n$ is the number of queries made.
 10. Output $(\overline{\mathsf{CRS}^*},(\overline{x_1},\dots,\overline{x_{n-j^*}},x_{n-j^*+1},\dots,x_n),(\overline{\pi_1},\dots,\overline{\pi_{n-j^*}},\pi_{n-j^*+1},\dots,\pi_n),\mathsf{Aux}_2)$.
```

The second experiment is one that that fixes the random coins of $\mathcal{A}_2$ to be $r_{\mathcal{A}_2}$ and replaces the few rounds of messages in $\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1)}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$ with the fixed partial transcript.

```
$\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$

  1. $\overline{\mathsf{CRS}^*,\mathsf{Aux}_1}$.
  2. $\overline{\tau}$.
  3. $\overline{(x_1,w_1,\mathsf{st}_1)}$.
  4. $\overline{\pi_1}$.
  5. For $1 \leq i \leq (n-j^*)$: $\overline{(x_i,w_i,\mathsf{st}_i)},\overline{\pi_i}$
  6. $(x_{n-j^*+1},w_{n-j^*+1},\mathsf{st}_{n-j^*+1}) \leftarrow \mathcal{A}_2(1^\lambda,\overline{\mathsf{st}_{n-j^*}},\overline{\pi_{n-j^*}})$.
  7. $\pi_{n-j^*+1} \leftarrow \mathsf{Prove}(1^\lambda,\overline{\mathsf{CRS}^*},\overline{x_{n-j^*+1}},\overline{w_{n-j^*+1}})$.
```

8. The remaining query phase which repeats for $i \in \{n - j^* + 2, \ldots, n - 1\}$ where n depends on $\mathcal{A}_2$:
   (a) $(x_{i+1}, w_{i+1}, \mathsf{st}_{i+1}) \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_i, \pi_i)$.
   (b) $\pi_{i+1} \leftarrow \mathsf{Sim}(1^\lambda, \overline{\tau}, x_{i+1})$.
9. $\mathsf{Aux}_2 \leftarrow \mathcal{A}_2(1^\lambda, \mathsf{st}_n)$ where n is the number of queries made.
10. Output $(\overline{\mathsf{CRS}^*}, (\overline{x_1}, \ldots, \overline{x_{n-j^*}}, x_{n-j^*+1}, \ldots, x_n), (\overline{\pi_1}, \ldots, \overline{\pi_{n-j^*}}, \pi_{n-j^*+1}, \ldots, \pi_n), \mathsf{Aux}_2)$.

Note in both experiments $\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$ and $\mathsf{ReNUZK}.\mathcal{H}^{(j^*),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$, the statement-witness pair $(x_{n-j^*+1}, w_{n-j^*+1})$ is identically sampled as the prior partial transcript, which includes the coins of $\mathcal{A}_2$, is fixed and identical in both experiments. We will denote the PPT distinguisher between these two experiments as $\mathcal{D}_{j^*,\mathsf{Sim},\mathsf{fixed}}$. That is, there exists a constant c' such that

$$\Bigg| \Pr_{\mathcal{D}_{j^*,\mathsf{Sim},\mathsf{fixed}},\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}} \left[\mathcal{D}_{j^*,\mathsf{Sim},\mathsf{fixed}} \left(\mathsf{ReNUZK}.\mathcal{H}^{(j^*),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)\right) = 1\right]$$
$$- \Pr_{\mathcal{D}_{j^*,\mathsf{Sim},\mathsf{fixed}},\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}} \left[\mathcal{D}_{j^*,\mathsf{Sim},\mathsf{fixed}} \left(\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1),,\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Ext},\mathsf{Sim}}(1^\lambda)\right) = 1\right] \Bigg| \geq \frac{1}{\lambda^{c'}}.$$

For any polynomial sized $\mathsf{Sim}_{\overline{\mathsf{CRS}^*}}$ we can now construct a PPT distinguisher $\mathcal{D}_{\mathsf{NUZK}}$ that distinguishes between

$$(\overline{\mathsf{CRS}^*}, x_{n-j^*+1}, \mathsf{Prove}(1^\lambda, \overline{\mathsf{CRS}^*}, x_{n-j^*+1}, w_{n-j^*+1}), \mathsf{trans})$$

and

$$(\overline{\mathsf{CRS}^*}, x_{n-j^*+1}, \mathsf{Sim}_{\overline{\mathsf{CRS}^*}}(1^\lambda, x_{n-j^*+1}), \mathsf{trans}).$$

$\mathcal{D}_{\mathsf{NUZK}}$ takes as input $(\overline{\mathsf{CRS}^*}, x_{n-j^*+1}, \tilde{\pi}, \mathsf{trans})$ and can efficiently construct the experiment above where in Step 7, it sets $\pi_{n-j^*+1} \leftarrow \tilde{\pi}$, because $\mathcal{D}_{\mathsf{NUZK}}$ has hardcoded the partial transcript trans that includes the trapdoor $\overline{\tau}$ so can run Sim itself.

If the input proof is generated by the real prove algorithm, that is $\tilde{\pi} \leftarrow \mathsf{Prove}(1^\lambda, \overline{\mathsf{CRS}^*}, x_{n-j^*+1}, w_{n-j^*+1})$, then $\mathcal{D}_{\mathsf{NUZK}}$ has sampled exactly from $\mathsf{ReNUZK}.\mathcal{H}^{(j^*+1),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}_{\overline{\mathsf{CRS}^*}},\mathsf{Ext}}(1^\lambda)$. Otherwise, the input proof was from the simulator, and $\tilde{\pi} \leftarrow \mathsf{Sim}_{\overline{\mathsf{CRS}^*}}(1^\lambda, x_{n-j^*+1})$ so $\mathcal{D}_{\mathsf{NUZK}}$ has sampled exactly[6] from $\mathsf{ReNUZK}.\mathcal{H}^{(j^*),\mathsf{fixed}}_{\mathcal{A}_1,\mathcal{A}_2,\mathsf{Sim}_{\overline{\mathsf{CRS}^*}},\mathsf{Ext}}(1^\lambda)$. Therefore, $\mathcal{D}_{\mathsf{NUZK}}$ can run $\mathcal{D}_{j^*,\mathsf{Sim}_{\overline{\mathsf{CRS}^*}},\mathsf{fixed}}$ on the constructed experiment instance and the distinguishing advantage of $\mathcal{D}_{\mathsf{NUZK}}$ is therefore at least $\lambda^{-c'}$, contradicting the assumption that Π satisfies malicious CRS non-uniform zero-knowledge with auxiliary information (Def. 4.12). □

Lemma 4.23 (Malicious Reusable CRS NUZK Implies Malicious Reusable CRS WSS)**.** *If a NIZK protocol $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ satisfies malicious reusable CRS non-uniform zero-knowledge with auxiliary information (Def. 4.21), then Π satisfies malicious reusable CRS witness semantic security (Def. 4.20).*

Proof. The intuition for the proof is straightforward and is entirely analogous to the proof of Lemma 4.15. If there exists a polynomial sized $\mathcal{B}$ that is at a non-negligible disadvantage of recovering $f(w_1, \ldots, w_n)$ when not given the proofs, then $\mathcal{B}$ can be used to distinguish between the real and ideal worlds by checking if $\mathcal{B}$'s output is $f(w_1, \ldots, w_n)$. The only remaining argument is how to give $f(w_1, \ldots, w_n)$ to the distinguisher, and this can be simply addressed by giving the distinguisher $w_1, \ldots, w_n$ and f in the auxiliary input.

A formal proof proceeds as follows. Let computationally unbounded Ext and polynomial sized Sim exist per the definition of malicious reusable CRS non-uniform zero-knowledge with auxiliary information (Def. 4.21) where their existence is universal (independent of the adversaries) by definition. Suppose for sake of contradiction that Π does not satisfy malicious reusable CRS witness semantic security (Def. 4.20). Then there exists a computationally unbounded $\mathcal{A}_1$, a stateful polynomial sized $\mathcal{A}_2$, a polynomial sized $\mathcal{B}$ such that for all computationally unbounded $\mathcal{A}'_1$, stateful polynomial sized $\mathcal{A}'_2$, and polynomial sized $\mathcal{B}'$, there exists a polynomial p such that for all $N \in \mathbb{N}$ there exists a $\lambda > N$ such that

$$\Pr\left[\mathsf{ReWSS}.\mathsf{WithProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}}(1^\lambda) = 1\right] \geq \Pr\left[\mathsf{ReWSS}.\mathsf{NoProofs}_{\mathcal{A}'_1,\mathcal{A}'_2,\mathcal{B}'}(1^\lambda) = 1\right] + \frac{1}{p(\lambda)} \quad (3)$$

[6] For syntax, one can think of $\mathsf{Sim}_{\overline{\mathsf{CRS}^*}}$ ignoring the trapdoor τ as it already has one built into it.

The existence of such $\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}$ almost directly contradicts malicious reusable CRS non-uniform zero-knowledge with auxiliary information (Def. 4.21) because when $\mathcal{B}$ is given honestly generated proofs, it correctly outputs $f(w_1, \dots, w_n)$ with probability given by the LHS of Equation 3 and when $\mathcal{B}$ is given simulated proofs, its equivalently a circuit $\mathcal{B}'$ that only used the statements $\{x_i\}_{i\in[n]}$ and the auxiliary inputs to attempt to recover $f(w_1, \dots, w_n)$ and succeeds with probability given by the RHS of Equation 3. The remaining issue is that a distinguisher needs to know if it has successfully recovered $f(w_1, \dots, w_n)$.

In order to give the distinguisher this value of $f(w_1, \dots, w_n)$ to compare and verify the output of $\mathcal{B}$ with, we consider a new stateful polynomial sized $\mathcal{A}_2^*$ that behaves exactly like $\mathcal{A}_2$ except that it additionally stores all queried witnesses $\{w_i\}$, Aux_1, and f in Aux_2^*, which it outputs in Step 6 of the experiments defined in Def. 4.21. Then, by additionally defining $\mathcal{A}_1^*$ to be an unbounded algorithm defined to be exactly $\mathcal{A}_1$, we claim that

$$\mathsf{ReNUZK.Real}_{\mathcal{A}_1^*,\mathcal{A}_2^*}(1^\lambda) \not\approx_c \mathsf{ReNUZK.Real}_{\mathcal{A}_1^*,\mathcal{A}_2^*,\mathsf{Sim},\mathsf{Ext}}(1^\lambda)$$

for infinitely many values of λ. The distinguisher is given by a polynomial sized D:

$D(1^\lambda, \mathsf{CRS}^*, \{x_i\}_{i\in[n]}, \{\pi_i\}_{i\in[n]}, \mathsf{Aux}_2^*)$:

1. Parse Aux_2^* as $(\{w_i\}_{i\in[n]}, f, \mathsf{Aux}_1, \mathsf{Aux}_2)$.
2. Compute $y \leftarrow \mathcal{B}(1^\lambda, \mathsf{Aux}_1, \{x_i\}, \{\pi_i\}_{i\in[n]}, \mathsf{Aux}_2, f)$.
3. If $y = f(w_1, \dots, w_n)$, then output 0 where 0 indicates a guess for the real world. Otherwise output 1 where 1 indicates a guess for the ideal world.

For the aforementioned values of λ for which Equation 3 hold, we consider two cases. In the first case, if the π_i are produced by the simulator, then computing Step 2 of D is equivalent to computing some polynomial sized circuit of the form $\mathcal{B}'(1^\lambda, \mathsf{Aux}_1, \{x_i\}_{i\in[n]}, \mathsf{Aux}_2, f)$ where $\mathcal{B}'$ first simulates the proofs itself and then runs $\mathcal{B}$, so the success probability of computing $y = f(w_1, \dots, w_n)$ is given by $\Pr\left[\mathsf{ReWSS.NoProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}'}(1^\lambda) = 1\right]$. In the second case, if the π_i are produced by the honest prover, then the success probability Step 2 of computing $y = f(w_1, \dots, w_n)$ is given by $\Pr\left[\mathsf{ReWSS.NoProofs}_{\mathcal{A}_1,\mathcal{A}_2,\mathcal{B}}(1^\lambda) = 1\right]$. Then, Equation 3 guarantees that the success probability in the second case is non-negligibly greater than that of the first case for all such values of λ. The existence of such a distinguisher contradicts malicious CRS reusable NUZK (Def. 4.21) and we conclude that Π must satisfy malicious reusable CRS witness semantic security (Def. 4.20). □

4.5.1 Malicious Reusable CRS Verifiable Witness Semantic Security

One can also consider an analogous extension of Def. 4.7 where we allow the adversary to adaptively query statements x_i and receive proofs π_i in order to learn y such that $y = f(w_1, \dots, w_n)$ for some valid witnesses $w_1, \dots, w_n$. As in Def. 4.7, we require the existence of a verification algorithm $V_f(x_1, \dots, x_n, y)$ that outputs 1 if and only if $y = f(w_1, \dots, w_n)$ for some valid witnesses $w_1, \dots, w_n$. Our NIZK will also satisfy malicious reusable CRS verifiable witness semantic security, since it can be shown that malicious reusable CRS NUZK implies malicious reusable CRS verifiable witness semantic security via an analogous proof to Lemma 4.23.

5 NIZK with Malicious CRS Witness Semantic Security from LWE

In this section, we construct a NIZK that additionally satisfies malicious CRS witness semantic security, malicious CRS verifiable witness semantic security, and malicious reusable CRS witness semantic security, assuming the subexponential hardness of LWE. We show the following main theorem and its immediate corollary.

Theorem 5.1 (NIZK with Malicious CRS NUZK from LWE)**.** *Assuming the subexponential hardness of LWE, there exists a NIZK argument* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *such that* Π *additionally satisfies malicious CRS non-uniform zero knowledge with auxiliary information (malicious CRS NUZK).*

Corollary 5.2. *Assuming the subexponential hardness of LWE, there exists a NIZK argument* $\Pi = (\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *such that* Π *additionally satisfies malicious CRS witness semantic security, malicious CRS verifiable witness semantic security, and malicious reusable CRS witness semantic security.*

Proof. This immediately follows from Theorem 5.1, Corollary 4.16, Corollary 4.18, and Lemma 4.23. □

Corollary 5.3. *Assuming the subexponential hardness of LWE, there exists two-round public-coin publicly-verifiable argument* (P, V) *in the plain model that satisfies witness semantic security and verifiable witness semantic security. Moreover, the first round message is reusable in that it preserves witness semantic security.*

Proof. This follows from Corollary 5.2, Lemma 4.6 and Lemma 4.10. □

To build our NIZK argument and show Theorem 5.1, we require the subexponential hardness of the SIS problem and two cryptographic primitives—perfectly binding non-interactive commitments and adaptively sound computational ZAPs—obtained from the subexponential hardness of the LWE problem. Therefore, a single hardness assumption, the subexponential hardness of the LWE suffices to imply the security guarantees of our NIZK argument.

5.1 Building Blocks

We require the following building blocks and assumptions for our construction:

1. Let $\lambda \in \mathbb{N}$ be a security parameter. For appropriate parameters n, m, q, β, we assume the subexponential hardness of the $\mathsf{SIS}_{n,m,q,\beta}$-problem against non-uniform adversaries of size 2^{λ^ϵ} for some constant $\epsilon \in (0, 1)$ and we assume that $\mathsf{SIS}_{n,m,q,\beta}$ is broken against 2^λ-sized adversaries.

2. Let Com be a perfectly binding non-interactive commitment scheme parameterized by a security parameter $\tilde{\lambda} \in \mathbb{N}$. We assume that Com is hiding against $2^{\tilde{\lambda}^\epsilon}$-sized adversaries and broken against $2^{\tilde{\lambda}}$-sized adversaries. This type of commitment can be obtained from LWE-based PKEs as mentioned in Remark 3.14.

3. Let $(\mathsf{ZAP}_1, \mathsf{ZAP}_2, \mathsf{ZAP}_V)$ be an adaptively computationally sound ZAP satisfying computational witness indistinguishability against all polynomial sized adversaries. This is obtained from subexponential LWE in Appendix B.

Definition 5.4 (SIS Parameter Generator)**.** *We define a parameter generator algorithm* $\mathsf{ParamGen}(1^\lambda)$ *for* $\lambda \in \mathbb{N}$ *that outputs* $n = n(\lambda), m = m(\lambda), q = q(\lambda), \beta = \beta(\lambda)$ *such that (1)* $n > 2m \log q$, $\beta > \sqrt{m}$ *(2) the* $\mathsf{SIS}_{n,m,q,\beta}$*-problem is secure against* 2^{λ^ϵ} *time adversaries for some constant* $\epsilon \in (0, 1)$*, and (3) broken against* 2^λ *time adversaries.*

Remark 5.5. (Complexity Leveraging Summary) Suppose we generate $n, m, q, \beta \leftarrow \mathsf{ParamGen}(1^\lambda)$ where the subexponential hardness involves some constant $\epsilon \in (0, 1)$. Then we have that $\mathsf{SIS}_{n,m,q,\beta}$ problem is secure against $T_{\mathsf{SIS}}(\lambda) \triangleq 2^{\lambda^\epsilon}$ sized adversaries and broken against 2^λ sized adversaries. Let $\tilde{\lambda} = \lambda^{\epsilon'}$ for some constant $\epsilon' = \epsilon/2$. Instantiating the commitment scheme with security parameter $\tilde{\lambda}$, we see that the commitment scheme is secure against adversaries of size $T_{\mathsf{Com}}(\lambda) \triangleq 2^{\tilde{\lambda}^\epsilon} = 2^{\lambda^{\epsilon \cdot (\epsilon/2)}}$ and broken against adversaries of size $T_{\mathsf{Extract}}(\lambda) \triangleq 2^{\tilde{\lambda}} = 2^{\lambda^{\epsilon/2}}$. Therefore, we have

$$T_{\mathsf{Com}} \ll T_{\mathsf{Extract}} \ll T_{\mathsf{SIS}}$$

Notation Throughout the construction, the notation $\mathsf{Com}(x; r)$ implicitly denotes a commitment instantiated with security parameter $\tilde{\lambda} \in \mathbb{N}$: $\mathsf{Com}(1^{\tilde{\lambda}}, x; r)$. As mentioned in a previous remark, the commitment to a string is performed bit-by-bit.

5.2 The Construction

Protocol 1 (NIZK with Malicious CRS (Verifiable) Witness Semantic Security)**.**

Parameters: Let L be a language in NP. Generate $n, m, q, \beta \leftarrow \mathsf{ParamGen}(1^\lambda)$ for constant $\epsilon \in (0, 1)$. Set $\tilde{\lambda} = \lambda^{\epsilon'}$ for constant $\epsilon' = \epsilon/2$. Instantiate the commitment with security parameter $\tilde{\lambda}$. Then we have the following:

$\mathsf{GenCRS}(1^\lambda)$:

1. $z \leftarrow \mathsf{ZAP}_1(1^\lambda)$.
2. Choose a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
3. Output $\mathsf{CRS} = (\mathsf{CRS}_0, \mathsf{CRS}_1)$ where $\mathsf{CRS}_0 := \mathbf{A}$ and $\mathsf{CRS}_1 := z$.

$\mathsf{Prove}(\mathsf{CRS}, x, w)$:

1. Parse CRS as $(\mathsf{CRS}_0, \mathsf{CRS}_1)$. Let $\mathbf{A} := \mathsf{CRS}_0$.
2. $c \leftarrow \mathsf{Com}(\mathbf{0}; R)$ for uniformly chosen randomness R.
3. Define statement $S_{\mathsf{CRS}_0,x,c}$ as $\exists \mathbf{r}, R, w'$ such that (a) OR (b) holds where
   (a) $c = \mathsf{Com}(\mathbf{r}; R)$
   $\wedge\ \mathbf{r} \in \{-1, 0, 1\}^m$
   $\wedge\ \mathbf{r} \neq \mathbf{0}$
   $\wedge\ \mathbf{A}\mathbf{r} \equiv \mathbf{0} \mod q$
   (b) $(x, w') \in R_L$.

   Observe that a witness is a triple of the form $(\mathbf{r}, R, w')$. The set of statements that have a witness define a language L'.
4. $\pi_z \leftarrow \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\perp, \perp, w))$
5. Output $\pi \leftarrow (c, \pi_z)$.

$\mathsf{Verify}(\mathsf{CRS}, x, \pi)$:

1. Parse CRS as $(\mathsf{CRS}_0, \mathsf{CRS}_1)$ and parse π as (c, π_z).
2. Output $\mathsf{ZAP}_V(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, \pi_z)$.

Remark 5.6. Perfectly binding commitments are required in the above construction to ensure that the statements for the new language constructed are not trivially true.

We begin by showing that the above construction satisfies the properties of a NIZK.

Completeness Completeness immediately follows from the completeness of ZAP.

Soundness

Lemma 5.7. *The NIZK construction satisfies soundness.*

Proof. Let $x \notin L$. We argue soundness in two steps. First, we show that the probability a polynomial-sized adversary can produce a proof that does not contain a commitment c that satisfies property (a) for the statement $S_{\mathsf{CRS}_0,x,c}$, as defined above in the protocol, yet convinces the verifier to accept, is negligible. This first step uses the adaptive computational soundness of the ZAP. Assuming the first step, the second step is a standard complexity leveraging argument. Namely, if there exists a polynomial-sized adversary that produces a proof that convinces the verifier to accept with non-negligible probability, then there exists a polynomial-sized adversary that breaks the $\mathsf{SIS}_{n,m,q,\beta}$-problem with non-negligible probability.

1. In the first step, assume the polynomial-sized adversary P^* provides a proof π to the honest verifier V that does not contain a commitment c that satisfies property (a) as defined above in the protocol.

   Lemma 5.8. *Suppose $x \notin L$. Let $\phi_{\mathsf{CRS}_0}(\cdot)$ be a predicate that takes a proof π and outputs* 1 *if and only if π contains a commitment c that satisfies condition (a), defined in the protocol above, and $\phi_{\mathsf{CRS}_0}(\cdot)$ outputs* 0 *otherwise. For any polynomial-sized adversary P^*,*

   $$\Pr_{\substack{(\mathsf{CRS}_0,\mathsf{CRS}_1)\leftarrow\mathsf{GenCRS}(1^\lambda)\\ \pi^*\leftarrow P^*(1^\lambda,(\mathsf{CRS}_0,\mathsf{CRS}_1),x)\\ \phi_{\mathsf{CRS}_0}(\pi^*)=0}}[\mathsf{Verify}((\mathsf{CRS}_0,\mathsf{CRS}_1),x,\pi^*)=1]=\mu(\lambda)$$

   where μ is a negligible function.

   Proof. Fix $x \notin L$. Let P^* be an adversary the breaks the computational soundness of Protocol 1 so that

   $$\Pr_{\substack{(\mathsf{CRS}_0,\mathsf{CRS}_1)\leftarrow\mathsf{GenCRS}(1^\lambda)\\ \pi^*\leftarrow P^*(1^\lambda,(\mathsf{CRS}_0,\mathsf{CRS}_1),x)\\ \phi_{\mathsf{CRS}_0}(\pi^*)=0}}[\mathsf{Verify}((\mathsf{CRS}_0,\mathsf{CRS}_1),x,\pi^*)=1]=\rho(\lambda)$$

   for some non-negligible function ρ. We construct an adversary B that breaks the adaptive computational soundness of the ZAP with probability determined by ρ. B takes as input security parameter 1^λ and a first-round message CRS_1 generated by the ZAP verifier. B samples a uniform random matrix $\mathbf{A}$ from $\mathbb{Z}_q^{n\times m}$ and runs $P^*\left(1^\lambda,(\mathsf{CRS}_0,\mathsf{CRS}_1),x\right)$, where $\mathsf{CRS}_0 \triangleq \mathbf{A}$ to obtain a proof π^*. B parses π^* as (c^*,π_z^*). B now chooses an instance S_{CRS_0,x,c^*} and outputs S_{CRS_0,x,c^*} and π_z^*. By the initial assumption on predicate ϕ_{CRS_0}, the condition (a) for S_{CRS_0,x,c^*} is false. Moreover $x \notin L$ was assumed to be false, so both conditions (a) and (b) defined in Protocol 1 do not hold for the instance S_{CRS_0,x,c^*}. Therefore, $S_{\mathsf{CRS}_0,x,c^*} \notin L'$. Then, by construction of Verify in Protocol 1, $\mathsf{Verify}(\mathsf{CRS},x,\pi^*)=1$ if and only if $\mathsf{ZAP}_V(\mathsf{CRS}_1,S_{\mathsf{CRS}_0,x,c^*},\pi_z^*)=1$. Therefore B produces an instance $S_{\mathsf{CRS}_0,x,c^*}\notin L'$ and a proof π_z^* such that $\mathsf{ZAP}_V(\mathsf{CRS}_1,S_{\mathsf{CRS}_0,x,c^*},\pi_z^*)=1$ with probability at least $\rho(\lambda)$, a contradiction to the adaptive soundness of the ZAP. □

2. In the second step, we apply complexity leveraging to show that any polynomial-sized adversary P^* that succeeds in convincing the verifier can be used to construct an algorithm that breaks the $\mathsf{SIS}_{n,m,q,\beta}$-problem in time $O(T_{\mathsf{Extract}}(\lambda)+p(\lambda))$ for some polynomial p.

   Lemma 5.9. *For $x \notin L$ and for any polynomial-sized adversary P^*,*

   $$\Pr_{\substack{(\mathsf{CRS}_0,\mathsf{CRS}_1)\leftarrow\mathsf{GenCRS}(1^\lambda)\\ \pi^*\leftarrow P^*(1^\lambda,(\mathsf{CRS}_0,\mathsf{CRS}_1),x))}}[\mathsf{Verify}((\mathsf{CRS}_0,\mathsf{CRS}_1),x,\pi^*)=1]=\mu(\lambda)$$

   where μ is a negligible function.

   Proof. By Lemma 5.8, if P^* outputs π^* that does not contain a commitment that satisfies condition (a), then P^*'s probability of fooling the verification algorithm is negligible in λ. Therefore if P^* fools the verifier with non-negligible probability $\rho(\lambda)$ it must fool the verifier with non-negligible probability conditioned on the assumption that proof π^* contains a commitment that satisfies condition (a). Moreover, P^* fooling the verifier with non-negligible probability implies that P^* outputs a proof that satisfies condition (a) with non-negligible probability. One observes this fact formally through the law of total probability applied on the formal soundness condition.

   We now construct a time T_{Extract} adversary $\mathcal{A}$ that finds a short integer solution with non-negligible probability $\rho(\lambda)-\mathsf{negl}(\lambda)$. $\mathcal{A}$ takes as input a matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ and is tasked with producing a short $\mathbf{x}$, $\|\mathbf{x}\|\le\beta$, such that $\mathbf{Ax}\equiv\mathbf{0} \bmod q$. $\mathcal{A}$ chooses a random ZAP_1 message z according to the ZAP verifier's first message distribution and runs P^* with $(\mathsf{CRS}_0,\mathsf{CRS}_1)$ as the CRS, where $\mathsf{CRS}_0\triangleq\mathbf{A}$ and $\mathsf{CRS}_1\triangleq z$, and $x\notin L$ as input. P^* responds with a message π. Without loss of generality, we parse π as a commitment and ZAP proof, (c,π_z). $\mathcal{A}$ checks that $\mathsf{Verify}(1^\lambda,x,\pi)=1$. When $\mathsf{Verify}(1^\lambda,x,\pi)=1$, as noted above due to Lemma 5.8, with non-negligible probability, c is a commitment to $\mathbf{r} \in \{-1, 0, 1\}^m$ such that $\mathbf{r} \neq \mathbf{0}$, and $\mathbf{A}\mathbf{r} \equiv \mathbf{0} \mod q$. Since $\|\mathbf{r}\|_2 \leq \sqrt{m} < \beta$, $\mathbf{r}$ is a solution to the $\mathsf{SIS}_{n,m,q,\beta}$-problem. By assumption there exists a $T_{\mathsf{Extract}}(\lambda)$-time algorithm that breaks the commitment hiding property and recovers some $\mathbf{r}$ such that condition (a) is satisfied. This construction of $\mathcal{A}$ is a contradiction to the hardness of the $\mathsf{SIS}_{n,m,q,\beta}$-problem against $O(T_{\mathsf{SIS}}(\lambda))$-time uniform adversaries. Therefore no such P^* can exist. □

The two steps conclude the proof of soundness since Lemma 5.9 directly proves computational soundness. □

Adaptive Computational Zero-Knowledge

Lemma 5.10. *Protocol 1 satisfies adaptive computational zero-knowledge.*

Proof. We construct a PPT algorithm Sim_1 and Sim_2 as follows:

$\mathsf{Sim}_1(1^\lambda)$:

1. Choose matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ in the following manner: choose uniform random matrix $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-1)}$ and uniform random vector $\mathbf{r}' \in \{-1, 0, 1\}^{m-1}$. Then concatenate the column vector $\mathbf{A}'\mathbf{r}' \in \mathbb{Z}_q^n$ to form matrix $\mathbf{A} \triangleq [\mathbf{A}' \mid \mathbf{A}'\mathbf{r}'] \in \mathbb{Z}_q^{n \times m}$.
2. $\mathsf{CRS}_1 \leftarrow \mathsf{ZAP}_1(1^\lambda)$
3. Let state τ be the randomness used by Sim_1 so that Sim_2 can recompute $\mathbf{r}'$.
4. Output $(\mathsf{CRS}_0, \mathsf{CRS}_1), \tau$.

$\mathsf{Sim}_2(1^\lambda, x, \tau)$:

1. Compute $\mathbf{r} \triangleq (\mathbf{r}', -1) \in \{-1, 0, 1\}^m$ from τ.
2. $c \leftarrow \mathsf{Com}(\mathbf{r}; R)$ for some uniformly chosen randomness R.
3. $\pi_z \leftarrow \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\mathbf{r}, R, \perp))$.
4. Output $\pi \leftarrow (c, \pi_z)$.

We proceed via a series of hybrids. We highlight the changes using the color red below.

$H_{0,\mathcal{A}}(1^\lambda) = \mathsf{Expt}_{Real,\mathcal{A}}(1^\lambda)$:

1. $(\mathsf{CRS}_0, \mathsf{CRS}_1) \leftarrow \mathsf{GenCRS}(1^\lambda)$.
2. $(x, w) \leftarrow \mathcal{A}(1^\lambda, (\mathsf{CRS}_0, \mathsf{CRS}_1))$ such that $(x, w) \in R_L$.
3. $(c, \pi_z) \leftarrow \mathsf{Prove}((\mathsf{CRS}_0, \mathsf{CRS}_1), x, w)$ where $c = \mathsf{Com}(\mathbf{0}; R)$ for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\perp, \perp, w))$.
4. Output $((\mathsf{CRS}_0, \mathsf{CRS}_1), x, \pi)$.

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

1. $(\mathsf{CRS}_0, \mathsf{CRS}_1), \tau \leftarrow \mathsf{Sim}_1(1^\lambda)$ where $\mathsf{CRS}_0 = \mathbf{A}$ is chosen as described in the definition of Sim_1. Define $\mathbf{r} \triangleq (\mathbf{r}', -1) \in \{-1, 0, 1\}^m$ where $\mathbf{r}' \in \{-1, 0, 1\}^{m-1}$ is generated as described in the definition of Sim_1.
2. $(x, w) \leftarrow \mathcal{A}(1^\lambda, (\mathsf{CRS}_0, \mathsf{CRS}_1))$ such that $(x, w) \in R_L$.
3. $(c, \pi_z) \leftarrow \mathsf{Prove}((\mathsf{CRS}_0, \mathsf{CRS}_1), x, w)$ where $c = \mathsf{Com}(\mathbf{0}; R)$ for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\perp, \perp, w))$.

4. Output $((\mathsf{CRS}_0, \mathsf{CRS}_1), x, \pi)$.

$\underline{H_{2,\mathcal{A}}(1^\lambda)}$:

1. $(\mathsf{CRS}_0, \mathsf{CRS}_1), \tau \leftarrow \mathsf{Sim}_1(1^\lambda)$ where $\mathsf{CRS}_0 = \mathbf{A}$ is chosen as described in the definition of Sim_1. Define $\mathbf{r} \triangleq (\mathbf{r}', -1) \in \{-1, 0, 1\}^m$ where $\mathbf{r}' \in \{-1, 0, 1\}^{m-1}$ is generated as described in the definition of Sim_1.
2. $(x, w) \leftarrow \mathcal{A}(1^\lambda, (\mathsf{CRS}_0, \mathsf{CRS}_1))$ such that $(x, w) \in R_L$.
3. $(c, \pi_z) \leftarrow \mathsf{Prove}((\mathsf{CRS}_0, \mathsf{CRS}_1), x, w)$ where $c = \mathsf{Com}(\mathbf{r}; R)$ for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\bot, \bot, w))$.
4. Output $((\mathsf{CRS}_0, \mathsf{CRS}_1), x, \pi)$.

$\underline{H_{3,\mathcal{A}}(1^\lambda) = \mathsf{Expt}_{Sim,\mathcal{A}}(1^\lambda)}$:

1. $(\mathsf{CRS}_0, \mathsf{CRS}_1), \tau \leftarrow \mathsf{Sim}_1(1^\lambda)$ where $\mathsf{CRS}_0 = \mathbf{A}$ is chosen as described in the definition of Sim_1. Define $\mathbf{r} \triangleq (\mathbf{r}', -1) \in \{-1, 0, 1\}^m$ where $\mathbf{r}' \in \{-1, 0, 1\}^{m-1}$ is generated as described in the definition of Sim_1.
2. $(x, w) \leftarrow \mathcal{A}(1^\lambda, (\mathsf{CRS}_0, \mathsf{CRS}_1))$ such that $(x, w) \in R_L$.
3. $(c, \pi_z) \leftarrow \mathsf{Prove}((\mathsf{CRS}_0, \mathsf{CRS}_1), x, w)$ where $c = \mathsf{Com}(\mathbf{r}; R)$ for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1, S_{\mathsf{CRS}_0,x,c}, (\mathbf{r}, R, \bot))$.
4. Output $((\mathsf{CRS}_0, \mathsf{CRS}_1), x, \pi)$.

$H_{0,\mathcal{A}}(1^\lambda) \approx_s H_{1,\mathcal{A}}(1^\lambda)$ follows from the leftover hash lemma. Namely, the matrix-vector pair $(\mathbf{A}', \mathbf{A}'\mathbf{r}') \in \mathbb{Z}_q^{n \times (m-1)} \times \mathbb{Z}_q^n$ produced by Sim_1 is statistically indistinguishable from $(\mathbf{A}', \mathbf{u})$ for a vector $\mathbf{u}$ uniformly sampled from $\mathbb{Z}_q^n$.
$H_{1,\mathcal{A}}(1^\lambda) \approx_c H_{2,\mathcal{A}}(1^\lambda)$ follows from the computational hiding property of the commitment.
Finally, $H_{2,\mathcal{A}}(1^\lambda) \approx_c H_{3,\mathcal{A}}(1^\lambda)$ follows by the computational witness indistinguishability property of the ZAP. □

Malicious CRS NUZK

Lemma 5.11 (Malicious CRS NUZK)**.** *Protocol 1 satisfies the malicious CRS NUZK property.*

Proof. For every λ and $\mathsf{CRS} \in \{0,1\}^*$, we parse the CRS as $(\mathsf{CRS}_0, \mathsf{CRS}_1)$ where $\mathsf{CRS}_0 \in \mathbb{Z}_{q(\lambda)}^{n(\lambda) \times m(\lambda)}$, consider the following PPT algorithm $\mathsf{Sim}_{\mathsf{CRS}}$ which takes an input $x \in \{0,1\}^*$:

$\mathsf{Sim}_{\mathsf{CRS}}(1^\lambda, x)$

1. Parse CRS^* as $(\mathsf{CRS}_0^*, \mathsf{CRS}_1^*)$ where $\mathsf{CRS}_0^* \in \mathbb{Z}_{q(\lambda)}^{n(\lambda) \times m(\lambda)}$.
2. Output (c, π_z) where $c = \mathsf{Com}(\mathbf{r}; R)$ for some randomness R and string $\mathbf{r} \in \{-1, 0, 1\}^m$ is such that the following holds:
   (a) $\mathbf{r} \neq \mathbf{0}$.
   (b) $\mathbf{A}\mathbf{r} \equiv \mathbf{0} \bmod q$.

   and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1^*, S_{\mathsf{CRS}_0^*,x,c}, (\mathbf{r}, R, \bot))$.

Observe that $\mathsf{Sim}_{\mathsf{CRS}}$ depends on CRS and has the required string $\mathbf{r}$ hardcoded.
We proceed by a series of hybrid arguments. We highlight changes using the color red.

$\underline{H_{0,\mathcal{A}}(1^\lambda) = \mathsf{Expt}_{MalAuth,\mathcal{A}}(1^\lambda)}$:

1. $(x, w, \mathsf{Aux}, \mathsf{CRS}^*) \leftarrow \mathcal{A}_1(1^\lambda)$ where $\mathcal{A}_1$ is an computationally unbounded adversary. We parse CRS^* as $(\mathsf{CRS}_0^*, \mathsf{CRS}_1^*)$ where $\mathsf{CRS}_0^* \in \mathbb{Z}_q^{n \times m}$.

2. $(c, \pi_z) \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x, w)$ where $c = \mathsf{Com}(\mathbf{0}; R)$ for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1^*, S_{\mathsf{CRS}_0^*,x,c}, (\perp, \perp, w))$.
3. $b \leftarrow \mathcal{A}_2(\mathsf{CRS}^*, x, (c, \pi_z), \mathsf{Aux})$ where $b \in \{0, 1\}$ and $\mathcal{A}_2$ is a polynomial-sized adversary. The output of the experiment is b.

$\underline{H_{1,\mathcal{A}}(1^\lambda)}$:

1. $(x, w, \mathsf{Aux}, \mathsf{CRS}^*) \leftarrow \mathcal{A}_1(1^\lambda)$ where $\mathcal{A}_1$ is an computationally unbounded adversary. We parse CRS^* as $(\mathsf{CRS}_0^*, \mathsf{CRS}_1^*)$ where $\mathsf{CRS}_0^* \in \mathbb{Z}_q^{n \times m}$.
2. Compute (c, π_z) where $c = \mathsf{Com}(\mathbf{r}; R)$ for $\mathbf{r} \in \{-1, 0, 1\}^m, \mathbf{r} \neq \mathbf{0}$, and $\mathbf{Ar} \equiv \mathbf{0} \bmod q$ and for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1^*, S_{\mathsf{CRS}_0^*,x,c}, (\perp, \perp, w))$.
3. $b \leftarrow \mathcal{A}_2(\mathsf{CRS}^*, x, (c, \pi_z), \mathsf{Aux})$ where $b \in \{0, 1\}$ and $\mathcal{A}_2$ is a polynomial-sized adversary. The output of the experiment is b.

$\underline{H_{2,\mathcal{A}}(1^\lambda) = H_{\mathsf{Sim},\mathcal{A}}(1^\lambda)}$:

1. $(x, w, \mathsf{Aux}, \mathsf{CRS}^*) \leftarrow \mathcal{A}_1(1^\lambda)$ where $\mathcal{A}_1$ is an computationally unbounded adversary. We parse CRS^* as $(\mathsf{CRS}_0^*, \mathsf{CRS}_1^*)$ where $\mathsf{CRS}_0^* \in \mathbb{Z}_q^{n \times m}$.
2. Compute (c, π_z) where $c = \mathsf{Com}(\mathbf{r}; R)$ for $\mathbf{r} \in \{-1, 0, 1\}^m, \mathbf{r} \neq \mathbf{0}$, and $\mathbf{Ar} \equiv \mathbf{0} \bmod q$ and for some randomness R and $\pi_z = \mathsf{ZAP}_2(\mathsf{CRS}_1^*, S_{\mathsf{CRS}_0^*,x,c}, (\mathbf{r}, R, \perp))$.
3. $b \leftarrow \mathcal{A}_2(\mathsf{CRS}^*, x, (c, \pi_z), \mathsf{Aux})$ where $b \in \{0, 1\}$ and $\mathcal{A}_2$ is a polynomial-sized adversary. The output of the experiment is b.

Lemma 5.12. *$H_{0,\mathcal{A}}(1^\lambda) \approx_c H_{1,\mathcal{A}}(1^\lambda)$.*

Proof. The statement follows by the computational hiding property of the commitment scheme. Let $\mathcal{A}$ be a PPT adversary that distinguishes between $H_{0,\mathcal{A}}(1^\lambda)$ and $H_{1,\mathcal{A}}(1^\lambda)$. Let $x_\lambda, w_\lambda, \mathsf{CRS}_{\lambda,0}^*, \mathsf{Aux}_\lambda$ be a fixed choice of values that maximizes $\mathcal{A}$'s probability of distinguishing $H_{0,\mathcal{A}}(1^\lambda)$ and $H_{1,\mathcal{A}}(1^\lambda)$. Parse $\mathsf{CRS}_{\lambda,0}^*$ as a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Let $\mathbf{r}_\lambda \in \{-1, 0, 1\}^m$ be a non-zero vector such that $\mathbf{Ar}_\lambda \equiv \mathbf{0} \bmod q$; such an $\mathbf{r}_\lambda$ must exist by Lemma 3.12. Then we construct a non-uniform algorithm $\mathcal{B}$ that on input 1^λ has $x_\lambda, w_\lambda, \mathsf{CRS}_{\lambda,0}^*, \mathbf{r}_\lambda, \mathsf{Aux}_\lambda$ hardcoded and $\mathcal{B}$ breaks the computational hiding property of the commitment scheme.

$\mathcal{B}$ on input 1^λ chooses messages $\mathbf{m}_0 = \mathbf{0}$ and $\mathbf{m}_1 = \mathbf{r}_\lambda$. The commitment scheme challenger responds with a commitment c_b to $\mathbf{m}_b$ for some $b \in \{0, 1\}$ unknown to $\mathcal{B}$. Since $\mathbf{r}_\lambda$ is given as non-uniform advice, $\mathcal{B}$ efficiently constructs the experiment $H_b(1^\lambda)$ interacting with $\mathcal{A}$ where $\mathcal{B}$ plants c_b as the commitment c in step (3). If $\mathcal{A}$ outputs $b' \in \{0, 1\}$, then $\mathcal{B}$ outputs b'. Therefore, if $\mathcal{A}$ distinguishes between $H_0(1^\lambda, D)$ and $H_1(1^\lambda)$ with non-negligible probability, $\mathcal{B}$ will distinguish between commitments c_0 and c_1 with non-negligible probability, breaking computational hiding. □

Lemma 5.13. *$H_{1,\mathcal{A}}(1^\lambda) \approx_c H_{2,\mathcal{A}}(1^\lambda)$.*

Proof. The statement follows by the computational witness indistinguishability of the ZAP. Let $\mathcal{A}$ be an adversary that distinguishes between $H_{1,\mathcal{A}}(1^\lambda)$ and $H_{2,\mathcal{A}}(1^\lambda)$. Let $x_\lambda, w_\lambda, \mathsf{CRS}_{\lambda,0}^*, \mathsf{CRS}_{\lambda,1}^*, \mathsf{Aux}_\lambda$ be a fixed choice of values that maximizes $\mathcal{A}$'s probability of distinguishing $H_{1,\mathcal{A}}(1^\lambda)$ and $H_{2,\mathcal{A}}(1^\lambda)$. While $\mathcal{A}_1$ is computationally unbounded, in the reduction we consider a computationally bounded malicious ZAP verifier who has this choice of first round message, $\mathsf{CRS}_{\lambda,1}^*$ hardcoded to maximize the probability of winning the security game. Parse $\mathsf{CRS}_{\lambda,0}^*$ as matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Let $\mathbf{r}_\lambda \in \{-1, 0, 1\}^m$ be a non-zero vector such that $\mathbf{Ar}_\lambda \equiv \mathbf{0} \bmod q$; such an $\mathbf{r}_\lambda$ must exist by Lemma 3.12. Then we construct a non-uniform algorithm $\mathcal{B}$ that on input 1^λ has x_λ, w_λ, CRS_λ^*, $\mathbf{r}_\lambda$, Aux_λ hardcoded and $\mathcal{B}$ breaks the computational witness indistinguishability of the ZAP.

$\mathcal{B}$ on input 1^λ chooses statement $S_{\mathsf{CRS}_{\lambda,0}^*,x,c_\lambda}$ where c_λ is a commitment with randomness R to string $\mathbf{r}_\lambda$. $\mathcal{B}$ also chooses two witnesses $w_1 = (\perp, \perp, w_\lambda)$ and $w_2 = (\mathbf{r}_\lambda, R, \perp)$. $\mathcal{B}$ sends these two witnesses w_1, w_2 and statement $S_{\mathsf{CRS}_{\lambda,0}^*,x,c_\lambda}$ to the witness indistinguishability challenger. The challenger will respond with a proof π_b where π_b uses witness w_b and such that $b \in \{1, 2\}$ is unknown to $\mathcal{B}$. $\mathcal{B}$ can now efficiently construct $H_{b,\mathcal{A}}(1^\lambda)$ by interacting with $\mathcal{A}$ where in step (3), $\mathcal{B}$ chooses the challenge proof π_b as the value of π_z as defined in the experiments above. If $\mathcal{A}$ outputs $b' \in \{1,2\}$, then $\mathcal{B}$ outputs b'. If $\mathcal{A}$ distinguishes between $H_{1,\mathcal{A}}(1^\lambda)$ and $H_{2,\mathcal{A}}(1^\lambda)$ with non-negligible probability, then $\mathcal{B}$ breaks the computational witness indistinguishability of the ZAP with non-negligible probability. □

□

6 Strengthening Verifiable Witness Semantic Security

A natural strengthening of the definition of malicious CRS verifiable witness semantic security (Def. 4.9) would be to remove the verifiability requirement, which we will refer to as strong witness semantic security. We show that, perhaps counterintuitively, the notion of distributional zero-knowledge is unlikely to imply strong witness semantic security. In fact, we give an explicit counterexample in the random oracle model.

We begin by defining distributional zero-knowledge in the same manner as [CLP15]. Intuitively, this form of zero-knowledge says that for random statements a proof satisfies zero-knowledge. The simulator is allowed to depend on the distribution.

Definition 6.1 (Distributional Zero-knowledge)**.** *An interactive protocol (P, V) for language L is distributional zero-knowledge if for any probability ensemble D over a support given by statements $x \in L$, witnesses w, and auxiliary inputs* Aux *of length bounded polynomially in $|x|$, there exists a polynomial-sized algorithm* Sim *that takes as input x such that the distribution of transcripts $\langle P(x,w), V(x)\rangle$ is computationally indistinguishable from the output of* Sim *for all polynomial sized adversaries $\mathcal{A}$:*

$$\left| \Pr_{(x,w,\mathsf{Aux})\leftarrow D(\lambda)}[\mathcal{A}(x, \langle P(x,w), V(x)\rangle, \mathsf{Aux}) = 1] - \Pr_{(x,w,\mathsf{Aux})\leftarrow D(\lambda)}[\mathcal{A}(x, \mathsf{Sim}(x), \mathsf{Aux}) = 1] \right| \leq \mathsf{negl}(\lambda)$$

Definition 6.2 (Strong Witness Semantic Security)**.** *A non-interactive argument system* $(\mathsf{GenCRS}, \mathsf{Prove}, \mathsf{Verify})$ *for a language $L \in \mathsf{NP}$ is strong witness semantic security if for all polynomially-bounded efficiently samplable probability ensembles D with support over over $\{(x, w, \mathsf{Aux}, f) \mid (x,w) \in R_L, \mathsf{Aux} \in \{0,1\}^*, f \in \mathscr{F}\}$ where $\mathscr{F}$ is a set of deterministic functions, if for all polynomial-sized $\mathcal{A}$, there exists polynomial-sized $\mathcal{B}$ and a negligible function $\mu(\cdot)$ such that*

$$\Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)} \left[\begin{array}{c} \mathsf{CRS}^* \leftarrow \mathsf{GenCRS}(1^\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{CRS}^*, x, w) \\ y \leftarrow \mathcal{A}(1^\lambda, x, \pi, \mathsf{Aux}, f) : \exists w', y = f(w') \wedge (x, w') \in R_L \end{array} \right]$$
$$\leq \Pr_{(x,w,\mathsf{Aux},f)\leftarrow D(\lambda)} \left[y \leftarrow \mathcal{B}(1^\lambda, x, \mathsf{Aux}, f) : \exists w', y = f(w') \wedge (x, w') \in R_L \right] + \mu(\lambda)$$

Remark 6.3. In Definition 4.1, the distributions D considered are with respect to an adversary that is asked to output $y = f(w)$ for a specific witness w. In the above Definition 6.2, however, the adversary is allowed to output $f(w')$ for any witness w'. Note, that there are distributions D over statements, witnesses, and functions such that finding $y = f(w)$ for a specific w is hard finding $f(w')$ for any witness w' is trivially easy. For example, consider SAT instances $\varphi(x_1, \ldots, x_n)$ whose first $n/2$ variables $x_1, \ldots, x_{n/2}$ are unconstrained (do not appear in φ). It's hard to find $(x_1, \ldots, x_{n/2})$ relative to a specific witness (a specific assignment of the variables) yet trivially easy to find a value of $x_1, \ldots, x_{n/2}$ for some satisfying assignment (any string is valid).

6.1 Counterexample Construction

For intuition on why there should exist a distributional zero-knowledge protocol that is not strong witness semantic secure, consider an attempt at proving that a protocol that is distributional zero-knowledge satisfies strong witness semantic security.

In this proof attempt, we assume for sake of contradiction that $f(w')$ for some witness w' is hard to recover when the verifier interacts with a simulator Sim yet $f(w')$ is easy to recover when the verifier interacts with an honest prover. Then, we must produce an efficient algorithm, the distinguisher, that distinguishes between the real world—interacting with the honest prover—and the simulated world—interacting with the

Simulator—to contradict the zero-knowledge property. Such a contradiction implies that recovering $f(w')$ is hard in the real world, so the protocol satisfies witness semantic security. The distinguisher will produce a value y and must check if $y = f(w')$ for some valid witness w'. This verification step is efficient when we consider the notion of witness hiding. For witness hiding, the deterministic function f is the identity function so $f(w') = w'$. Then, to check if y is a valid witness, we observe that the language L belongs to the class NP so the distinguisher may run an efficient verification circuit to check if y is indeed a valid witness. If the verification circuit says y is a valid witness, the distinguisher declares that itself being in the real world, and otherwise the distinguisher declares itself in the simulated world. Crucially the distinguisher requires that verifying that y is $f(w')$ for a witness w' is efficient.

This efficient verifiablility property is certainly not true for general deterministic functions f. This issue of efficient verifiability can be circumvented when the number of witnesses is polynomially bounded in the length of the statement. In this case, the auxiliary information is allowed to contain $f(w)$ for all witnesses w . Such auxiliary information would allow the distinguisher to check if y is present in the auxiliary information, thereby verifying if $y = f(w')$ for some witness w'. Again, this hotfix does not apply generally: There exists natural distributions D for which all statements have exponentially many witnesses. One such distribution comes from observing a family of three-colorable graphs with highly rerandomizable witnesses.

Conjecture 6.4. *Let $\mathcal{G}$ be the set of 3-colorable graphs G of n^2 vertices with n separate connected components each of n vertices for $n \in \mathbb{N}$. For any graph $G \in \mathcal{G}$, let w be a witness for the three-colorability of G. Let π be a randomly chosen permutation on the coloring w. There exists a commitment scheme $\mathsf{Com}(\cdot)$ and a hash function $h(\cdot)$ such that for all witnesses w the commitment $\mathsf{Com}(w; h(w))$ is indistinguishable from $\mathsf{Com}(0; r)$ for uniform random r.*

Remark 6.5. Observe this conjecture is true if $h(\cdot)$ is a random oracle (a truly random function).

Lemma 6.6. *Assuming Conjecture 6.4, there exists a distributional zero knowledge protocol that is not strong witness semantic security.*

Proof. Consider a distribution D over tuples (G, w, f, Aux) where G is taken uniformly from the set $\mathcal{G}$ of 3-colorable graphs of n^2 vertices containing n disconnected components each of n vertices and where function f is fixed and takes as input a witness w and outputs $f(w) = \mathsf{Com}(w; h(w))$, Aux is any auxiliary information polynomially bounded in $|x|$. To construct our distributional zero-knowledge protocol, we take the standard sigma-protocol for three-colorability of a graph and construct the honest prover to also pick a random permutation π and sends $f(\pi(w)) = \mathsf{Com}(\pi(w); h(\pi(w)))$ to the verifier where $\pi(w)$ is a permuted coloring of the coloring w. Note that $\pi(w)$ is still a valid witness. Under Conjecture 6.4, a simulator is constructed by taking the original simulator for the standard sigma-protocol and additionally outputting $\mathsf{Com}(0; r)$ for some fresh randomness r. Yet, the verifier has learned $f(w')$ for some witness $w' = \pi(w)$. □

Acknowledgements

We thank the anonymous Eurocrypt reviewers for their helpful feedback regarding this work. We also thank Aayush Jain and Rex Fernando for elucidating discussions regarding ZAPs and Abhishek Jain for a helpful discussion about the two-round public-coin setting.

This research was supported in part from a Simons Investigator Award, DARPA SIEVE award, NTT Research, BSF grant 2018393, a Xerox Faculty Research Award, a Google Faculty Research Award, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through Award HR00112020024.

7 References

[AADG21] Prabhanjan Ananth, Gilad Asharov, Hila Dahari, and Vipul Goyal. Towards accountability in CRS generation. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 278–308. Springer, Heidelberg, October 2021.

[AAG$^+$24] Prabhanjan Ananth, Gilad Asharov, Vipul Goyal, Hadar Kaner, Pratik Soni, and Brent Waters. Nizks with maliciously chosen CRS: subversion advice-zk and accountable soundness. *IACR Cryptol. ePrint Arch.*, page 207, 2024.

[ABLZ17] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017.

[AEKP19] Gilad Asharov, Naomi Ephraim, Ilan Komargodski, and Rafael Pass. On perfect correctness without derandomization. Cryptology ePrint Archive, Report 2019/1025, 2019. `https://eprint.iacr.org/2019/1025`.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[Bag19] Karim Baghery. Subversion-resistant simulation (knowledge) sound nizks. In *IMA International Conference on Cryptography and Coding*, pages 42–63. Springer, 2019.

[BCG$^+$15] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 287–304. IEEE Computer Society, 2015.

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In David B. Shmoys, editor, *46th ACM STOC*, pages 505–514. ACM Press, May / June 2014.

[BFJ$^+$20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667. Springer, Heidelberg, May 2020.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

[BFS16] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016.

[BGG18] Sean Bowe, Ariel Gabizon, and Matthew D. Green. A multi-party protocol for constructing the public parameters of the pinocchio zk-snark. In Aviv Zohar, Ittay Eyal, Vanessa Teague, Jeremy Clark, Andrea Bracciali, Federico Pintore, and Massimiliano Sala, editors, *Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers*, volume 10958 of *Lecture Notes in Computer Science*, pages 64–77. Springer, 2018.

[BJMS20] Saikrishna Badrinarayanan, Aayush Jain, Nathan Manohar, and Amit Sahai. Secure MPC: Laziness leads to GOD. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 120–150. Springer, Heidelberg, December 2020.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1091–1102. ACM Press, June 2019.

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, Heidelberg, March 2015.

[CFN$^+$14] Stephen Checkoway, Matthew Fredrikson, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, and Hovav Shacham. On the practical exploitability of dual ec in tls implementations. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, page 319–335, 2014.

[CHM$^+$20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 738–768. Springer, Heidelberg, May 2020.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 66–92. Springer, Heidelberg, March 2015.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st FOCS*, pages 283–293. IEEE Computer Society Press, November 2000.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999.

[DRZ20] Vanesa Daza, Carla Ràfols, and Alexandros Zacharakis. Updateable inner product argument with logarithmic verifier and applications. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 527–557. Springer, Heidelberg, May 2020.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990.

[Fuc18] Georg Fuchsbauer. Subversion-zero-knowledge SNARKs. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 315–347. Springer, Heidelberg, March 2018.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 668–699. Springer, Heidelberg, May 2020.

[GKM$^+$18] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-SNARKs. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 698–728. Springer, Heidelberg, August 2018.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.*, 7(1):1–32, 1994.

[GO07] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 323–341. Springer, Heidelberg, August 2007.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[HJK$^+$16] Dennis Hofheinz, Tibor Jager, Dakshita Khurana, Amit Sahai, Brent Waters, and Mark Zhandry. How to generate and use universal samplers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 715–744. Springer, Heidelberg, December 2016.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 60–73, New York, NY, USA, 2021. Association for Computing Machinery.

[Khu21] Dakshita Khurana. Non-interactive distributional indistinguishability (NIDI) and non-malleable commitments. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 186–215. Springer, Heidelberg, October 2021.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Chris Umans, editor, *58th FOCS*, pages 564–575. IEEE Computer Society Press, October 2017.

[KZ20] Benjamin Kuykendall and Mark Zhandry. Towards non-interactive witness hiding. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 627–656. Springer, Heidelberg, November 2020.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. `https://eprint.iacr.org/2019/279`.

[LVW19] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. 2-message publicly verifiable WI from (subexponential) LWE. Cryptology ePrint Archive, Report 2019/808, 2019. `https://eprint.iacr.org/2019/808`.

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2111–2128. ACM Press, November 2019.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 21–39. Springer, Heidelberg, August 2013.

[Pas03] Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 316–337. Springer, Heidelberg, August 2003.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019.

A Super Dense PKE from LWE

We now give a definition for a "super" dense PKE scheme. The term "dense" is in relation to the public key space. In existing literature, dense PKE schemes have the property that with overwhelming probability a randomly chosen public key has a decryption key for which correctness holds. We define super dense PKE schemes to be schemes in which *every* possible public key has a decryption key for which correctness holds.

Definition A.1 (Super Dense Public Key Encryption Scheme)**.** *Let* $\Pi_{\mathsf{pub}} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ *be a a public-key encryption scheme. A string* $p \in \{0,1\}^{\ell_{\mathsf{pk}}(\lambda)}$*, where* $\ell_{\mathsf{pk}}(\lambda)$ *is the bit length of public keys for security parameter* λ*, is a "valid" public key if there exists a string* sk_p *such that correctness holds,*

$$\Pr\left[\mathsf{Dec}(1^\lambda, \mathsf{sk}_p, (\mathsf{Enc}(1^\lambda, p, b))) = b\right] \geq 1 - \mathsf{negl}(\lambda)$$

where the probability is over the coins of the encryption algorithm. We say that Π_{pub} *is "super dense" if for all* $\lambda \in \mathbb{N}$ *any string* $p \in \{0,1\}^{\ell_{\mathsf{pk}}(\lambda)}$ *is a valid public key.*

To build a super dense PKE scheme from LWE, we recall the dual Regev encryption scheme [GPV08].

Dual Regev Encryption Scheme [GPV08]:

Let $n = n(\lambda), m = m(\lambda), q = q(\lambda)$ be polynomials in λ where $m > 2n \log q$. Let χ denote the LWE error distribution.

- $\mathsf{Keygen}(1^\lambda)$: Sample uniform randomly $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$ and $\mathbf{s} \in \{-1,0,1\}^m$. Let $\mathsf{pk} = \mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \mathbf{s}^\top \mathbf{A} \end{bmatrix}$ and $\mathsf{sk} = (\mathbf{s}^\top, -1)$. Output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(1^\lambda, \mathsf{pk}, b \in \{0,1\})$: Parse pk as $\mathbf{A}'$. Sample a random error vector $\mathbf{e}'$ from χ^{m+1}. Sample a random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Let $\mathbf{b} \in \mathbb{Z}_q^{m+1}$ be a vector with $b \cdot \lfloor q/2 \rfloor$ in $(m+1)$th index and 0 elsewhere. Output $\mathsf{ct} = \mathbf{A}'\mathbf{u} + \mathbf{e} + \mathbf{b}$.
- $\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{ct})$: Parse sk as $\mathbf{s}'^\top = (\mathbf{s}^\top, -1) \in \mathbb{Z}_q^{m+1}$. Parse ct as $\mathbf{a}'' \in \mathbb{Z}_q^{m+1}$. Compute $r \leftarrow \mathbf{s}'^\top \cdot \mathbf{a}'' \in \mathbb{Z}_q$ and round r to closer of 0 and $\lfloor q/2 \rfloor$. If r rounds to 0, then output 0. Otherwise output 1.

Remark A.2. By observation, the Dual Regev cryptosystem's decryption algorithm is expressible as a circuit in NC^1.

Remark A.3. By a valid secret key, we mean a secret key such that correctness of decryption holds. While any secret key generated by $\mathsf{Keygen}(1^\lambda)$ will have a -1 in the last entry, this requirement is not a necessary condition for a vector $\mathbf{s}' \in \mathbb{Z}_q^{m+1}$ to be a valid secret key. To see this, observe that for any $\mathsf{pk} = \mathbf{A}'$, correctness of decryption only requires that the secret key $\mathbf{s}'$ satisfy three conditions: (1) $\mathbf{s}'^\top \cdot \mathbf{A}' = \mathbf{0} \in \mathbb{Z}_q^n$, (2) $\mathbf{s}'$ is short, (3) $\mathbf{s}'$ has a non-zero $(m+1)$th entry.

Dual Regev is Dense but not Super Dense The totality of the SIS problem, as expressed in Lemma 3.12, guarantees a non-zero short solution to for any matrix $\mathcal{A} \in \mathbb{Z}_q^{(m+1) \times n}$ where $m > 2n \log q$. However, this short solution may have a zero in its last entry. A zero in its last entry violates correctness. Specifically, using this short solution as a secret key sk in the decryption algorithm will wipeout the message bit that was placed in the $(m+1)$th index by the $\mathsf{Enc}(\cdot,\cdot,\cdot)$ algorithm. In fact, the dual Regev encryption scheme cannot be super dense. Consider the public key given by a matrix $\mathbf{A} \in \mathbb{Z}_q^{(m+1) \times n}$ whose entries are all zero except for entries of the last column which are $\lfloor q/2 \rfloor$. Then the only short solutions $\mathbf{s} \in \mathbb{Z}_q^{m+1}$ for $\mathbf{A}$ must have a zero in the last entry. Such short solutions cannot serve as valid decryption keys because they would wipe out the message bit.

A.1 Super Dense Dual Regev Encryption Scheme

We give a simple modification to the dual Regev encryption scheme to prevent this correctness violation. To encrypt bit b, our scheme give out $n+1$ dual Regev encryptions of the bit b where the ith encryption, instead of using $\mathbf{b} \in \mathbb{Z}_q^{m+1}$ as defined in the above scheme, uses a vector $\mathbf{b}^{(i)} \in \mathbb{Z}_q^{m+1}$ whose ith-entry is $b \cdot \lfloor q/2 \rfloor$ and other entries are 0. The correctness violation is addressed because any non-zero short solution to a matrix $\mathbf{A} \in \mathbb{Z}_q^{(m+1)\times n}$ has a non-zero entry, and that non-zero entry guarantees that we can recover the message bit from at least one of the $m+1$ encryptions. The security remains by the hardness of decisional LWE with polynomially many samples.

Super Dense Dual Regev Encryption Scheme:

Let $n = n(\lambda), m = m(\lambda), q = q(\lambda)$ be polynomials in λ where $m > 2n \log q$. Let χ denote the LWE error distribution.

- $\mathsf{Keygen}(1^\lambda)$: Sample uniform randomly $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}$ and $\mathbf{s} \in \{-1, 0, 1\}^m$. Let $\mathsf{pk} = \mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \mathbf{s}^\top \mathbf{A} \end{bmatrix}$ and $\mathsf{sk} = (\mathbf{s}^\top, -1)$. Output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(1^\lambda, \mathsf{pk}, b \in \{0,1\})$: Parse pk as $\mathbf{A}'$. For $i \in [n+1]$, let $\mathbf{b}^{(i)} \in \mathbb{Z}_q^{n+1}$ denote a vector with $b \cdot \lfloor q/2 \rfloor$ in $(i+1)$th index and 0 elsewhere. For each $i \in [n+1]$, sample a random error vector $\mathbf{e}'^{(i)}$ from χ^{m+1}. Sample a random vector $\mathbf{u}^{(i)} \leftarrow \mathbb{Z}_q^{n+1}$. Let $\mathsf{ct}_i = \mathbf{A}'\mathbf{u}^{(i)} + \mathbf{e}^{(i)} + \mathbf{b}^{(i)}$ Output $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_{n+1})$.
- $\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{ct})$: Parse sk as $\mathbf{s}'^\top = (\mathbf{s}^\top, -1) \in \mathbb{Z}_q^{m+1}$. Parse $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_{n+1})$ and parse ct_{n+1} as $\mathbf{a}'' \in \mathbb{Z}_q^{m+1}$. Compute $r \leftarrow \mathbf{s}'^\top \cdot \mathbf{a}'' \in \mathbb{Z}_q$ and round r to closer of 0 and $\lfloor q/2 \rfloor$. If r rounds to 0, then output 0. Otherwise output 1.

Remark A.4. By observation, the super dense dual Regev cryptosystem's decryption algorithm is also expressible as a circuit in NC^1.

Lemma A.5. *The super dense dual Regev cryptosystem is a super dense semantically-secure public key encryption scheme when $m > 2n \log q$.*

Proof. Correctness holds by the standard argument of choosing q large enough relative to χ and m.

Semantic security follows in two additional hybrids. Starting in the real world (the scheme above), we build our first hybrid by switching out $\mathbf{A}'$ for a uniform random matrix under the leftover hash lemma which states that $\mathbf{s}'^\top \mathbf{A}'$ is statistically close to random. Now observe that in this new hybrid every ciphertext is of the form $\mathsf{ct}_i = \left(\mathbf{A}'\mathbf{u}^{(i)} + \mathbf{e}^{(i)}\right) + \mathbf{b}^{(i)}$ where $\mathbf{A}'$. By the hardness of decisional-LWE, for all $i \in [m+1]$, we have that $\mathbf{A}'\mathbf{u}^{(i)} + \mathbf{e}^{(i)}$ is indistinguishable from random $\mathbf{r}^{(i)} \in \mathbb{Z}_q^{m+1}$. This usage of decisional-LWE gives our final hybrid where the ciphertexts $\mathsf{ct}_i = \mathbf{r}^{(i)} + \mathbf{b}^{(i)}$ is distributed uniform randomly.

Super denseness follows from the totality of the SIS problem namely by Lemma 3.12 and the observation that any non-zero short solution $\mathbf{s}' \in \mathbb{Z}_q^{m+1}$ to a matrix $\mathbf{A}' \in \mathbb{Z}_q^{(m+1)\times n}$ has a non-zero entry at some index i^*. Therefore, the decryption obtained by computing $\mathbf{s}'^\top \mathsf{ct}_{i^*}$ can recover message bit b with overwhelming probability (by the correctness of the dual Regev encryption scheme). □

B Adaptively Sound Computational ZAP from LWE

B.1 Ingredients

We present the following construction for adaptively sound statistical ZAPs from LWE which requires the following building blocks.

- A super dense PKE scheme, namely super dense dual Regev encryption (Appendix A). We observe that the key generation algorithm can be easily made surjective on the public key space.
- A μ_{CI}-correlation intractable hash function family $\mathcal{H}$ for circuit family $\mathcal{F}$ of circuits that correspond to the decryption circuit of the super dense dual Regev Encryption Scheme.
- Blum's Σ-protocol for HAM.

Remark B.1. We consider a λ-parallel repetition, where λ is the security parameter, of Blum's Σ-protocol for Graph Hamiltonicity, denoted Π_Σ whose commitment scheme is instantiated with the Dual Regev Cryptosystem.

Single iteration of Blum's Σ-protocol Π_Σ for a statement $G \in \mathsf{HAM}$:

1. The Prover samples a key pair $(\mathsf{pk}, \mathsf{sk})$ using the Dual Regev Cryptosystem key generation algorithm. It then picks a random permutation of vertices, π, and computes $\pi(G)$ and computes using pk the string a which contains the entry-wise commitment to the adjacency matrix of $\pi(G)$ and a commitment to π. The Prover sends (pk, a) to the Verifier.
2. The Verifier responds by choosing a uniform random challenge bit e.
3. If $e = 0$, the Prover sets z to be the decommitments to all of the adjacency matrix of $\pi(G)$ and the commitment to π. If $e = 1$, the Prover sets z to be the decommitments to only the edges corresponding to the Hamiltonian cycle w in $\pi(G)$. The Prover sends z to the Verifier.

Notation For repetition $i \in [\lambda]$, let a_i denote the commitments to a permutation and the permuted graph adjacency matrix, let e_i denote the challenge bit sent by the Verifier, and let z_i denote the response to the challenge bit (either z_i contains the openings to all of a_i or z_i contains the openings to edges in the permuted graph corresponding to edges in the original Hamiltonian cycle).

Remark B.2. Blum's Σ-protocol is witness indistinguishable and a parallel repetition of Blum's Σ-protocol remains witness indistinguishable.

B.2 Construction

We consider the language $\mathsf{HAM} = \{G : G \text{ is Hamiltonian}\}$.

Construction of ZAP
$\Pi_{\mathsf{ZAP}} = (\mathsf{ZAP}_1, \mathsf{ZAP}_2, \mathsf{ZAP}_V)$ for HAM.

- $K \leftarrow \mathsf{ZAP}_1(1^\lambda)$: Take as input a security parameter 1^λ. Sample a CI hash function family key $K \leftarrow \mathcal{H}.\mathsf{Setup}(1^\lambda, 0^\ell)$. Output K where K is sent to the Prover.
- $(G, \mathsf{pk}, a, e, z) \leftarrow \mathsf{ZAP}_2(1^\lambda, K, G, w)$: Take as input a security parameter 1^λ, a CI hash key K, a statement G, and a witness w.

  Run Π_Σ for instance-witness pair (G, w) in parallel for λ repetitions, more specifically:

  1. Sample a uniformly random public key pk. For repetition $i \in [\lambda]$, compute a_i using the Dense Encryption Scheme as the commitment scheme with key public key pk. For each i, a_i contains a commitment to a randomly chosen permutation π_i, commitments to every entry of the adjacency matrix of $\pi_i(G)$, and commitments to the indices of Hamiltonian cycle. Set $a = (a_1, \ldots, a_\lambda)$.
  2. Compute $e = \mathcal{H}(K, a)$ where $e = (e_1, \ldots, e_\lambda) \in \{0,1\}^\lambda$.
  3. Then, using the witness w_{new} that is a Hamiltonian cycle in graph G_{new}, compute $z = (z_1, \ldots, z_\lambda)$ according to Π_Σ where z_i is the response to challenge bit e_i in Π_Σ.

  Output $(G, \mathsf{pk}, a, e, z)$.
- $0/1 \leftarrow \mathsf{ZAP}_V\ (G, \mathsf{pk}, a, e, z)$: For each repetition $i \in \lambda$, run the Verifier for Π_Σ using $\mathsf{pk}, a_i, e_i, z_i$. If the Verifier accepts in all repetitions, output 1. Otherwise output 0.

Lemma B.3 (Correctness of Π_{ZAP})**.** Π_{ZAP} *satisfies correctness.*

Proof. Correctness follows immediately by the perfectly correctness of Blum's Σ-protocol for HAM. The knowledge of a Hamiltonian cycle and producing a commitment corresponding to this cycle allows the honest Prover to always give the appropriate decommitments z in response to any challenge string e. □

Lemma B.4 (Soundness of Π_{ZAP})**.** *Π_{ZAP} is adaptively computationally sound.*

Proof. We proceed via a series of hybrids. We highlight the changes using the color red below.

$\underline{H_{0,\mathcal{A}}(1^\lambda)}$:

1. The Verifier computes $(\widehat{\mathsf{pk}}, \widehat{\mathsf{sk}}) \leftarrow \mathsf{PKE.Keygen}(1^\lambda)$ and computes $(K, \tau) \leftarrow \mathsf{ZAP}_1(1^\lambda)$ and sends K to the Prover.
2. The malicious Prover $\mathcal{A}$ chooses and sends $(G^*, \mathsf{pk}^*, a, e, z)$ to the Verifier.
3. The Verifier outputs $b \leftarrow \mathsf{ZAP}_V\,(G^*, \mathsf{pk}^*, a, e, z)$. The output value of $H_0(1^\lambda)$ is b.

$\underline{H_{1,\mathcal{A}}(1^\lambda)}$:

1. The Verifier computes $(\widehat{\mathsf{pk}}, \widehat{\mathsf{sk}}) \leftarrow \mathsf{PKE.Keygen}(1^\lambda)$ and constructs a circuit for the efficient function $f_{\mathsf{bad},\widehat{\mathsf{sk}}}$ defined below.

   $f_{\mathsf{bad},\widehat{\mathsf{sk}}}(G, a)$: takes a statement G and takes commitments $a = (a_1, \ldots, a_\lambda)$ as input. For each $i \in [\lambda]$, the function sets $e_i = 0$ or $e_i = 1$ depending on which challenge bit in Π_Σ has a response that would cause the Verifier to accept. To compute this, the function decrypts/decommmits a_i using $\widehat{\mathsf{sk}}$ to recover the permutation $\pi^{(i)}$ and an adjacency matrix H_i.

   - If $H_i = \pi^{(i)}(G)$, then set $e_i = 0$ denoting a challenge in Π_Σ asking for the entire decommitment.
   - Otherwise, set $e_i = 1$ denoting a challenge in Π_Σ asking for the decommitment to edges in H_i that correspond to a Hamiltonian cycle in x.

   The function outputs $e = (e_1, \ldots, e_\lambda)$.

   The Verifier computes $K^* \leftarrow \mathcal{H}.\mathsf{Setup}(1^\lambda, f_{\mathsf{bad},\widehat{\mathsf{sk}}})$ and sends K^* to the Prover.
2. The malicious Prover $\mathcal{A}$ chooses and sends $(G^*, \mathsf{pk}^*, a, e, z)$ to the Verifier.
3. The Verifier outputs $b \leftarrow \mathsf{ZAP}_V\,(G^*, \mathsf{pk}^*, a, e, z)$. The output value of $H_1(1^\lambda)$ is b.

Consider an arbitrary PPT adversary P^*. Suppose for sake of contradiction that there exists some polynomial $q(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$ such that in $\mathcal{H}_{0,\mathcal{A}}(1^\lambda)$

$$\Pr_{\substack{K,(\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda)\\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda,K)}} [G^* \notin L \wedge V(K, G^*, \mathsf{pk}^*, a, e, z) = 1] > \frac{1}{q(\lambda)}.$$

Note by the super dense public key property, there exists a valid secret key sk^* for any pk^* chosen by P^*. In the real world, the CI hash key K is not generated using sk, so K is independent from pk, sk.

$$\Pr_{\substack{K,(\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda)\\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda,K)}} [(G^* \notin L) \wedge (V(K, G^*, \mathsf{pk}^*, a, e, z) = 1) \wedge (\mathsf{pk} = \mathsf{pk}^*)] \geq 2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)}.$$

Lemma B.5 (Soundness part 1, $\mathcal{H}_0$ to $\mathcal{H}_1$)**.** *Assuming the ϵ_{key}-statistical key indistinguishability of the CI, if*

$$\Pr_{\substack{K,(\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda)\\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda,K)}} [(G^* \notin L) \wedge (V(K, G^*, \mathsf{pk}^*, a, e, z) = 1) \wedge (\mathsf{pk} = \mathsf{pk}^*)] \geq 2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)},$$

then

$$\Pr_{\substack{K \leftarrow \mathcal{H}.\mathsf{Setup}(1^\lambda, f_{\mathsf{bad},\widehat{\mathsf{sk}}}) \\ (\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda) \\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda, K)}} [(G^* \notin L) \wedge (V(K, G^*, \mathsf{pk}^*, a, e, z) = 1) \wedge (\mathsf{pk} = \mathsf{pk}^*)] \geq 2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)} - 2 \cdot \epsilon_{key}.$$

Proof. Assume for sake of contradiction that

$$\Pr_{\substack{K \leftarrow \mathcal{H}.\mathsf{Setup}(1^\lambda, f_{\mathsf{bad},\widehat{\mathsf{sk}}}) \\ (\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda) \\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda, K)}} [(G^* \notin L) \wedge (V(K, G^*, \mathsf{pk}^*, a, e, z) = 1) \wedge (\mathsf{pk} = \mathsf{pk}^*)] < 2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)} - 2 \cdot \epsilon_{key}$$

then we could construct an unbounded adversary $\mathcal{B}$ that distinguishes between the random key K and the key K for $f_{\mathsf{bad},\widehat{\mathsf{sk}}}$. This unbounded adversary picks a random $(\mathsf{pk}, \mathsf{sk})$ pair, constructs $f_{\mathsf{bad},\mathsf{sk}}$, and obtains a key to either $f_{\mathsf{bad},\mathsf{sk}}$ or a uniformly random key. $\mathcal{B}$ now interacts with P^* to decided what type of key it took as input. If P^* outputs $(G^*, \mathsf{pk}^*, a, e, z)$ such that $G^* \notin L$, the Verifier accepts $(G^*, \mathsf{pk}^*, a, e, z)$, and $\mathsf{pk} = \mathsf{pk}^*$, then $\mathcal{B}$ guesses it has a uniform key so $\mathcal{B}$ outputs the bit 0. Otherwise, $\mathcal{B}$ guesses it has a structured key for $f_{\mathsf{bad},\widehat{\mathsf{sk}}}$ and outputs the bit 1. The guessing advantage of $\mathcal{B}$ is strictly larger than

$$\frac{1}{2}\left(2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)} + \left(1 - \frac{1}{2}\left(2^{-|\mathsf{pk}|} + 2\epsilon_{key}\right)\right)\right) - \frac{1}{2} = \epsilon_{key}$$

contradicting the ϵ_{key}-statistical key indistinguishability of the CI hash family. □

Lemma B.6 (Soundness part 2, $\mathcal{H}_1$ is hard)**.** *Consider the function* $f_{\mathsf{bad},\widehat{\mathsf{sk}}}$ *defined in Hybrid* $\mathcal{H}_{1,P^*}(1^\lambda)$ *above. Assuming the* μ_{CI}*-correlation intractability of the CI hash family, defining Hybrid* $\mathcal{H}_{1,P^*}(1^\lambda)$ *as above, probability*

$$\Pr_{\substack{K \leftarrow \mathcal{H}.\mathsf{Setup}(1^\lambda, f_{\mathsf{bad},\widehat{\mathsf{sk}}}) \\ (\mathsf{pk},\mathsf{sk}) \leftarrow V(1^\lambda) \\ G^*,\mathsf{pk}^*,a,e,z \leftarrow P^*(1^\lambda, K)}} [(G^* \notin L) \wedge (V(K, G^*, \mathsf{pk}^*, a, e, z) = 1) \wedge (\mathsf{pk} = \mathsf{pk}^*)] \leq \mu_{\mathsf{CI}}$$

Proof. Note that $f_{\mathsf{bad},\widehat{\mathsf{sk}}}(G^*, a)$ is constructed to output the only possible bad challenge bits on which an adversary could respond to when it chooses statement G^* and commitments a. Therefore, if $P^*(K)$ outputs $(G^*, \mathsf{pk}^*, a, e, z)$ such that $\mathsf{pk} = \mathsf{pk}^*$ and V accepts, it must be that $((G^*, a), \mathcal{H}_K(G^*, a)) = ((G^*, a), f_{\mathsf{bad},\widehat{\mathsf{sk}}}(G^*, a))$. □

To obtain our desired contradiction, we apply complexity leveraging. Namely, assume the subexponential hardness of LWE so that $\mu_{\mathsf{CI}} = 2^{-\lambda^\delta}$ for a small constant $\delta > 0$. Then choose $\lambda^\delta > |\mathsf{pk}|^2$ so that $|\mathsf{pk}| < \lambda^{\delta/2}$. Then, $\mu_{\mathsf{CI}} = 2^{-\lambda^\delta} = o\left(2^{-|\mathsf{pk}|}\right)$. But observe by Lemma B.5 and Lemma B.6, we now have a contradiction since $\mu_{\mathsf{CI}} < 2^{-|\mathsf{pk}|} \cdot \frac{1}{q(\lambda)} - 2 \cdot \epsilon_{key}$, where ϵ_{key} can be made exponentially small in λ by the leftover hash lemma. □

Lemma B.7 (Computational witness indistinguishability of Π_{ZAP})**.** Π_{ZAP} *satisfies computational witness indistinguishability.*

Proof. We proceed via a series of hybrids. In these hybrids, we consider $G \in \mathsf{HAM}$ be a graph with two witnesses w_0, w_1 such that $w_0 \neq w_1$. We highlight the changes using the color red below.

$\underline{H_{0,\mathsf{WI},\mathcal{A}}(1^\lambda)}$:

1. The Verifier $\mathcal{A}$ sends (G, w_0, w_1, K) to the Prover.

2. The Prover sends $(G, \mathsf{pk}, a, e, z) \leftarrow \mathsf{ZAP}_2(1^\lambda, K, G, w_0)$ to the Verifier where w_0 is some witness for G.

3. The Verifier $\mathcal{A}$ outputs a guess $b \in \{0, 1\}$. We say that $H_{0,\mathsf{WI},\mathcal{A}}(1^\lambda) = b$.

$\underline{H_{1,\mathsf{WI},\mathcal{A}}(1^\lambda, G)}$:

1. The Verifier $\mathcal{A}$ sends (G, w_0, w_1, K) to the Prover.
2. The Prover sends $(G, \mathsf{pk}, a, e, z) \leftarrow \mathsf{ZAP}_2(1^\lambda, K, G, w_1)$ to the Verifier where w_1 is some witness for G.
3. The Verifier $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. We say that $H_{1,\mathsf{WI},\mathcal{A}}(1^\lambda) = b$.

Π_Σ is computationally zero knowledge so it is also computationally witness indistinguishable. Suppose $\mathcal{A}$ is an adversary, namely the malicious Verifier in the two hybrids, that distinguishes between the two hybrids with $1/p(\lambda)$ probability for some polynomial p. Then we construct an adversary $\mathcal{B}$ who breaks the witness indistinguishability of Π_Σ. $\mathcal{B}$ therefore acts as the challenger in the Hybrids and as the adversary for the computational witness indistinguishability game for Π_Σ.

In the hybrid indistinguishability game, $\mathcal{A}$ first chooses a statement $G \in \mathsf{HAM}$ with two witnesses w_0, w_1. $\mathcal{A}$ sends the tuple (G, w_0, w_1, K) that contains the challenge statement, the challenge witnesses, and a first round message K to $\mathcal{B}$. $\mathcal{B}$ forwards (G, w_0, w_1) to the challenger for the Π_Σ's computational witness indistinguishability game. The challenger will reply with a first round message (pk, a). Then $\mathcal{B}$ computes $e \leftarrow \mathcal{H}.\mathsf{Eval}(K, (G, a))$ and sends e back to the challenger. The challenger then responds to $\mathcal{B}$ with z. Finally, $\mathcal{B}$ sends $(G, \mathsf{pk}, a, e, z)$ to $\mathcal{A}$ who will respond with a single bit b which is forwarded to the challenger. If the challenger uses w_0 to produce its messages, then the messages that $\mathcal{B}$ sends to $\mathcal{A}$ exactly correspond to the transcript in $H_{0,\mathsf{WI},\mathcal{A}}(1^\lambda)$, and if the challenger uses w_1 to produce its messages then those messages correspond to the transcript in $H_{1,\mathsf{WI},\mathcal{A}}(1^\lambda)$. Therefore, if $\mathcal{A}$ succeeds in distinguishing between $H_{0,\mathsf{WI},\mathcal{A}}(1^\lambda)$ and $H_{1,\mathsf{WI},\mathcal{A}}(1^\lambda)$ with $\frac{1}{p(\lambda)}$ probability, then $\mathcal{B}$ will succeed in breaking the witness indistinguishability of Π_Σ with probability $\frac{1}{p(\lambda)}$. By the computational witness indistinguishability of Π_Σ, it must be the case that no such adversary $\mathcal{A}$ exists and the two hybrids are indistinguishable. □

C Diffie-Hellman Knowledge-of-Exponent Assumption

Bellare et al. [BFS16] introduce the following knowledge-of-exponent assumption (KEA) for which we import their notation. A PPT algorithm called the bilinear-group generator $\mathsf{GG}(1^\lambda)$ outputs a description of a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ where p is a prime of length λ, $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order p, g generates $\mathbb{G}$, and $\mathbf{e} : \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$ is a billinear map that is non-degenerate ($\langle \mathbf{e}(g,g) \rangle = \mathbb{G}_T$). In the following security game, both algorithms M and E are crucially uniform algorithms as Bellare et al. themselves note that the assumption is false if they are allowed to be non-uniform.

```
Game $\mathsf{KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$

    $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{GG}(1^\lambda)$; $h_0, h_1 \leftarrow \mathbb{G}$; $r \leftarrow \{0,1\}^{\mathsf{M.randLen}(\lambda)}$
    $(S_0, S_1, S_2) \leftarrow \mathsf{M}(1^\lambda, h_0, h_1; r)$; $s \leftarrow \mathsf{E}(1^\lambda, h_0, h_1, r)$
    Return $(\mathbf{e}(S_0, S_1) = \mathbf{e}(g, S_2)$ and $g^s \neq S_0$ and $g^s \neq S_1)$.
```

As a summary, the experiment aboveoutcome is 1 if E fails to extract and 0 if E extracts one of the exponents (or if M fails to generate proper triples).

Definition C.1. *Let* $\mathbf{Adv}^{\mathrm{ke}}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = \Pr[\mathsf{KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1]$. *The Diffie-Hellman knowledge-of-exponent assumption (DH-KEA) holds if for every* PPT *algorithm* M, *there exists a* PPT *algorithm* E, *called the extractor, such that* $\mathbf{Adv}^{\mathrm{ke}}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = \mathsf{negl}(\lambda)$.

While the assumption as stated verbatim by Bellare et. al. does not yet have an explicit attack, we show that this assumption becomes invalid in a natural setting in which *universal samplers* exist. The notion of universal samplers was introduced by [HJK+16] and their existence can be proved from the existence of indistinguishability obfuscation and one way functions. When we consider *uniform* algorithms M and E that receive an honestly generated universal sampler as auxiliary input, the analogous knowledge-of-exponent assumption becomes provably false. This observation points towards the instability of such knowledge assumptions, raising legitimate concerns about the security of schemes that rely on such assumptions.

C.1 Selectively Secure One-Time Universal Samplers

Let $\ell(\lambda), m(\lambda), k(\lambda)$ be efficiently computable polynomials. A pair of efficient algorithms $(\mathsf{Setup}, \mathsf{Sample})$, where $\mathsf{Setup}(1^\lambda) \to U, \mathsf{Sample}(U, d) \to p_d$, is a selectively-secure one-time universal sampler scheme if there exists an efficient algorithm $\mathsf{SimUGen}$ such that:

- There exists a negligible function $\mathsf{negl}(\cdot)$ such that for all circuits d of length ℓ, taking m bits of input, and outputting k bits, and for all strings $p_d \in \{0,1\}^k$, we have that:

$$\Pr[\mathsf{Sample}\left(\mathsf{SimUGen}(1^\lambda, d, p_d), d\right) = p_d] = 1 - \mathsf{negl}(\lambda)$$

- For every efficient adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_2$ outputs one bit, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Bigg| \Pr[\mathcal{A}_2(\mathsf{Setup}(1^\lambda), \sigma) = 1 : (d^*, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)] -$$

$$\Pr[\mathcal{A}_2(\mathsf{SimUGen}(1^\lambda, d^*, p_d), \sigma) = 1 : p_d \leftarrow d^*(r), r \leftarrow \{0,1\}^m, (d^*, \sigma) \leftarrow \mathcal{A}_1(1^\lambda)] \Bigg| = \mathsf{negl}(\lambda)$$

where σ denotes auxiliary information.

Theorem C.2 ([HJK$^+$16]). *If indistinguishability obfuscation and one-way functions exist, then there exists a selectively secure one-time universal sampler scheme.*

C.2 A Variant of Discrete Log

We introduce an assumption we state as the $\mathsf{a\text{-}OR\text{-}b}$ search problem and prove that this is at least as hard as the Discrete Logarithm Assumption. Let $\mathsf{dGG}(1^\lambda)$ be a PPT algorithm that outputs a group generator and prime of length λ. Formally, the $\mathsf{a\text{-}OR\text{-}b}$ search problem is hard if for every PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that,

$$\Pr[x \in \{a, b\} : x \leftarrow \mathcal{A}(g^a, g^b, g^{ab}), (a, b) \leftarrow \mathbb{Z}_p, (p, g) \leftarrow \mathsf{dGG}(1^\lambda)] \leq \mathsf{negl}(\lambda).$$

Claim C.3. *Assuming the hardness of the Discrete Logarithm problem, the* $\mathsf{a\text{-}OR\text{-}b}$ *search problem is hard.*

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ which breaks the $\mathsf{a\text{-}OR\text{-}b}$ search problem, then there exists a PPT adversary $\mathcal{B}$ which has a on negligible advantage of breaking the discrete log problem.

We construct $\mathcal{B}$ as follows. On input (g, g^a), it samples $b \in \mathbb{Z}_p$ at random, and computes g^b, g^{ab}. $\mathcal{B}$ runs, with probability 1/2, $\mathcal{A}$ on input (g^a, g^b, g^{ab}), and otherwise runs $\mathcal{A}$ on input (g^b, g^a, g^{ab}). This addresses the case that the $\mathsf{a\text{-}OR\text{-}b}$ breaker $\mathcal{A}$ is biased towards recovering one of a or b. $\mathcal{B}$ returns the output of $\mathcal{A}$.

Without loss of generality, the algorithm $\mathcal{A}$ recovers a with some non-negligible probability ε, so our algorithm $\mathcal{B}$ recovers a with non-negligible probability $\varepsilon/2$. The existence of $\mathcal{A}$ therefore leads to a contradiction on the hardness of the discrete log problem. □

C.3 Diffie Hellman-Knowledge of Exponent Assumption with Universal Samplers

A reasonable expectation for any hardness assumption involving uniform algorithms is that the hardness assumption continues to hold when all uniform algorithms are given some public auxiliary information. In particular, we consider a variant of the security game for the DH-KEA assumption in which a trusted universal sampler is accessible to all algorithms. We formally specify the assumption below.

```
GAME $\mathsf{UniSamp\text{-}KE}_{\mathsf{GG,M,E}}(\lambda)$

    $(p, \mathbb{G}, \mathbb{G}_T, \mathsf{e}, g) \leftarrow \mathsf{GG}(1^\lambda)$; $U \leftarrow \mathsf{UniversalSampler.Setup}(1^\lambda)$;
    $h_0, h_1 \leftarrow \mathbb{G}$; $r \leftarrow \{0,1\}^{\mathsf{M.randLen}(\lambda)}$
    $(S_0, S_1, S_2) \leftarrow \mathsf{M}(1^\lambda, h_0, h_1, U; r)$; $s \leftarrow \mathsf{E}(1^\lambda, h_0, h_1, U, r)$
    Return $(\mathbf{e}(S_0, S_1) = \mathbf{e}(g, S_2)$ and $g^s \neq S_0$ and $g^s \neq S_1)$.
```

The interpretation of this experiment is identical to the above experiment except for the fact that both of the uniform PPT algorithms M and E additionally take as input the universal sampler parameters U.

Analogously, we have the following hardness assumption that we hope to be true if the DH-KEA assumption is true.

Definition C.4. *Let* $\mathbf{Adv}^{\mathrm{ke}}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = \Pr[\mathsf{UniSamp\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1]$*. The Diffie-Hellman Knowledge-of-exponent Assumption with Universal Samplers (US-DH-KEA) holds if for every* PPT *algorithm* M*, there exists a* PPT *algorithm* E*, called the extractor, such that* $\mathbf{Adv}^{\mathrm{ke}}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = \mathsf{negl}(\lambda)$*.*

C.3.1 US-DH-KEA is False

We show that the US-DH-KEA assumption is false by constructing a uniform machine M for which there is no extractor E. In particular, we show that for this specific machine M, if there exists an extractor E that satisfies the game definition above, then this extractor will break the selective one-time security of the universal sampler.

We begin by defining a simple circuit C. Let $C(1^\lambda, x)$ be a circuit which interprets x as $(a, b) \in \mathbb{Z}_p^2$ and outputs (g^a, g^b, g^{ab}). We then define the algorithm M as follows

```
$\mathsf{M}(1^\lambda, h_0, h_1, U; r)$
    1. Output $\mathsf{UniversalSampler.Sample}(U, C)$.
```

Intuitively, the universal sampler samples the output of C by running it on a (pseudo)random input. Therefore, the universal sampler hides the point (a, b) at which the circuit C is being evaluated at. This is despite the fact that C is a public, extremely simple circuit. If an extractor existed, then it effectively recovers this hidden point out of the universal sampler parameters U, contradicting the selective one-time security of the universal sampler. We now formally explain.

Theorem C.5. *Let PPT* M *be defined as above. If there exists a PPT extractor* E *such that* $\Pr[\mathsf{UniSamp\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$*, then there is PPT algorithm* $\mathcal{B}$ *that breaks the hardness of the* $\mathsf{a\text{-}OR\text{-}b}$ *search problem.*

Proof. We show that if there exists PPT extractor E such that succeeds in the game $\mathsf{UniSamp\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$ with all but negligible probability, then the same extractor succeeds in the following game, $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$, with all but negligible probability. And if the extractor succeeds in the game $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$, then it can be used to build an efficient algorithm that solves the $\mathsf{a\text{-}OR\text{-}b}$ search problem.

First, we define the intermediate game, $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$, as follows:

```
GAME $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$
    $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{GG}(1^\lambda)$; $U^* \leftarrow \mathsf{SimUGen}((1^\lambda, C, (g^a, g^b, g^{ab})), C)$;
    $h_0, h_1 \leftarrow \mathbb{G}$; $r \leftarrow \{0,1\}^{\mathsf{M.randLen}(\lambda)}$
    $(S_0, S_1, S_2) \leftarrow \mathsf{M}(1^\lambda, h_0, h_1, U^*; r)$; $s \leftarrow \mathsf{E}(1^\lambda, h_0, h_1, U^*, r)$
    Return $(\mathbf{e}(S_0, S_1) = \mathbf{e}(g, S_2)$ and $g^s \neq S_0$ and $g^s \neq S_1)$.
```

Lemma C.6. *If there exists a PPT extractor* E *such that* $\Pr[\mathsf{UniSamp\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$*, then* $\Pr[\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$*.*

Proof. Suppose that $\Pr[\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1]$ is non-negligible. In other words, E fails to recover a or b with non-negligible probability in $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda)$. If $\Pr[\mathsf{UniSamp\text{-}KE}_{\mathsf{GG},\mathsf{M},\mathsf{E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$, then we can use E to break the selective one-time security of universal sampler as follows. We will construct the pair of PPT algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ to break the universal sampler's selective one-time security. $\mathcal{A}_1$ sends the circuit C, as defined previously, to the challenger and gives some state τ to $\mathcal{A}_2$. The challenger responds with $Z \in \{U, U^*\}$. $\mathcal{A}_2$ samples uniformly random h_0, h_1, randomness r and runs the extractor $\mathsf{E}(1^\lambda, h_0, h_1, Z, r)$ to obtain some value e. If $g^e \in \{g^a, g^b\}$, then $\mathcal{A}_2$ guesses that $Z = U$ (real world), and otherwise it guesses that $Z = U'$ (ideal world). The advantage $\mathcal{A}_2$ has in distinguishing between the two worlds is non-negligible. Therefore, it must be that $\Pr[\mathsf{UniSamp\text{-}KE}_{\mathsf{GG,M,E}}(\lambda) = 1]$ is also non-negligible, or else the selective one-time security of the universal sampler is false. This proves the contrapositive of the desired statement. □

Lemma C.7. *If there exists a PPT* E *such that* $\Pr[\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG,M,E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$*, then there is a PPT adversary* $\mathcal{B}$ *which solves the* a-OR-b *search problem.*

Proof. Construct the PPT algorithm $\mathcal{B}$ as follows:

```
$\mathcal{B}(g^a, g^b, g^{ab})$
    1. $U^* \leftarrow \mathsf{SimUGen}((1^\lambda, C, (g^a, g^b, g^{ab})), C)$
    2. $r_0, r_1 \leftarrow \mathbb{Z}_p$; $h_0 \leftarrow g^{r_0}$; $h_1 \leftarrow g^{r_1}$; $r \leftarrow \{0,1\}^m$.
    3. Output $\mathsf{E}(1^\lambda, h_0, h_1, U^*, r)$.
```

The input (g^a, g^b, g^{ab}) was chosen according to a distribution where we sample uniform random a and b. An equivalent input distribution, by choice of our C, is to run C on uniform random (a, b). So the input to E is sampled exactly according to the game $\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG,M,E}}(\lambda)$. Therefore $\mathcal{B}$ fails to recover a or b with only negligible probability.

Since C always outputs the correct triple, $\Pr[\mathsf{UniSamp^*\text{-}KE}_{\mathsf{GG,M,E}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$ implies that E extracts a or b with non-negligible probability. Thus, $\mathcal{B}$ recovers either a or b with non-negligible probability. □

Combining the two lemmas gives us our desired theorem statement. □