New Techniques for Zero-Knowledge: Leveraging Inefficient Provers to Reduce Assumptions and Interaction

Marshall Ball[1] *, Dana Dachman-Soled[2] *, **, and Mukul Kulkarni[3] *,** †

[1] Columbia University
marshall@cs.columbia.edu
[2] University of Maryland
danadach@umd.edu
[3] UMass Amherst
mukul.umass@gmail.com

Abstract. We investigate the *minimal assumptions necessary* for *minimal interaction* zero-knowledge type primitives—ZAPs (two-round, public coin, witness indistinguishable proofs), NIWI (non-interactive witness indistinguishable proofs) and NIZK (non-interactive zero-knowledge proofs)—in the *standard* (no trusted setup) model. Since our goal is to obtain constructions from *Minicrypt* and/or *worst-case* assumptions only, we consider the setting where the prover is computationally *more powerful* than the simulator/zero-knowledge distinguisher. This covers both the traditional setting of computationally unbounded provers, as well as a new "fine-grained" setting that we introduce, where the prover is polynomial time and the verifier/simulator/zero-knowledge adversary are in a lower complexity class, such as NC^1.

We present constructions of ZAPs and NIWI for AM from Minicrypt and worst-case assumptions. We also present (a form of) NIZK with uniform soundness for NP, from Minicrypt and worst-case assumptions. We present analogous "fine-grained" constructions of all of the above, where the zero-knowledge adversary is limited to NC^1. Specifically, we achieve "fine-grained" ZAPs and NIWI for NP from worst-case assumptions only and achieve a form of "fine-grained" NIZK with uniform soundness for NP from worst-case and Minicrypt assumptions.

1 Introduction

A long and important line of research has been dedicated to understanding the necessary and sufficient assumptions for the existence of zero knowledge proofs (with potentially unbounded provers) for a language $\mathcal{L}$ [BMO90, OW93, IOS94]. This line of research culminated with the work of Ong and Vadhan [OV08] which fully resolved the question by proving that a language in NP has a zero knowledge protocol if and only if the language has an "instance-dependent" commitment scheme. The minimal assumptions required in the *non-interactive* zero knowledge (NIZK) setting—assuming unbounded provers and a common *reference* string (CRS) [4] (also called the "public parameters" setting)—are also well-understood. The work of Pass

* Supported by an IBM Research PhD Fellowship. This work is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via Contract No. 2019-1902070006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

** Work supported in part by NSF grants #CNS-1933033, #CNS-1840893, #CNS-1453045 (CAREER), by a research partnership award from Cisco and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology.

† Part of this work was done while the author was a student at the University of Maryland.

[4] Throughout this work we make a distinction between common *reference* string denoted as CRS and uniform random string denoted as URS. URS is sometimes referred to common *random* string in literature. We write URS to avoid the confusion and overloading.

and Shelat [Ps05], showed that (non-uniform) one-way functions are sufficient for NIZK with unbounded provers in the CRS model for all of AM, whereas NIZK with unbounded provers in the CRS model for a hard-on-average language implies the existence of (non-uniform) one-way functions.

In this work, we extend the above fundamental research directions to other types of "zero knowledge" primitives. Specifically, we investigate the *minimal assumptions necessary* for *minimal interaction* zero knowledge type primitives in the *standard* (no trusted setup) model, such as ZAPs, NIWI, and NIZK with soundness against uniform adversaries. Our goal is to obtain constructions from *Minicrypt* [Imp95] and/or *worst-case* assumptions only[5]. Such minimal assumptions are hard to achieve when the prover is efficient. In this case even constructions of NIZK in the CRS model (let alone the standard model) seem to inherently require Cryptomania [Imp95] assumptions such as enhanced trapdoor permutations. (See Section 1.2 for details.)

Therefore, in this work, we consider the setting where the prover is computationally *more powerful* than the simulator/zero knowledge distinguisher. We refer to this setting as the inefficient prover setting. This covers both the traditional setting of computationally unbounded provers, as well as a new fine-grained setting that we consider for the first time (to the best of our knowledge), where the prover is polynomial time and the verifier/simulator/zero knowledge distinguisher are in a lower complexity class, such as NC^1 (logarithmic depth, polynomial-size circuits with constant fan-in). By imposing a restriction on adversarial (and honest) verification complexity, we will see that meaningful notions of zero knowledge are achievable for any language in NP, from worst-case assumptions alone.

In the inefficient prover setting, we present several new results on the *minimal assumptions* necessary for achieving various forms of zero knowledge with *minimal interaction* and *inefficient provers*. Specifically, we present constructions of ZAPs (publicly verifiable, two-message witness indistinguishable protocols), NIWI (non-interactive witness indistinguishable protocols), and (a form of) NIZK (non-interactive zero knowledge protocol) with uniform soundness in the standard model, with inefficient provers, from Minicrypt and worst-case assumptions only. We investigate "fine-grained" versions of all of the above, and present constructions of the analogous "fine-grained" zero knowledge primitives for NC^1 adversaries, from worst-case assumptions (for the case of ZAPs and NIWI) and worst-case and Minicrypt assumptions (for the case of NIZK with uniform soundness in the no-CRS model).

Technical hurdles introduced by inefficient provers. We begin our work with the following surprising observation (discussed in more detail in Section 1.1): While it is known that ZAPs and NIZKs in the uniform random string (URS) model are equivalent for the case of *efficient provers* [DN07], *both* the transformation from NIZK in the URS model to ZAPs and the transformation of ZAPs to NIZK in the URS model (which uses the well-known FLS paradigm [FLS99]) fail when the NIZK prover or ZAP prover are inefficient. (i.e. whenever the honest prover requires more computational power than the simulator and/or distinguisher). Briefly, this is because the *simulator* (in the case of constructing NIZK from ZAPs) and the *reduction* to the zero knowledge of the underlying NIZK (in the case of constructing ZAPs from NIZKs) does not have the computational power to run the *honest* prover's algorithm. In both cases, we use non-uniformity combined with novel techniques to overcome these obstacles, by "hardcoding" certain parts of the computation of the inefficient prover into the efficient simulator or reduction. Our reduction from the security of the ZAPs to the security of the underlying NIZK is therefore non-uniform. In addition, the form of zero knowledge that we obtain when we construct NIZK from ZAPs has a non-uniform flavor, which we discuss next.

Our notions of Zero Knowledge: "offline simulation." The notion of zero knowledge for which we prove equivalence with ZAPs and which our constructions of NIZK with uniform soundness in the no-CRS model enjoy, is weaker than the standard notion of zero knowledge. Specifically, the standard notion of zero knowledge in the non-interactive setting requires the existence of a single simulator Sim, that for any statement $x \in \mathcal{L}$ produces a distribution over CRS's and proofs (URS', π') that is computationally

[5] We understand Minicrypt to be chiefly characterized by the lack of key agreement (KA), and note that one-way permutations (OWP) are separated from KA via the original Impagliazzo and Rudich separation [IR89]. For the same reason, we consider Collision-Resistant Hashing to be in Minicrypt.

indistinguishable from honest CRS's and proofs (URS, π). In contrast, our definition requires existence of a distribution $\mathcal{D}_{\mathsf{Sim}}$ over small circuit simulators Sim, such that for any statement $x \in \mathcal{L}$, the distribution over (URS', π') obtained by drawing Sim from $\mathcal{D}_{\mathsf{Sim}}$ and outputting $(\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x)$ is computationally indistinguishable from honest CRS's and proofs (URS, π). The standard model setting with uniform soundness is nearly identical, except that there is no longer a CRS. Note that our definition does not trivialize the zero knowledge property, since the draw of Sim from $\mathcal{D}_{\mathsf{Sim}}$ is independent of the statement x. Indeed, we prove that our definition implies the standard notion of *witness hiding*. We additionally show it implies *weak zero knowledge*, with inverse-polynomial bounds on distinguishing advantage. Another way to view our definition, is that we allow our simulator Sim to perform an expensive "offline" pre-processing step that is independent of the statement x (corresponding to the draw of the circuit from the potentially inefficiently samplable distribution $\mathcal{D}_{\mathsf{Sim}}$)[6]. Once the statement x is received, the simulator is required to be efficient. We therefore refer to our notion as *NIZK with offline simulation* (abbreviated oNIZK).

Our notions of Zero Knowledge: the "fine-grained" setting. We additionally introduce fine-grained analogs of zero knowledge and witness indistinguishability that are a radical departure from the standard setting. In fine-grained zero knowledge, we are concerned with (very) low complexity verifiers. We wish the honest verifier to have low complexity (we will use NC^1 as a running example), but we also want to scale down the claim "no additional knowledge" leaked (beyond validity of the statement) what can be computed in this low complexity class (NC^1). The standard definition of zero knowledge simply requires that real transcripts can be simulated in probabilistic polynomial time. But if the verifier is in NC^1 the simulation complexity could in fact be substantially larger than that of the verifier, which does not capture the idea that "no additional knowledge" was leaked. While this simulation notion is stronger, we only require interactions with malicious verifiers in NC^1 to be simulatable. Moreover, simulation is only required to be indistinguishable from real to NC^1 distinguishers. In this sense, our notion of fine-grained zero knowledge is orthogonal to the standard, poly-time zero knowledge.[7]

We consider both a *standard* flavor fine-grained zero knowledge with a fixed low complexity simulator, as well as an offline-simulation notion fine-grained zero knowledge, where the simulator is (possibly inefficiently) sampled from a distribution over very low complexity circuits that take the instance as input. We also define a notion of fine-grained witness indistinguishability where indistinguishability of interactions is only required to hold for low complexity distinguishers/verifiers.

We note that interactive fine-grained zero knowledge is straightforward to achieve using fine-grained commitments (which follow from the work of [DVV16]) and a commitment-based ZK protocol (e.g. Blum-Hamiltonicity). We therefore focus exclusively on fine-grained ZAPs and NIZK.

Equivalence of ZAPs and oNIZK for unbounded provers. Our first contribution in this work is to prove that ZAPs and oNIZKs in the uniform *random* string (URS) model are equivalent even in the case of *inefficient provers*. Specifically, we show the following:

Theorem 1 (Informal). *Assuming the existence of non-adaptive oNIZK proof systems for NP with inefficient prover in the URS model, there exist ZAPs for NP with inefficient provers.*

Theorem 2 (Informal). *Assuming the existence of one-way functions (OWF) and ZAPs for NP with inefficient provers, there exist non-adaptive oNIZK proof systems for NP with inefficient prover in the URS model.*

[6] Our notion of offline simulation is comparable in some sense to non-uniform "constructions" of cryptographic primitives, such as the notion of pseudo-random functions considered in [OS17].

[7] Note that this is very different from other fine-grained flavors of zero knowledge such as "knowledge tightness" or "precise zero knowledge" [GMW91, Gol01, MP06, DG11, DG12] which look for a simulation complexity that is tight to *each* simulator. Under these notions, if a malicious verifier, V, runs for n^{c_V} steps, then the interaction with the prover should simulatable with order $O(n^{c_V})$ steps. These verifier-by-verifier notions, in some sense, recover fine-grained zero knowledge with respect to $\mathsf{TIME}(n^c)$ for all c simultaneously. In this work, we aren't concerned with such verifier-by-verifier simulation of malicious poly-time verifiers, but instead what can be achieved if one is *only* concerned with (very) simple malicious verifiers (in order to minimize assumptions).

In order to prove the first result, we surprisingly leverage a type of *design*—a combinatorial object that was used in the derandomization of BPP by Nisan and Wigderson [NW88]. To the best of our knowledge, this is a novel application of designs to the cryptographic setting.

Since even standard NIZK with inefficient provers in the URS model can be constructed from one-way permutations (OWP), our result immediately yields ZAPs with inefficient provers from the minicrypt assumption of OWP.

Theorem 3 (Informal). *Assuming the existence of one-way permutations, there exist ZAPs for NP with inefficient provers.*

Extending to the fine-grained setting. Next, we observe that our same proof technique to transform oNIZK in the URS model with inefficient provers to ZAPs also applies to the *fine-grained setting*. Here, we assume that the prover is polynomial-time, but that the verifier and distinguisher are in a lower complexity class, $\mathcal{F}$. We then require that zero knowledge/witness indistinguishability hold against distinguishers from complexity class $\mathcal{F}$. Moreover, in the case of zero knowledge, we require a distribution over simulators in $\mathcal{F}$.

For the proof technique from above to work, we require the class $\mathcal{F}$ to satisfy some mild compositional requirements, which are, in particular, satisfied by the class NC^1. Thus, our above proof strategy shows that to construct NC^1-fine-grained ZAPs, it is necessary and sufficient to construct NC^1-fine-grained oNIZKs in the URS model. Specifically, we show:

Theorem 4 (Informal). *Assuming the existence of non-adaptive NC^1-fine-grained oNIZK proof systems for NP in the URS model, there exist NC^1-fine-grained ZAPs for NP.*

Theorem 5 (Informal). *Assuming that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ and the existence of NC^1-fine-grained ZAPs for NP then there exist non-adaptive NC^1-fine-grained oNIZK proof systems for NP in the URS model.*

We next show how to construct NC^1-fine-grained *standard* NIZK in the URS model for all of NP, assuming the *worst-case assumption* that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$. Recall that in contrast to an NC^1-fine-grained oNIZK, a NC^1-fine-grained NIZK has a fixed (simulator) in (randomized) NC^1. Our result begins by converting the NIZK construction of [AR16] that works for languages $\mathcal{L}$ with randomized encodings from the CRS model to the URS model. Since randomized encodings are known for the class $\oplus L/\operatorname{poly}$, this yields an NIZK proof system in the URS model (which actually achieves *statistical zero knowledge*). We then introduce a new primitive, which we call a *$\mathcal{G}$-extractable, $\mathcal{F}$-Fine-Grained Commitment*. This is a commitment that is perfectly binding, hiding against $\mathcal{F}$, but extractable by $\mathcal{G}$. We show how to construct $\oplus L/\operatorname{poly}$-extractable, NC^1-Fine-Grained Commitment under the *worst-case* assumption that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ using techniques of [DVV16]. Then, using $\oplus L/\operatorname{poly}$-extractable, NC^1-Fine-Grained Commitment we show how to bootstrap the NIZK proof system in the URS model for the class $\oplus L/\operatorname{poly}$ to an $\mathcal{F}$-fine-grained NIZK proof system for NP in the URS model. We obtain the following theorem:

Theorem 6 (Informal). *Assuming that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$, there exist non-adaptive NC^1-fine-grained NIZK proof systems for NP in the URS model.*

Beyond ZAPs. One reason that ZAPs are a crucial tool in cryptography, is that they can be used as a building block to construct NIWI in the standard (no trusted setup) model under certain types of assumptions that are common in the derandomization literature. Indeed, the seminal work of Barak et al. [BOV07] was the first to establish this connection between derandomization assumptions and NIWI. Furthermore, NIWI in the standard model can be used to construct NIZK with soundness against uniform adversaries in the standard model. In this work we consider both constructions of NIWI and oNIZK with soundness against uniform adversaries in the standard model. The construction of NIWI is straightforward for the case of ZAPs with unbounded provers (it proceeds exactly as in the case of efficient provers), and we obtain the following:

Theorem 7 (Informal). *Assuming the existence of one-way permutations (OWP) and efficient 1/2-hitting set generator (HSG) against co-nondeterministic uniform algorithms, there exist NIWI for NP with inefficient provers.*

The assumption of the existence of a 1/2-HSG against co-nondeterministic uniform algorithms is an assumption in the derandomization literature. It is implied by worst-case assumptions about the complexity class E—consisting of languages decided by Turing machines running in $\mathsf{DTIME}(2^{O(n)})$. Specifically, it is known how to construct such a generator if E has a function of nondeterministic circuit complexity $2^{\Omega(n)}$. We therefore consider the above to be a construction from Minicrypt assumptions (the existence of OWP) and worst-case assumptions (the existence of a 1/2-HSG against co-nondeterministic uniform algorithms).

Some technicalities arise when applying the construction transforming $\mathcal{F}$-fine-grained ZAPs to $\mathcal{F}$-fine-grained NIWI. However, for the case of NC^1-fine-grained ZAPs, the construction does go through since any computation that the verifier must perform in the transformed protocol that is not in NC^1, can simply by performed by the prover and its correctness verified by the verifier in NC^1. We thus obtain the following theorem:

Theorem 8 (Informal). *Assuming that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ and the existence of efficient 1/2-HSG against co-nondeterministic uniform algorithms, there exist NC^1-fine-grained NIWI for NP.*

When going from NIWI to oNIZK in the standard model, it no longer makes sense to assume a completely computationally unbounded prover. This is because a computationally unbounded prover can trivially break soundness, which now only holds against uniform adversaries running in some time T. Instead, we show how to obtain oNIZK in the standard model (from sub-exponential OWP, subexponential uniform collision-resistant hash functions, and derandomizaton assumptions) with the following properties: It retains soundness against adversaries running in time 2^{n^c}, for constant $c \geq 1$, while the honest prover runs in time 2^{n^ϵ}, where constant $\epsilon < 1$. Moreover, it provides zero knowledge against polynomial time adversaries. This works because the non-uniform simulator can receive all the "hidden bits" as non-uniform advice, and then execute the underlying honest "hidden bits" efficient prover, given the secondary witness.

Theorem 9 (Informal). *Assuming the existence of one-way permutations, efficient 1/2-HSG against co-nondeterministic uniform algorithms, and sub-exponentially-hard uniform collision resistant hash functions, then for any constant $0 < \epsilon < 1$ and constant $c \geq 1$, there exist oNIZK in the standard model for NP with honest provers running in uniform time 2^{n^ϵ} and soundness against uniform adversaries running in time 2^{n^c}, where n is security parameter.*

For the fine-grained case, we encounter a difficulty since our underlying proof does not have the property that the honest prover is efficient in the "hidden bits" model. We therefore go through a more complicated process where different components of the proof are pre-processed and then selected appropriately once the input statement is received by the simulator. Our final result yields NC^1-fine-grained oNIZK in the standard model with soundness against uniform PPT adversaries, assuming $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$, uniform collision resistant hash functions and derandomization assumptions. Our final theorem follows:

Theorem 10 (Informal). *Assuming that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$, the existence of efficient 1/2-HSG against co-nondeterministic uniform algorithms, and the existence of uniform collision resistant hash functions, there exist NC^1-oNIZK in the standard model for NP.*

1.1 Technical Overview

ZAPs from NIZK with inefficient provers. Let us begin by recapping the construction of ZAPs from a non-adaptive NIZK proof system with an efficient prover in the URS model.

The public coin verifier sends a random string r, which is partitioned into n' sections $r_1||\cdots||r_{n'}$. Each r_i is a bitstring of length n, where n is also the bit length of the URS for the underlying NIZK proof system. Upon receiving $r_1||\cdots||r_{n'}$, the prover chooses a string $x \in \{0,1\}^n$. For $i \in [n']$, the prover then sets $\mathsf{URS}_i := r_i \oplus x$ and runs the prover of the underlying NIZK proof system on the input statement, witness and URS_i, to produce proof π_i. The prover then sends $x, \pi_1, \ldots, \pi_{n'}$ to the verifier. For $i \in [n']$, the verifier recomputes $\mathsf{URS}_i := r_i \oplus x$ and runs the verifier of the underlying NIZK proof system on URS_i, π_i. If all the proofs accept, then the verifier accepts; otherwise, it rejects.

To prove soundness of the above proof system, a counting argument is employed. Specifically, fix any statement st that is not in the language. Since the underlying NIZK is statistically sound, the number of "bad" URS's for which there exists a proof π that accepts for st is small; say the fraction of "bad" URS's is at most $1/2$. This means that for a fixed statement st not in the language and a fixed x, the probability over random choice of $r_1, \ldots, r_{n'}$ that there exists an accepting proof π_i relative to each $\mathsf{URS}_i, i \in [n']$ is at most $2^{-n'}$. Taking a union bound over all possible choices for x, we have that for a fixed st, the probability over choice of $r_1, \ldots, r_{n'}$ that there *exists* an x of length n for which there exists an accepting proof relative to each $\mathsf{URS}_i, i \in [n']$ is at most $2^{n-n'}$. Setting $n' = 2n$ provides us with negligible statistical soundness in n.

On the other hand, to prove witness indistinguishability, one proceeds via a hybrid argument. In the original hybrid, witness w_1 is used for each of the n' number of honestly generated proofs $\pi_1, \ldots, \pi_{n'}$. In the final hybrid, witness w_2 is used for each of the n' number of honestly generated proofs $\pi_1, \ldots, \pi_{n'}$. In each intermediate hybrid, we switch from honestly generating a proof using w_1 to using w_2. Indistinguishability of intermediate hybrids is proved by showing that an efficient distinguisher between the hybrids implies an efficient distinguisher between real and simulated proofs of the underlying NIZK system. Specifically, a *reduction* is constructed as follows: Given the verifier's string $r = r_1 || \cdots || r_{n'}$ and a real or simulated URS/proof pair (URS^*, π^*), the reduction sets x such that $\mathsf{URS}_i = x \oplus r_i = \mathsf{URS}^*$. The reduction then runs the honest prover with w_2 for the first $i-1$ proofs, runs the honest prover with w_1 for the last $n'-i$ proofs, and embeds π^* in the i-th location. The reduction then applies the distinguisher between Hybrids $i-1$ and i to the resulting transcript, and outputs whatever it does. Since a distinguisher between Hybrids $i-1$ and i must either distinguish the above when (URS^*, π^*) were generated using the honest prover and w_1 versus using the simulator or when (URS^*, π^*) were generated using the honest prover and w_2 versus using the simulator, the above reduction succeeds in one of those cases. If one of the cases succeeds, we obtain a contradiction to the zero knowledge property.

Note that for the soundness argument for the ZAP, soundness against unbounded provers in the underlying NIZK is crucial since we use a counting argument based on the number of "bad" URS's for which there exists an accepting proof of the false statement. Furthermore, the fact that the prover in the underlying NIZK is *efficient* is crucial for the witness indistinguishability argument. The reason can be seen from the sketch of the hybrid argument above. Specifically, we will have a hybrid in which we reduce to the zero knowledge of the underlying NIZK (note that the zero knowledge must always be *computational*, since we require the soundness to be statistical). This means that existence of a distinguisher for consecutive hybrids must imply a ZK distinguisher, and the ZK distinguisher that is constructed, given an efficient distinguisher for consecutive hybrids, must be efficient. But in the approach outlined above, to generate the correct hybrid distributions for the efficient distinguisher, we must run the honest prover with witness w_2 for the first $i-1$ number of proofs and run the honest prover with witness w_1 for the last $n'-i$ number of proofs. This cannot be done efficiently if the honest prover is inefficient. An immediate thought would be to use non-uniform advice to hardcode all the proofs except the i-th proof into the ZK distinguisher. However, this does not work because $\mathsf{URS}_{i'}$ for $i' \neq i$ *depends* on URS^*, which is part of the input to the ZK distinguisher. Specifically, on input (URS^*, π^*), x is set to $\mathsf{URS}^* \oplus r_i$ and only once x is fixed do we learn $\mathsf{URS}_{i'} := r_{i'} \oplus x$ for $i' \neq i$. So we cannot know the URS's $\mathsf{URS}_{i'}, i' \neq i$ ahead of time. In this case, we cannot hope to hardcode the proofs $\pi_{i'}$ as non-uniform advice.

We will resolve this issue and show that non-uniform advice *can* help in our setting, by allow limited pairwise dependency across the URSs, instead of imposing pairwise independence. Specifically, our construction leverages the notion of a *design*, introduced by Nisan and Wigderson in their seminal work [NW88]. A *design* with parameters (l, n, c, n') is a set of n' number of sets $\mathcal{S}_1, \ldots, \mathcal{S}_{n'}$, where each $\mathcal{S}_i, i \in [n']$ is a subset of $[l]$ and has size $|\mathcal{S}_i| = n$. Moreover for every pair $i, j \in [n']$, $i \neq j$, it holds that $|\mathcal{S}_i \cap \mathcal{S}_j| \leq c$. It is known how to construct designs with $l = n^2$, constant c and $n' := n^c$ (see e.g. [NW88]). Let us see how a design with parameters $(l = n^2, n, c = 3, n' = n^3)$ can be used to resolve our problems above. Upon receiving string $r = r_1 || \cdots || r_{n'}$ from the verifier, we now allow the prover to choose a bit string $x = [x_j]_{j \in [l]}$ of length l. URS_i is then defined as $r_i \oplus [x_j]_{j \in \mathcal{S}_i}$, where $[x_j]_{j \in \mathcal{S}}$ for a set $\mathcal{S} \subseteq [l]$ denotes the substring of x corresponding to the positions $j \in \mathcal{S}$ and $\mathcal{S}_i$ is the corresponding set in the design. Now, soundness is ensured by the same argument as above (i.e. via a union bound), since

$2^{-n'} \cdot 2^l = 2^{-n^3} \cdot 2^{n^2} = 2^{-n^3+n^2}$ is negligible in n. Furthermore, since for each pair $i, j \in [n']$, $i \neq j$, it holds that $|\mathcal{S}_i \cap \mathcal{S}_j| \leq 3$, we can use the following proof strategy to argue indistinguishability of consecutive hybrids: In the i-th hybrid, we fix the string $[x_j]_{j \notin \mathcal{S}_i}$ at random. We then generate $n' - 1$ truth tables with constant input length. The input to the i'-th truth table ($i' \in [n'], i' \neq i$) will be at most 3 bits, corresponding to $[x_j]_{j \in \mathcal{S}_{i'} \cap \mathcal{S}_i}$. For $i' < i$, the output of the truth table $T_{i'}$ will be a proof $\pi_{i'}$ that is honestly computed using witness w_2 and $\mathsf{URS}_{i'} = [x_j]_{j \in \mathcal{S}_{i'}}$ For $i' > i$, the output of the truth table $T_{i'}$ will be a proof $\pi_{i'}$ that is honestly computed using witness w_1 and $\mathsf{URS}_{i'} = [x_j]_{j \in \mathcal{S}_{i'}}$. Note that since everything is fixed (including all the bits of $[x_j]_{j \in \mathcal{S}_{i'}}$ except for $[x_j]_{j \in \mathcal{S}_{i'} \cap \mathcal{S}_i}$), each truth table can be computed by an NC^0 circuit.

Now, given a real or simulated URS/proof pair (URS^*, π^*), the reduction will set $[x_j]_{j \in \mathcal{S}_i}$ such that $\mathsf{URS}_i = [x_j]_{j \in \mathcal{S}_i} \oplus r_i = \mathsf{URS}^*$. The reduction will then use the truth table $T_{i'}$ to generate proof $\pi_{i'}$ for $i' \neq i$, and will embed π^* in the i-th location. The reduction will then evaluate the distinguisher D (represented as a poly-sized circuit) on the resulting transcript and output whatever it outputs. Note that the reduction can now be represented as a poly-sized circuit and note that it outputs exactly the correct distribution to the distinguisher. Thus, an efficient distinguisher for intermediate hybrids yields a poly-sized circuit that breaks the zero knowledge property of the underlying NIZK, resulting in contradiction.

NIZK from ZAPs with inefficient provers. The standard transformation from ZAPs to NIZK, involves the well-known FLS technique [FLS99] to go from witness indistinguishability to zero knowledge. In this technique, instead of proving the statement s, the honest prover proves $s \vee s'$. s' is a statement for which only the simulator can prove using a special trapdoor. Thus, soundness is maintained. For zero knowledge, the simulator now runs the honest prover with the witness for s'. Due to witness indistinguishability, the honest and simulated proofs are indistinguishable. In the ZAP to NIZK setting, the simulator gets its trapdoor by including in the URS outputs of a PRG. A randomly generated uniform random string will not be in the image of the PRG. However, the simulator URS will be pseudorandom and the simulator's trapdoor will be the seed to the PRG that generates the URS. Clearly, the problem with using the FLS technique is that the zero knowledge simulator must be polynomial-time, whereas the honest prover in the ZAP construction is inefficient. Thus, the above approach does not yield an efficient simulator. To solve this problem, we show how to construct a oNIZK for the circuit SAT language by combining together proofs for many sub-languages. Now there will be only a small number of choices for the inputs of each sublanguage. The simulator will hardwire proofs for all choices of inputs. When the statement to be proven is received, the simulator will choose from among the various proofs corresponding to the possible inputs to each sublanguage and will combine them together to construct a single simulated proof. The techniques used are very similar to those needed to obtain fine-grained NIZK with uniform soundness in the standard model. We therefore refer the reader to the paragraph on "Achieving NIZK with uniform soundness in the standard model" below for more details.

Fine-Grained ZAPs. As discussed previously, fine-grained ZAPs relative to a class $\mathcal{F}$ are ZAPs that have a poly-time prover and provide witness indistinguishability against class $\mathcal{F}$ that is conjectured to not contain P. The same difficulty of converting a single-theorem fine-grained NIZK in the common random string model into a ZAP arises as above. Luckily, if circuits $f \in \mathcal{F}$ composed with NC^0 circuits are also in $\mathcal{F}$, then the same proof as above can work (since the reduction sketched above can be implemented with a NC^0 circuit. Thus, given a non-adaptive, fine-grained NIZK in the URS model against NC^1, we obtain a fine-grained ZAP relative to NC^1.

Fine-Grained NIZK in uniform random string (URS) model. We first modify a construction of [AR16] in the CRS model to yield a construction in the URS model. This is done by observing that a random string is a good CRS for the construction of [IK00] with probability 1/2 (which follows from the fact that randomized encodings of [IK00] are "balanced"). We then construct a URS by sampling many reference strings at random, and having the prover either prove that the reference string is invalid or provide a proof of the statement relative to the reference string. Note that this yields a construction with a poly-time prover and provides *statistical*-zero knowledge as well as soundness against unbounded provers. However,

this construction only allows proving statements for languages that have randomized encodings (such as languages in $\oplus L/\text{poly}$). We would like to obtain a proof system for all languages in NP, while sacrificing the statistical zero knowledge property and obtaining a fine-grained NIZK with poly-time prover against the class NC^1. It turns out that to obtain this, we can use the fact that, assuming $\oplus L/\text{poly} \neq \mathsf{NC}^1$, there exist "commitments" with the following properties: (1) Commitments can be constructed in the class NC^1. (2) Given a commitment, extracting the committed value can be performed in the class $\oplus L/\text{poly}$ (i.e. the decision problem $\mathcal{L}_{det}$ which, given a commitment *com* outputs 1 if it is a commitment to 1 is in $\oplus L/\text{poly}$). (3) Commitments are hiding against a NC^1 adversary. Such commitments can be easily constructed by computing the randomized encoding of a "canonical" 0 (resp. 1) input to commit to 0 (resp. 1). Now, using the fact that $\oplus L/\text{poly}$ is closed under negation, disjunction and conjunction (see [BG99]), we can use the statistical-zero knowledge NIZK in the URS model for languages in $\oplus L/\text{poly}$ to obtain a fine-grained NIZK in the URS model against NC^1 for all of NP as follows: Given a circuit-SAT instance $\mathcal{C}$, where $\mathcal{C}$ is a circuit consisting of NAND gates and we assume that it has z wires. the prover will commit to the values of all the wires of $\mathcal{C}$ for some satisfying assignment. This commitment will be performed using the "commitment" scheme described above. The prover will then prove that the sequence of "commitments" $com_1, \ldots, com_z$ is in the language $\mathcal{L}_{\mathcal{C}}$, where $com_z \in \mathcal{L}_{det}$, and for each NAND gate, with input wires i, j and output wire k, com_i, com_j, com_k are commitments to valid inputs/outputs for a NAND gate (i.e. $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$). Since $\mathcal{L}_{\mathcal{C}}$ will consist of negation/conjunction/disjunction of languages in $\oplus L/\text{poly}$ and since $\oplus L/\text{poly}$ is closed under negation/conjunction/disjunction, we have that $\mathcal{L}_{\mathcal{C}} \in \oplus L/\text{poly}$. Moreover, given $com_1, \ldots, com_z$, we can simulate a proof in NC^1 (using the simulator for the NIZK for languages in $\oplus L/\text{poly}$), indicating that the NIZK provides zero knowledge against NC^1.

We also present an alternative construction, which will be useful in some of the applications below. Here, the prover commits to the wire values $(com_1, \ldots, com_z)$, then individually proves statements about subsets of commitments, with each proof utilizing a separate URS. Specifically, one URS is used to prove that $com_z \in \mathcal{L}_{det}$ and for each gate with input wires i, j and output wire z, a separate URS is used to prove that $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$.

Achieving NIWI in the standard model. Assuming standard derandomization assumptions, it was shown in [BOV07] how to go from ZAPs to NIWI. This same transformation still works in both the unbounded prover and fine-grained settings. We note that in the fine-grained setting, we must apply the transformation while ensuring that the verifier remains in complexity class NC^1. However, evaluation of the hitting set generator may not be possible in NC^1. Therefore, the Prover must evaluate the hitting set generator to obtain the URS's and then provide a tableau for the verifier to verify that the computation was performed correctly. Note that this verification can be implemented in NC^1, even if evaluation of the hitting set generator itself is not in NC^1.

Achieving NIZK with uniform soundness in the standard model. Given the NIWI without CRS, we again employ the FLS technique [FLS99] to obtain NIZK without CRS with soundness against uniform adversaries. Specifically, the prover will prove an OR of two statements: The first statement is the statement that is actually to be proven, while the second statement is that two commitments provided by the prover are commitments to values that collide under some keyless collision resistant hash function. A uniform adversary will not be able to find a collision and will thus be forced to provide a proof for the first statement. The ZK simulator will receive the collision as non-uniform advice and will create a simulated proof by proving the second statement. In the following, we focus on using the above template to construct a simulator for the fine-grained NIZK with uniform soundness. This task is more complex than constructing a simulator in the regular NIZK case, since now the simulator is restricted to be in the class NC^1. This captures the fact that, informally, even an NC^1 adversary learns no more about the witness of an NP statement after seeing a proof than what it knew beforehand.

We now assume existence of a uniform collision resistant hash function h. Let $\mathcal{C}_h$ be the circuit that takes two inputs x_1, x_2 and outputs 1 if x_1, x_2 form a valid collision in h. On input Circuit SAT circuit $\mathcal{C}$, the prover now proves circuit satisfiability of the circuit $\mathcal{C}'$, where $\mathcal{C}'$ is defined as follows: $\mathcal{C}'$ takes public input

$\mathsf{desc}(\mathcal{C})$, which is a description of the circuit $\mathcal{C}$, and private input x. $\mathcal{C}'$ outputs 1 on input $(\mathsf{desc}(\mathcal{C}), x)$ if and only if x is a satisfying assignment of all wire values (including intermediate ones) in $\mathcal{C}$ or x is a satisfying assignment of all wire values (again, including intermediate ones) for $\mathcal{C}_h$. Note that $\mathcal{C}'$ is a NC^1 circuit. The prover (1) "commits" to a satisfying assignment for $\mathcal{C}'$ using the scheme described above (i.e. the output wire is a commitment to 1 and all NAND gates are evaluated correctly) and (2) proves that the commitments corresponding to the public input wires of $\mathcal{C}$ are consistent with $\mathsf{desc}(\mathcal{C})$. Note that each of (1) and (2) are composed of individual proofs of statements in languages contained in $\oplus L/\,\mathrm{poly}$, which each use a different CRS and each involve a constant number of committed values.

Towards constructing a NC^1 simulator, note that even given a satisfying assignment for $\mathcal{C}_h$ (i.e. colliding inputs and a tableau of the computation), not all wire values for a satisfying assignment of the circuit $\mathcal{C}'$ are known ahead of time, since the public input $\mathsf{desc}(\mathcal{C})$ (the circuit-SAT circuit) is not yet known. However, in a preprocessing stage, for each wire i, one can commit to both a 0 and 1 value, yielding com_i^0, com_i^1. Then, for each public input wire i, one can prove validity of both possibilities, resulting in proof $\pi_{in,i}^0$ corresponding to committed wire value com_i^0 and $\pi_{in,i}^1$ corresponding to committed wire value com_i^1. Furthermore, for each NAND gate, with input wires i, j and output wire k, one can prove validity of all the four possible valid triples, resulting in proofs $\pi_{gate,i,j,k}^{0,0}$ corresponding to committed wire values $(com_i^0, com_j^0, com_k^1)$, $\pi_{gate,i,j,k}^{0,1}$ corresponding to committed wire values $(com_i^0, com_j^1, com_k^1)$, $\pi_{gate,i,j,k}^{1,0}$ corresponding to committed wire values $(com_i^1, com_j^0, com_k^1)$, $\pi_{gate,i,j,k}^{1,1}$ corresponding to committed wire values $(com_i^1, com_j^1, com_k^0)$. Note that a separate CRS is used for each proof of validity, as in the alternate NIZK construction in the common random string model, described above. All of the above can be hardwired into the simulator. Then, when the public input $\mathsf{desc}(\mathcal{C})$ is received, the simulator knows a satisfying assignment (since it knows a satisfying assignment for $\mathcal{C}_h$) as well as all the corresponding values of the wires (since $\mathcal{C}' \in \mathsf{NC}^1$. The simulator can then choose the corresponding commitment com_i^0 or com_i^1 for the i-th wire, and once all the wires are selected, can choose the corresponding proofs $\pi_{in,i}^0$ or $\pi_{in,i}^1$ for the i-th public input wire and $\pi_{gate,i,j,k}^{0,0}$, $\pi_{gate,i,j,k}^{0,1}$, $\pi_{gate,i,j,k}^{1,0}$, or $\pi_{gate,i,j,k}^{1,1}$ for each gate, corresponding to the correct setting of wires. Put all together, the commitments and proofs for each public input wire and gate yield an accepting proof, that is indistinguishable (due to the witness indistinguishability property) from an honestly generated proof.

1.2 Related Work

Zero Knowledge Zero knowledge (ZK) proofs were introduced by Goldwasser, Micali, and Rackoff [GMR89]. Since its introduction, ZK proof systems and its variants have been studied with great interest. Some of the notable results related to ZK proofs are – [GMW91] which showed ZK proofs exist for all languages in NP, and [GO94] which showed that interaction is crucial for achieving zero knowledge property in case of non-trivial languages. Specifically, [GO94] showed that if for language $\mathcal{L}$, 2-message ZK proof system exists then $\mathcal{L} \in \mathsf{BPP}$. The research aimed at minimizing the interaction has since relied on either constructing Non-Interactive Zero Knowledge proof systems (NIZKs) with the help of trusted setup assumptions such as uniform random string (URS) [BFM88] or constructing non-interactive protocols with weaker security guarantees such as *non-interactive witness indistinguishability* (NIWI). Intuitively, witness indistinguishability ensures that the verifier does not learn which witness (out of multiple valid witnesses) is used by the prover to generate the proof. Dwork and Naor [DN07] showed introduced two-message, witness indistinguishable proof systems (ZAPs) and showed that ZAPs (in a no-CRS model) are equivalent to NIZKs in uniform *random* string (URS) model.

Zero Knowledge Primitives and Underlying Assumptions Blum et al. [BFM88], gave the first construction of NIZK in CRS model from number-theoretic assumptions (e.g. quadratic residuosity). Since then, NIZKs have been constructed in URS model from one-way permutations and certified trapdoor permutations [FLS99], whereas Lapidot and Shamir [LS91], constructed publicly verifiable NIZK from one-way permutations in URS model, Groth et al. [GOS06] constructed NIZK from DLIN assumption in URS model. Recently, Peikert and Shiehian [PS19] constructed NIZK from LWE assumption in URS model.

NIZKs have also been studied in other models [BG03, CV07, CCKV08], and models which consider preprocessing along with other assumptions such as one-way encryption schemes exist [DMP90], lattices (LWE) [KW18], and DDH/CDH [KNYY19]. Few of the other works on NIZKs include [BY96, Ps05, GS08, BL18, CCH+19, GJS19, ADKL19]. For more details on NIZK related research, we refer the interested readers to [WW14]

We now present an overview of research related to ZAP and NIWI systems. The notion of witness indistinguishable proofs was introduced by [FS90]. As discussed earlier, Dwork and Naor [DN07] introduced ZAP (two-message, witness indistinguishable proofs) and presented a construction in plain (no-CRS) model assuming the existence of certified trapdoor permutations. Barak et al. [BOV07] gave a construction of NIWI based on derandomization assumptions and certified trapdoor permutations (by derandomizing the verifier of [DN07] construction). Groth et al. [GOS12] constructed first non-interactive ZAP from DLIN assumption, whereas Bitansky and Paneth [BP15] showed a construction of ZAP based on indistinguishabliy obfuscation (iO) and one-way functions, and NIWI from iO and one-way permutations. Recently ZAP were constructed assuming quasi-polynomial hardness of DDH [JKKR17, KKS18], and quasi-polynomial hardness of LWE [BFJ+19, JJ19].

Fine-Grained Cryptography Fine-grained cryptography refers to construction of primitives which provide security guarantees against adversaries with sharper complexity bounds than simply "polynomial time." Both adversaries with *specific* polynomial runtime bounds (e.g. $\mathsf{TIME}[O(n^2)]$) and adversaries with *specific* parallel-time complexity (e.g. NC^1) have been considered under this moniker in the literature. In [DVV16] Degwekar et al. constructed primitives like one-way functions, pseudo-random generators, collision-resistant hash functions and public key encryption schemes based on well-studied complexity theoretic assumptions. Ball et al. [BRSV17, BRSV18] worst-case to average-case reduction for different type of fine-grained hardness problems and then extended their work to construct Proofs of Work. Campanelli and Gennaro [CG18] initiated the study of fine-grained secure computation by constructing a verifiable computation protocol secure against NC^1 adversaries based on worst-case assumptions. Recently, LaVigne et al. [LLW19] constructed a fine-grained key-exchange protocol.

Organization

The paper is organized as follows: We present preliminaries and definitions related to NIZK and oNIZK in Section 2. Section 3, presents a construction of ZAPs from oNIZK and oNIZK from ZAPs. The constructions of NIWI and oNIZK for polynomial verifiers are presented in Section 4. The constructions of ZAPs, NIWI, and oNIZK in the fine-grained setting are presented in Section 5.

2 Definitions

Definition 1. *Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be a class of circuits parameterized by n with input length $\ell(n)$. We say that two distribution ensembles $\{\mathcal{D}^0_n\}_{n\in\mathbb{N}}, \{\mathcal{D}^0_n\}_{n\in\mathbb{N}}$, with support $\{0,1\}^{\ell(n)}$, are indistinguishable by $\mathcal{F}$ if*

$$\max_{f_n\in\mathcal{F}_n} \left|\Pr[f_n(x)=1 \mid x\sim\mathcal{D}^0_n] - \Pr[f_n(x)=1 \mid x\sim\mathcal{D}^1_n]\right| \leq \mathsf{negl}(n).$$

Definition 2 (Fine-Grained Pseudorandom Generator (PRG) [DVV16]). *Let $\mathcal{H} = \left\{h_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}\right\}$ be a function family. $\mathcal{H}$ is a $\mathcal{F}$-fine-grained-pseudorandom generator (PRG) if* [8] *:*

- ***Computability**: For each n, h_n is deterministic.*
- ***Expansion**: $\ell(n) > n$ for all n.*

[8] Note that unlike the definition of [DVV16], our notion $\mathcal{F}$-PRG means $\mathcal{H}$ is secure against adversaries in class $\mathcal{F}$. Whereas, [DVV16] use notation $\mathcal{F}$-PRG to indicate that PRG $\mathcal{H}$ can be computed in class $\mathcal{F}$.

– ***Pseudorandomness**: For any $F = \{f_n : \{0,1\}^{\ell(n)} \to \{0,1\}\} \in \mathcal{F}$, and for all $n \in \mathbb{N}$:*

$$\left| \Pr_{x \leftarrow U_n}[f_n(h_n(x)) = 1] - \Pr_{x' \leftarrow U_{\ell(n)}}[f_n(x') = 1] \right| \leq \mathsf{negl}(n)$$

Definition 3 ($\mathcal{G}$-Extractable, $\mathcal{F}$-Fine-Grained Commitment Scheme). *A commitment scheme comprising of three algorithms* $(\mathsf{Commit}, \mathsf{Open}, \mathsf{Extract})$ *is called* $\mathcal{G}$-Extractable, $\mathcal{F}$-Fine-Grained Commitment Scheme *if the following hold:*

– $\mathsf{Commit} \in \mathcal{F}$ *and* $\mathsf{Open} \in \mathcal{F}$ *for class* $\mathcal{F}$.
– ***Correctness**: For all $n \in \mathbb{N}$ and for $b \in \{0,1\}$:*

$$\Pr[(com, d) \leftarrow \mathsf{Commit}(1^n, b) : \mathsf{Open}(com, d) = b] = 1$$

– ***Perfect Binding**: There does not exist a tuple (com, d, d') such that*

$$\mathsf{Open}(com, d) = 0 \wedge \mathsf{Open}(com, d') = 1.$$

– ***$\mathcal{F}$-Hiding**: For any $\mathsf{Open}^* \in \mathcal{F}$,*

$$\left| \Pr_{b \leftarrow \{0,1\}}[(com, d) \leftarrow \mathsf{Commit}(1^n, b) : \mathsf{Open}^*(c) = b] - \frac{1}{2} \right| \leq \mathsf{negl}(n)$$

– ***$\mathcal{G}$-Extractability**: There exists $\mathsf{Extract} \in \mathcal{G}$ such that for any string com,*

$$\mathsf{Extract}(com) = b \text{ iff } \exists d \text{ s.t. } \mathsf{Open}(com, d) = b.$$

An $\mathcal{F}$-Fine-Grained Commitment Scheme is the same as the above definition, but does not enjoy the $\mathcal{G}$-Extractability property.

2.1 NIZK and Fine-Grained NIZK in the URS Model

Definition 4 (Non-Interactive Proofs in the URS Model). *A pair of algorithms* $(\mathsf{Prover}, \mathsf{Verifier})$ *is called a non-interactive proof system in the URS model for a language $\mathcal{L}$ if the algorithm* $\mathsf{Verifier}$ *is deterministic polynomial-time, there exists a polynomial $p(\cdot)$ and a negligible function $\mu(\cdot)$ such that the following two conditions hold:*

– ***Completeness:** For every $x \in \mathcal{L}$*

$$\Pr[\mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi) = 1] \geq 1 - \mu(|x|).$$

– ***Soundness:** For every $x \notin \mathcal{L}$, every algorithm P^**

$$\Pr[\mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi' \leftarrow P^*(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi') = 1] \leq \mu(|x|).$$

Definition 5 (Non-Interactive Zero-Knowledge with Offline Simulation (oNIZK) in the URS Model). *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a non-interactive proof system in the URS model for the language $\mathcal{L}$. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is* non-adaptively zero-knowledge with offline simulation *in the URS model if there exists a distribution $\mathcal{D}_{\mathsf{Sim}}$ over polynomial-sized circuits* Sim *such that the following two distribution ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of $|x|$)*

$$\{(\mathsf{URS}, \pi) : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)\}_{x \in \mathcal{L}}$$
$$\{(\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}\}_{x \in \mathcal{L}}.$$

Relation to Zero Knowledge Notions in the Literature. We next discuss the relationship between Definition 5 and the notions of *witness hiding* (WH) and *weak zero knowledge* (WZK).

First, we show that Definition 5 in fact implies *witness hiding (WH).*

Let $\mathcal{D}$ be a distribution over statements $x \in \mathcal{L}$. Assume that $\mathcal{L}$ has witness relation $\mathcal{R}$, computable by a poly-size circuit, such that $x \in \mathcal{L}$ if and only if there exists a witness w such that $(x, w) \in \mathcal{R}$. We wish to show that if for all polynomial sized circuits C,

$$\Pr_{x\sim\mathcal{D}}[\mathcal{R}(x, C(x)) = 1] \leq \mathsf{negl}(|x|).$$

Then for all polynomial sized circuits C'

$$\Pr_{x\sim\mathcal{D}}[\mathcal{R}(x, C'(x, \mathsf{URS}, \pi)) = 1 : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)] \leq \mathsf{negl}(|x|).$$

Towards contradiction, assume that there exists a polynomial sized circuit C' such that

$$\Pr_{x\sim\mathcal{D}}[\mathcal{R}(x, C'(x, \mathsf{URS}, \pi)) = 1 : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)] \geq \mu(|x|),$$

where $\mu(\cdot)$ is non-negligible. Consider the following poly-size distinguisher D: On input (URS, π) and implicit x, run $C'(x, \mathsf{URS}, \pi)$ and obtain w. If $R(x, w) = 1$ output 1. Otherwise, output 0. Then we have that

$$\Pr_{x\sim\mathcal{D}}[D(\mathsf{URS}, \pi) = 1 : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)] \geq \mu(|x|).$$

By the above zero knowledge property (Def. 5), we have that

$$\Pr_{x\sim\mathcal{D},\mathsf{Sim}\sim\mathcal{D}_{\mathsf{Sim}}}[D(\mathsf{URS}', \pi') = 1 : (\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x)] \geq \mu'(|x|),$$

where μ' is non-negligible. But this implies that

$$\Pr_{x\sim\mathcal{D},\mathsf{Sim}\sim\mathcal{D}_{\mathsf{Sim}}}[R(x, C'(x, \mathsf{Sim}(x))) = 1] \geq \mu'(|x|),$$

and equivalently, that

$$\mathbb{E}_{x\sim\mathcal{D},\mathsf{Sim}\sim\mathcal{D}_{\mathsf{Sim}}}[R(x, C'(x, \mathsf{Sim}(x)))] \geq \mu'(|x|).$$

This, in turn, implies that there exists some Sim in the support of $\mathcal{D}_{\mathsf{Sim}}$ such that

$$\Pr_{x\sim\mathcal{D}}[\mathcal{R}(x, C'(x, \mathsf{Sim}(x))) = 1] \geq \mu'(|x|).$$

But the above implies a poly-sized circuit C (corresponding to C' composed with Sim) such that $\Pr_{x\sim\mathcal{D}}[R(x, C(x)) = 1]$ is non-negligible, which is a contradiction to the hardness of distribution $\mathcal{D}$ over language $\mathcal{L}$.

Recall that Weak Zero Knowledge (WZK) defined in ([DNRS99]), allows the simulator Sim to depend on the zero knowledge distinguisher D. Weak zero knowledge is known to be sufficient to argue security in For any cryptographic protocol that only aims to achieve indistinguishability-based security, weak zero knowledge is known to be sufficient, since the security reduction has access to an efficient distinguisher [JKKR17].

We also show that Definition 5 implies Weak Zero Knowledge (WZK) with an inverse-polynomial bound on distinguishing advantage. (This follows more or less directly from Lipton and Young's sparse min-max theorem [LY94], but we give full details here for completeness.)

Fix any distinguisher A, $\epsilon(n) = 1/n^c$ for some c, and define

$$D_x^{\mathsf{P}} := \{(\mathsf{URS}, \pi, x) : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{P}(\mathsf{URS}, x)\}$$
$$D_x^{\mathsf{Sim}} := \{(\mathsf{URS}', \pi', x) \leftarrow \mathsf{Sim}(x)\}, \forall \mathsf{Sim} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{Sim}})\}$$

Consider Sim', and corresponding distributions $\mathcal{D}_x^{\mathsf{Sim}'}$, that works by sampling $k \geq \frac{2n+\ln(2)}{2\epsilon(n)^2}$ i.i.d. samples $\mathsf{Sim}_1, \ldots, \mathsf{Sim}_k \leftarrow \mathcal{D}_{\mathsf{Sim}}$, before selecting a Sim_{i^*} uniformly at random from $\{\mathsf{Sim}_i\}_{i=1}^k$ and evaluating it on x.

Notice that by Chernoff/Hoeffding

$$\Pr_{(\mathsf{Sim}_1,\ldots,\mathsf{Sim}_k)\leftarrow\mathcal{D}_{\mathsf{Sim}'}}\left[|\sum_{i=1}^{k}\frac{\mathbb{E}[A(D_x^{\mathsf{Sim}_i})]}{k} - \mathbb{E}[A(D_x^{\mathsf{Sim}'})]| > \epsilon\right] < 2e^{-2k\epsilon(n)^2} \leq 2^{-2n}$$

Additionally, for any fixed $i \in [k]$ we have

$$\mathbb{E}_{\mathsf{Sim}_1,\ldots,\mathsf{Sim}_k\leftarrow\mathcal{D}_{\mathsf{Sim}'}}[A(D_x^{\mathsf{Sim}_i})] = \mathbb{E}_{\mathsf{Sim}\leftarrow\mathcal{D}_{\mathsf{Sim}}}[A(D_x^{\mathsf{Sim}})].$$

It follows that

$$\mathbb{E}_{\substack{i^*\leftarrow[k],\\ \mathsf{Sim}_1,\ldots,\mathsf{Sim}_k\leftarrow\mathcal{D}_{\mathsf{Sim}'}}}[A(D_x^{\mathsf{Sim}_{i^*}})] = \mathbb{E}_{\mathsf{Sim}\leftarrow\mathcal{D}_{\mathsf{Sim}}}[A(D_x^{\mathsf{Sim}})].$$

Thus by the triangle inequality we have that with probability at least $1-2^{-2n}$ over the choice of $S \leftarrow \mathcal{D}_{\mathsf{Sim}'}$, for any x

$$\begin{aligned}
|\mathbb{E}[A(D_x^S)] - \mathbb{E}[A(D_x^{\mathsf{P}})]| &= |\mathbb{E}[A(D_x^S)] \\
&\quad - (\mathbb{E}_{\substack{i^*\leftarrow[k],\\ \mathsf{Sim}_1,\ldots,\mathsf{Sim}_k\leftarrow\mathcal{D}_{\mathsf{Sim}'}}}[A(D_x^{\mathsf{Sim}_{i^*}}] - \mathbb{E}_{\mathsf{Sim}\leftarrow\mathcal{D}_{\mathsf{Sim}}}[A(D_x^{\mathsf{Sim}})]) \\
&\quad - \mathbb{E}[A(D_x^{\mathsf{P}})]| \\
&\leq |\mathbb{E}[A(D_x^S)] - \mathbb{E}_{i^*\leftarrow[k],\mathsf{Sim}_1,\ldots,\mathsf{Sim}_k\leftarrow\mathcal{D}_{\mathsf{Sim}'}}[A(D_x^{\mathsf{Sim}_{i^*}}]| \\
&\quad + |\mathbb{E}_{\mathsf{Sim}\leftarrow\mathcal{D}_{\mathsf{Sim}}}[A(D_x^{\mathsf{Sim}})] - \mathbb{E}[A(D_x^{\mathsf{P}})]| \\
&\leq \epsilon(n) + \eta(n)
\end{aligned}$$

So, for all x we have that $\Pr[|\mathbb{E}[A(D_x^S)] - \mathbb{E}[A(D_x^{\mathsf{P}})]| > \epsilon + \eta] < 2^{-2n}$. It follows that

$$\Pr[\exists x : |\mathbb{E}[A(D_x^S)] - \mathbb{E}[A(D_x^{\mathsf{P}})]| > \epsilon + \eta] \leq \sum_x \Pr[|\mathbb{E}[A(D_x^S)] - \mathbb{E}[A(D_x^{\mathsf{P}})]| > \epsilon + \eta] < \frac{2^n}{2^{2n}} = 2^{-n}$$

So, $\Pr[\forall x : |\mathbb{E}[A(D_x^S)] - \mathbb{E}[A(D_x^{\mathsf{P}})]| \leq \epsilon + \eta] > 1 - 2^{-n}$. In other words, the probability A has advantage at most $\epsilon + \eta$ (on any x) is bounded away from zero. By an averaging argument, it follows that there must exist some *fixed* choice of $\mathsf{Sim}_1, \ldots, \mathsf{Sim}_k$ that satisfies this bound on distinguishing advantage for A.

Definition 6 (Fine-Grained Non-Interactive Proofs in the URS Model). *A pair of algorithms* $(\mathsf{Prover}, \mathsf{Verifier})$ *is called a $\mathcal{F}$-fine-grained non-interactive proof system in the URS model for a language $\mathcal{L}$ if the algorithm* Prover *is polynomial-time, (uniformly generated)* $\mathsf{Verifier} \in \mathcal{F}_{|x|}$ *(*$\mathsf{Verifier}$ *can be uniformly generated), there exists a polynomial $p(\cdot)$ and a negligible function $\mu(\cdot)$ such that the following two conditions hold:*

- ***Completeness:*** *For every $x \in \mathcal{L}$*

$$\Pr[\mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi) = 1] \geq 1 - \mu(|x|).$$

- ***Soundness:*** *For every $x \notin \mathcal{L}$, every algorithm P^**

$$\Pr[\mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi' \leftarrow P^*(x, \mathsf{URS}) : \mathsf{Verifier}(x, \mathsf{URS}, \pi') = 1] \leq \mu(|x|).$$

Definition 7 (Fine-Grained Non-Interactive Zero-Knowledge in the URS Model). *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a $\mathcal{F}$-fine-grained non-interactive proof system in the URS model for the language $\mathcal{L}$. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is a $\mathcal{F}$-fine-grained* non-adaptively zero-knowledge *in the URS model if there exists a randomized circuit* Sim *in $\mathcal{F}$ such that the following two distribution ensembles are computationally indistinguishable by circuits in $\mathcal{F}$ (when the distinguishing gap is a function of $|x|$)*

$$\{(\mathsf{URS}, \pi) : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)\}_{x\in\mathcal{L}}$$
$$\{(\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x)\}_{x\in\mathcal{L}}.$$

We say that a fine-grained non-interactive proof system in the URS model is a *statistical* NIZK protocol (or alternatively *achieves statistical zero knowledge*) if the above distribution ensembles are statistically close.

Definition 8 (Fine-Grained Non-Interactive Zero-Knowledge with Offline Simulation (oNIZK) in the URS Model). *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a* $\mathcal{F}$*-fine-grained non-interactive proof system in the URS model for the language* $\mathcal{L}$*. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is a* $\mathcal{F}$*-fine-grained* non-adaptively zero-knowledge with offline simulation *in the URS model if there exists a distribution* $\mathcal{D}_{\mathsf{Sim}}$ *over circuits in* $\mathcal{F}$ *such that the following two distribution ensembles are computationally indistinguishable by circuits in* $\mathcal{F}$ *(when the distinguishing gap is a function of* $|x|$*)*

$$\{(\mathsf{URS}, \pi) : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)\}_{x \in \mathcal{L}}$$
$$\{(\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}\}_{x \in \mathcal{L}}.$$

Note that by the same argument as above, our fine-grained NIZK definition (for $\mathcal{F} = \mathsf{NC}^1$) implies witness hiding and weak zero knowledge with inverse-polynomial distinguishing advantage. Specifically, for the witness hiding case: Let $\mathcal{D}$ be a distribution over statements $x \in \mathcal{L}$. Assume that $\mathcal{L}$ has witness relation $\mathcal{R}$ such that $x \in \mathcal{L}$ if and only if there exists a witness w such that $(x, w) \in \mathcal{R}$. Note that WLOG we can assume that $\mathcal{R} \in \mathsf{NC}^1$. Assume that for all circuits $C \in \mathsf{NC}^1$,

$$\Pr_{x \sim \mathcal{D}}[R(x, C(x)) = 1] \leq \mathsf{negl}(|x|).$$

Then we have that for all circuits $C' \in \mathsf{NC}^1$

$$\Pr_{x \sim \mathcal{D}}[R(x, C'(x, \mathsf{URS}, \pi)) = 1 : \mathsf{URS} \leftarrow \{0,1\}^{p(|x|)}; \pi \leftarrow \mathsf{Prover}(\mathsf{URS}, x)] \leq \mathsf{negl}(|x|).$$

The definitions related to witness indistinguishability, ZAP, NIWI and their fine-grained versions are presented in Section **??** of supplementary material due to lack of space.

2.2 Witness Indistinguishability and Fine-Grained Witness Indistinguishability

Definition 9 (Witness Indistinguishability). *A proof system* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle$ *for a language* $\mathcal{L}$ *is witness-indistinguishable if for any polynomial time* V^**, for all* $x \in \mathcal{L}$*, for all* $w_1, w_2 \in w(x)$*, and for all auxiliary inputs* z *to* V^**, the distribution on the views of* V^* *following an execution* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle(x, w_1, z)$ *is indistinguishable from the distribution on the views of* V^* *following an execution* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle(x, w_2, z)$ *to a non-uniform probabilistic polynomial-time distinguisher receiving one of the above transcripts as well as* (x, w_1, w_2, z)*.*

Definition 10 ($\mathcal{F}$-fine-grained Witness Indistinguishability). *A proof system* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle$ *for a language* $\mathcal{L}$ *is* $\mathcal{F}$*-fine-grained witness-indistinguishable if* Prover *is polynomial-time,* $\mathsf{Verifier}$ *is in the class* $\mathcal{F}$ *and for any* $V^* \in \mathcal{F}$*, for all* $x \in \mathcal{L}$*, for all* $w_1, w_2 \in w(x)$*, and for all auxiliary inputs* z *to* V^**, the distribution on the views of* V^* *following an execution* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle(x, w_1, z)$ *is indistinguishable from the distribution on the views of* V^* *following an execution* $\langle \mathsf{Prover}, \mathsf{Verifier} \rangle(x, w_2, z)$ *to a non-uniform distinguisher in class* $\mathcal{F}$ *receiving one of the above transcripts as well as* (x, w_1, w_2, z)*.*

2.3 ZAPs and Fine-Grained ZAPs

Definition 11 (ZAP). *A ZAP is a 2-round (2-message) protocol for proving membership of* $x \in \mathcal{L}$*, where* $\mathcal{L}$ *is a language in* NP*. Let the first-round (verifier to prover) message be denoted* ρ *and the second-round (prover to verifier) response be denoted* π *satisfying the following conditions:*

- ***Public Coins:*** *There is a polynomial* $p(\cdot)$ *such that the first round messages form a distribution on strings of length* $p(|x|)$*. The verifier's decision whether to accept or reject is a polynomial time function of* $x, \rho,$ *and* π *only.*

- ***Completeness:*** *Given x, a witness $w \in w(x)$, and a first-round ρ, the prover generates a proof π that will be accepted by the verifier with overwhelming probability over the choices made by the prover and the verifier.*
- ***Soundness:*** *With overwhelming probability over the choice of ρ, there exists no $x' \notin \mathcal{L}$ and second round message π such that the verifier accepts (x', ρ, π).*
- ***Witness-Indistinguishability:*** *Let $w, w' \in w(x)$ for $x \in L$. Then $\forall \rho$, the distribution on π when the prover has input (x, w) and the distribution on π when the prover has input (x, w') are nonuniform probabilistic polynomial time (in $|x|$) indistinguishable, even given both witnesses w, w'.*

Definition 12 ($\mathcal{F}$-fine-grained ZAP). *A $\mathcal{F}$-fine-grained ZAP is a 2-round (2-message) protocol for proving membership of $x \in \mathcal{L}$, where $\mathcal{L}$ is a language in* NP*. Let the first-round (verifier to prover) message be denoted ρ and the second-round (prover to verifier) response be denoted π satisfying the following conditions:*

- ***Public Coins and Fine-Grained Verifier:*** *There is a polynomial $p(\cdot)$ such that the first round messages form a distribution on strings of length $p(|x|)$. The verifier's decision whether to accept or reject is a function of x, ρ, and π only, and is contained in $\mathcal{F}_{|x|}$.*
- ***Completeness:*** *Given x, a witness $w \in w(x)$, and a first-round ρ, the prover, running in time polynomial in $|x|$, can generates a proof π that will be accepted by the verifier with overwhelming probability over the choices made by the prover and the verifier.*
- ***Soundness:*** *With overwhelming probability over the choice of ρ, there exists no $x' \notin \mathcal{L}$ and second round message π such that the verifier accepts (x', ρ, π).*
- ***$\mathcal{F}$-fine-grained Witness-Indistinguishability:*** *Let $w, w' \in w(x)$ for $x \in L$. Then $\forall \rho$, the distribution on π when the prover has input (x, w) and the distribution on π when the prover has input (x, w') are indistinguishable to nonuniform algorithms in the class $\mathcal{F}_{|x|}$, even given both witnesses w, w'.*

2.4 NIWI and Fine-Grained NIWI

Definition 13 (NIWI). *A NIWI is a non-interactive protocol for proving membership of $x \in \mathcal{L}$, where $\mathcal{L}$ is a language in* NP*. A single message π is sent from the prover to the verifier.*

- ***Efficient Verifier:*** *The verifier's decision whether to accept or reject is a polynomial time function of x and π only.*
- ***Completeness:*** *Given x and a witness $w \in w(x)$, the prover generates a proof π that will be accepted by the verifier.*
- ***Soundness:*** *There exists no $x' \notin \mathcal{L}$ and message π such that the verifier accepts (x', π).*
- ***Witness-Indistinguishability:*** *Let $w, w' \in w(x)$ for $x \in L$. Then the distribution on π when the prover has input (x, w) and the distribution on π when the prover has input (x, w') are nonuniform probabilistic polynomial time (in $|x|$) indistinguishable, even given both witnesses w, w'.*

Definition 14 ($\mathcal{F}$-fine-grained NIWI). *A $\mathcal{F}$-fine-grained NIWI is a non-interactive protocol for proving membership of $x \in \mathcal{L}$, where $\mathcal{L}$ is a language in* NP*. A single message π is sent from the prover to the verifier.*

- ***Fine-Grained Verifier:*** *The verifier's decision whether to accept or reject is a function of the statement x and proof π only, and the verifier's circuit is contained in $\mathcal{F}_{|x|}$.*
- ***Completeness:*** *Given x, and a witness $w \in w(x)$ the prover, running in time polynomial in $|x|$, can generate a proof π that will be accepted by the verifier with overwhelming probability over the choices made by the prover and the verifier.*
- ***Soundness:*** *There exists no $x' \notin \mathcal{L}$ and message π such that the verifier accepts (x', π).*
- ***$\mathcal{F}$-fine-grained Witness-Indistinguishability:*** *Let $w, w' \in w(x)$ for $x \in L$. Then the distribution on π when the prover has input (x, w) and the distribution on π when the prover has input (x, w') are indistinguishable by the class $\mathcal{F} := \{\mathcal{F}_{|x|}\}_{|x| \in \mathbb{N}}$, even given both witnesses w, w'.*

2.5 NIZK and Fine-Grained NIZK without CRS and with uniform soundness

Definition 15 (Non-Interactive Proofs with uniform soundness). *A pair of algorithms* $(\mathsf{Prover}, \mathsf{Verifier})$ *is called a non-interactive proof system with uniform soundness* $T := T(|x|)$*, for a language* $\mathcal{L}$ *if the algorithm* $\mathsf{Verifier}$ *is deterministic polynomial-time, there exists a polynomial* $p(\cdot)$ *and a negligible function* $\mu(\cdot)$ *such that the following two conditions hold:*

- ***Completeness:*** *For every* $x \in \mathcal{L}$

$$\Pr[\pi \leftarrow \mathsf{Prover}(x) : \mathsf{Verifier}(x, \pi) = 1] \geq 1 - \mu(|x|).$$

- ***Soundness:*** *For every* $x \notin \mathcal{L}$*, every algorithm* P^* *running in uniform time* T*,*

$$\Pr[\pi' \leftarrow P^*(x) : \mathsf{Verifier}(x, \pi') = 1] \leq \mu(|x|).$$

Definition 16 (Non-Interactive Zero-Knowledge with Offline Simulation (oNIZK) in the standard model with uniform soundness). *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a non-interactive proof system with uniform soundness* $T := T(|x|)$ *for the language* $\mathcal{L}$*. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is* zero-knowledge with offline simulation *if there exists a distribution* $\mathcal{D}_{\mathsf{Sim}}$ *over polynomial-sized circuits* Sim *such that the following two distribution ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of* $|x|$*)*

$$\{\pi \leftarrow \mathsf{Prover}(x)\}_{x \in \mathcal{L}}$$
$$\{\pi' \leftarrow \mathsf{Sim}(x) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}\}_{x \in \mathcal{L}}.$$

As discussed previously, our NIZK definition above implies witness hiding, via the same argument.

Definition 17 (Fine-Grained Non-Interactive Proofs with uniform soundness). *A pair of algorithms* $(\mathsf{Prover}, \mathsf{Verifier})$ *is called a* $\mathcal{F}$*-fine-grained non-interactive proof system with uniform soundness for a language* $\mathcal{L}$ *if the algorithm* Prover *is polynomial-time, (uniformly generated)* $\mathsf{Verifier} \in \mathcal{F}_{|x|}$*, there exists a polynomial* $p(\cdot)$ *and a negligible function* $\mu(\cdot)$ *such that the following two conditions hold:*

- ***Completeness:*** *For every* $x \in \mathcal{L}$

$$\Pr[\pi \leftarrow \mathsf{Prover}(x, \mathsf{URS}) : \mathsf{Verifier}(x, \pi) = 1] \geq 1 - \mu(|x|).$$

- ***Soundness:*** *For every* $x \notin \mathcal{L}$*, every uniform, PPT algorithm* P^*

$$\Pr[\pi' \leftarrow P^*(x) : \mathsf{Verifier}(x, \pi') = 1] \leq \mu(|x|).$$

Definition 18 (Fine-Grained Non-Interactive Zero-Knowledge with Offline Simulation (oNIZK) in the standard model with uniform soundness). *Let* $(\mathsf{Prover}, \mathsf{Verifier})$ *be a* $\mathcal{F}$*-fine-grained non-interactive proof system with uniform soundness for the language* $\mathcal{L}$*. We say that* $(\mathsf{Prover}, \mathsf{Verifier})$ *is* $\mathcal{F}$*-fine-grained* zero-knowledge with offline simulation *if there exists a distribution* $\mathcal{D}_{\mathsf{Sim}}$ *over circuits in* $\mathcal{F}$ *such that the following two distribution ensembles are computationally indistinguishable by circuits in* $\mathcal{F}$ *(when the distinguishing gap is a function of* $|x|$*)*

$$\{\pi \leftarrow \mathsf{Prover}(x)\}_{x \in \mathcal{L}}$$
$$\{\pi' \leftarrow \mathsf{Sim}(x) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}\}_{x \in \mathcal{L}}.$$

As discussed previously, our fine-grained NIZK definition above implies witness hiding, via the same argument.

3 Equivalence of ZAPs and oNIZK

3.1 ZAPs from oNIZK

For our construction of ZAPs from oNIZK in the URS model, we will require a certain type of *design*, defined next and first used by Nisan and Wigderson in their derandomization of BPP [NW88].

Definition 19 (Design). *A* (l, n', n, c)-design *consists of sets* $\mathcal{S}_1, \ldots, \mathcal{S}_{n'} \subseteq [l]$ *such that the following hold:*

- *For each* $i \in [n']$, $|\mathcal{S}_i| = n$,
- *For each* i, i' *s.t.* $i \neq i'$, $|\mathcal{S}_i \cap \mathcal{S}_{i'}| \leq c$.

(l, n', n, c) designs are known to exist for $l := n^2$, constant $c \in \mathbb{N}$, and $n' := n^c$ [NW88].

Let $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ be a non-adaptive oNIZK in the URS model with unbounded prover that has soundness 1/2 or better. Let sets $\mathcal{S}_1, \ldots, \mathcal{S}_{n'} \subseteq [l]$ form a (l, n', n, c)-design, where $l := n^2$, $c := 3$, and $n' := n^3$.

Verifier's First Round Message: Recall that in the first round of a ZAP, the Verifier sends a random string r to the Prover.

Prover Algorithm: On input statement st, witness w, and random string $r = r_1 || \cdots || r_{n'}$ from the Verifier:

1. Choose bits $[x_j]_{j \in [l]}$ at random. For a set $\mathcal{S} \subseteq [l]$, let $[x_j]_{j \in \mathcal{S}}$ denote the substring of $[x_1, \ldots, x_l]$ corresponding to indeces in set $\mathcal{S}$.
2. For each $i \in [n']$, let $\mathsf{URS}_i = r_i \oplus [x_j]_{j \in \mathcal{S}_i}$, where each r_i has length n and each $|\mathcal{S}_i| = n$ (recall that the sets $\mathcal{S}_i$ are the sets of the design).
3. For $i \in [n']$, run $\mathsf{Prover}^{\mathsf{NIZK}}$ on input URS_i and witness w, outputting proof π_i.
4. Output $[\pi_i]_{i \in [n']}$ along with $[x_1, \ldots, x_l]$.

Verifier's Algorithm after the Second Round: Recall that the Verifier's first message is denoted r and that the verifier gets input statement st. After observing the Prover's message consisting of $[\pi_i]_{i \in [n']}$, $[x_1, \ldots, x_l]$, the Verifier does the following:

1. For $i \in [n']$, set $\mathsf{URS}_i = r_i \oplus [x_j]_{j \in \mathcal{S}_i}$
2. For $i \in [n']$, verify proof π_i relative to URS_i by running the verifier $\mathsf{Verifier}^{NIZK}$.
3. If all checks accept, then accept. Otherwise reject.

Theorem 11. *Assume* $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ *is a non-adaptive oNIZK proof system for NP with an inefficient prover in the URS model. Then the above construction is a ZAP for NP with an inefficient prover.*

Soundness Proof: We say that a URS is "bad" relative to a statement $\mathsf{st} \notin \mathcal{L}$ that is not in the language, if there exists an accepting proof relative to that URS (recall that the verifier is deterministic). For statement $\mathsf{st} \notin \mathcal{L}$ and fixed $[x_j]_{j \in [l]}$, the probability over choice of r that *every* URS_i, $i \in [n']$ is bad is at most $2^{-n'}$. Since there are at most 2^l choices for $[x_j]_{j \in [l]}$ (where $l := n^2$), the probability over random choice of r that there exists a setting of $[x_j]_{j \in [l]}$ such that each URS_i is bad is at most $2^{n^2} \cdot 2^{-n'}$. Since we have set $n' := n^3$, we have that that $2^{n^2} \cdot 2^{-n'} = 2^{-n^3 + n^2}$ is negligible.

Witness Indistinguishability Proof: We consider the following distributions:

Hybrid H^{w_1}: This is the real world distribution with statement st and witness w_1.

Hybrid H^{w_2}: This is the real world distribution with statement st and witness w_2.

To prove WI, we must show that for every malicious verifier V^*.

$$H^{w_1} \approx H^{w_2}.$$

Towards this goal, we define the following sequences of hybrid distributions:

Hybrid H^{i,w_1,w_2}, for $i \in [n']$: Proofs with $\mathsf{URS}_{i'}$ for $i' \leq i$ are honest proofs using w_2. Proofs with $\mathsf{URS}_{i'}$ for $i' > i$ are honest proofs using w_1.

Note that $H^{w_1} = H^{0,w_1,w_2}$ and $H^{w_2} = H^{n',w_1,w_2}$.

Claim. For $i \in [n']$,

$$H^{i-1,w_1,w_2} \approx H^{i,w_1,w_2}.$$

Proof. Consider the distribution $H^{*,i,w_1,w_2}(\mathsf{URS}^*, \pi^*)$, where a draw from the distribution is defined as follows:

- Run V^* to produce $r = r^1 || \cdots || r^{n'}$, sample $[x_j]_{j \in [l] \setminus \mathcal{S}_i}$
- Set $[x_j]_{j \in \mathcal{S}_i} := \mathsf{URS}^* \oplus r^i$.
- Set $\pi_i := \pi^*$.
- For each $i' \in [i-1]$, run the honest prover Prover^{NIZK} on witness w_2 and $\mathsf{URS}_{i'} = r^{i'} \oplus [x_j]_{j \in \mathcal{S}_{i'}}$ to obtain proof $\pi_{i'}$.
- For each $i' \in \{i+1, \ldots, n'\}$, run the honest prover Prover^{NIZK} on witness w_1 and $\mathsf{URS}_{i'} = r^{i'} \oplus [x_j]_{j \in \mathcal{S}_{i'}}$ to obtain proof $\pi_{i'}$.
- Output $[\pi_{i'}]_{i' \in [n']}$ and $x := [x_j]_{j \in [l]}$.

Note that when $(\mathsf{URS}^* = \mathsf{URS}_{\mathsf{honest}}, \pi^* = \pi_{w_1})$ (resp. $(\mathsf{URS}^* = \mathsf{URS}_{\mathsf{honest}}, \pi^* = \pi_{w_2})$) are generated as honest CRS/proofs with witness w_1 (resp. w_2), then $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_1})$ (resp. $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_2})$) is equivalent to H^{i-1,w_1,w_2} (resp. H^{i,w_1,w_2}). We must also have that $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_1})$ (resp. $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_2})$) is indistinguishable from $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{Sim}}, \pi_{\mathsf{Sim}})$ (where $\mathsf{URS}_{\mathsf{Sim}}, \pi_{\mathsf{Sim}}$ are generated by drawing a simulator from the oNIZK distribution and obtaining its output), since otherwise we obtain a non-uniform PPT adversary that breaks the zero knowledge of the underlying NIZK proof system. We will elaborate on how this indistinguishability is proved below. Assuming that this is the case, we conclude that H^{i-1,w_1,w_2} and H^{i-1,w_1,w_2} are indistinguishable, which completes the proof.

We now show that $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_1})$ (resp. $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_2})$) is indistinguishable from $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{Sim}}, \pi_{\mathsf{Sim}})$ (where $\mathsf{URS}_{\mathsf{Sim}}, \pi_{\mathsf{Sim}}$ are generated by drawing a simulator from the oNIZK distribution and obtaining its output). Towards contradiction, assume the existence of non-uniform PPT verifier V^* and non-uniform PPT distinguisher D distinguishing $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_1})$ (resp. $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{honest}}, \pi_{w_2})$) from $H^{*,i,w_1,w_2}(\mathsf{URS}_{\mathsf{Sim}}, \pi_{\mathsf{Sim}})$. Using V^*, D as above, we construct the following distribution over poly-sized circuits that receive as input (URS^*, π^*):

- Run V^* to produce $r = r^1 || \cdots || r^{n'}$, sample $[x_j]_{j \in [l] \setminus \mathcal{S}_i}$ uniformly at random as well as any auxiliary state state_{V^*}, which will be used by the distinguishing circuit D.
- **Hardwired values:**
  1. Statement s and witnesses w_1, w_2.
  2. Auxiliary state state_{V^*}.
  3. $r = r^1 || \cdots || r^{n'}$, $[x_j]_{j \in [l] \setminus \mathcal{S}_i}$.
  4. For each $i' \in [i]$, hardwire truthtable $T_{i'}$ that takes as input $[x_j]_{j \in \mathcal{S}_i \cap \mathcal{S}_{i'}}$ (at most 3 input bits) and outputs $\mathsf{URS}_{i'} = r_{i'} \oplus [x_j]_{j \in \mathcal{S}_{i'}}$, and proof $\pi_{i'}$ honestly computed with statement st and witness w_2.
  5. For each $i' \in \{i+1, \ldots, n'\}$, hardwire truthtable $T_{i'}$ that takes as input $[x_j]_{j \in \mathcal{S}_i \cap \mathcal{S}_{i'}}$ and outputs $\mathsf{URS}_{i'} = r_{i'} \oplus [x_j]_{j \in \mathcal{S}_{i'}}$, and proof $\pi_{i'}$ honestly computed with statement st and witness w_1.
- **Circuit Evaluation:** On input (URS^*, π^*), do the following:
  - **Embed** (URS^*, π^*)**:** Set $[x_j]_{j \in \mathcal{S}_i} := r_i \oplus \mathsf{URS}^*$. Set $\pi_i := \pi^*$.

- **Compute Honest Proofs:** Use the truthtables to compute $\mathsf{URS}_{i'}$ and $\pi_{i'}$ for all $i' \neq i$, where the i'-th truthtable $T_{i'}$ takes input $[x_j]_{j \in \mathcal{S}_i \cap \mathcal{S}_{i'}}$.
- **Output of Prover:** Combine the above two steps to obtain the Prover's message: $([\pi_{i'}]_{i' \in [n'']}, x := [x_j]_{j \in [l]})$.
- **Application of Distinguisher:** Apply D (which may require state_{V^*} as auxiliary input) to the transcript and output $D(r, [\pi_{i'}]_{i' \in [n'']}, x := [x_j]_{j \in [l]})$.

Note that since each of the truth tables $T_{i'}$ takes a constant number of input bits, and since all the truth tables can be evaluated in parallel, the above is a distribution over circuits corresponding to a (non-uniform) NC^0 circuit composed with the distinguisher D. When D is a poly-sized circuit, the resulting circuit drawn from the distribution is poly-sized. Moreover, the expected distinguishing probability of a circuit drawn from the above distribution is exactly equal to D's distinguishing probability (which is assumed to be non-negligible). But this contradicts the the zero knowledge property of the underlying oNIZK.

Note the same proof as above holds for the case of $\mathcal{F}$-fine-grained oNIZK, as long as the distribution defined above is a distribution over circuits contained in $\mathcal{F}$, whenever D is contained in $\mathcal{F}$. This holds when instantiating $\mathcal{F}$ with the class non-uniform NC^1 since, as discussed above, the depth of "Embed" + "Compute Honest Proofs" + "Output of Prover" is constant. So if the depth of D in the "Application of Distinguisher" is logarithmic, then the depth of the entire "Circuit Evaluation" is logarithmic. We therefore obtain the following theorem:

Theorem 12. *Assume $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ is a NC^1-fine-grained, non-adaptive oNIZK proof system in the URS model. Then the above construction is a NC^1-fine-grained ZAP.*

3.2 oNIZK from ZAPs

Assume the existence of a commitment scheme in the common random string model with commitments *com* and common random string w of length λ'. Assume a PRG G with stretch 1 (i.e. it takes inputs of length λ to outputs of length $\lambda + 1$). Existence of both of the above is implied by the existence of one-way functions. We require that the commitments are perfectly binding with all but negligible probability over choice of common random string. Let $\mathcal{L}_{com,1,w}$ be the language consisting of strings that can be opened to 1 relative to common random string w. Let $\mathcal{L}_{com,0,w}$ be the language consisting of strings that can be opened to 0 relative to common random string w. We consider the following language $\mathcal{L}_{gate}$: $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$ if and only if:

$$(com_i \in \mathcal{L}_{com_0} \wedge com_k \in \mathcal{L}_{com_1}) \vee (com_j \in \mathcal{L}_{com_0} \wedge com_k \in \mathcal{L}_{com_1}) \vee (com_i \in \mathcal{L}_{com_1} \wedge com_j \in \mathcal{L}_{com_1} \wedge com_k \in \mathcal{L}_{com_0}) .$$

In other words, $\mathcal{L}_{gate}$ is the language of strings that correspond to commitments of valid inputs/output of a NAND gate.

Let $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$ be the circuit that takes as input $s_1, \ldots, s_\lambda$ and outputs 1 if for all $i \in [\lambda]$, $G(s_i) = y_i$. and outputs 1 if $x_1 \neq x_2$ and $h(x_1) = h(x_2)$. Given circuits $\mathcal{C}$ and $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$, $\mathcal{C}'$ is defined as follows: $\mathcal{C}'$ takes public input $\mathsf{desc}(\mathcal{C})$, which is a description of the circuit $\mathcal{C}$, and private input x. $\mathcal{C}'$ outputs 1 on input $(\mathsf{desc}(\mathcal{C}), x)$ if and only if x is a satisfying assignment for $\mathcal{C}$ or x is a satisfying assignment for $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$.

Let $\Pi = (\mathsf{Prover}^{ZAP}, \mathsf{Verifier}^{ZAP})$ be a ZAP with inefficient prover for NP. We construct an oNIZK in the URS model for the circuit-SAT language as follows:

The URS has form $w||y_1|| \cdots ||y_\lambda||\mathsf{URS}_0||\mathsf{URS}_1|| \cdots ||\mathsf{URS}_{z'}$, where w is a string of length λ', each $y_i, i \in [\lambda]$ has bitlength $\lambda + 1$, URS_0 has length m, and $\mathsf{URS}_i, i \in [z']$ has length n, where λ is security parameter, n is the length of the random string sent by $\mathsf{Verifier}^{ZAP}$ for the language $\mathcal{L}_{gate}$ and m is the length of the random string sent by $\mathsf{Verifier}^{ZAP}$ for the language $\mathcal{L}_{com,1,w}$.

The Prover does as follows: On input instance $\mathcal{C}$, and a satisfying assignment x for $\mathcal{C}$, the Prover constructs $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$ as well as the transformed circuit $\mathcal{C}'$. The Prover now proves circuit satisfiability of $\mathcal{C}'$, which we assume to be a circuit composed entirely of NAND gates, which has z number of wires and z' number of gates.

1. Commit to the values of the wires corresponding to the satisfying assignment x for $\mathcal{C}$—which is also a satisfying assignment for $\mathcal{C}'$—producing commitments $com_1, \ldots, com_z$.
2. Open all the commitments corresponding to public input wires and prove that they are consistent with $\mathsf{desc}(\mathcal{C})$.
3. Run Prover^{ZAP} on input Verifier's message URS_0 to prove that $com_z \in \mathcal{L}_{com_1}$, obtaining proof π_0.
4. For the ℓ-th gate $\ell \in [z']$, with input wires i, j and output wire k, use Prover^{ZAP} on input Verifier's message URS_ℓ to prove that $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$, obtaining proof π_ℓ.
5. Finally, output $\pi_0 || \pi_1 || \cdots || \pi_{z'}$.

Theorem 13. *Assume the existence of one-way functions and assume that $\Pi = (\mathsf{Prover}^{ZAP}, \mathsf{Verifier}^{ZAP})$ is a ZAP system for NP with inefficient provers. Then the above construction is an oNIZK for NP in the URS model with inefficient prover.*

Proof. To prove zero knowledge with offline simulation, we must show a distribution $\mathcal{D}_{\mathsf{Sim}}$ over polynomial sized circuits Sim, such that a circuit drawn from this distribution, evaluated on input statement $\mathcal{C}$ produces a distribution over common random strings and proofs that is indistinguishable from real common random strings and proofs for polynomial sized distinguishers.

A draw from $\mathcal{D}_{\mathsf{Sim}}$ is defined as follows:

- Sample each of $s_1, \ldots, s_\lambda$ uniformly at random from $\{0,1\}^\lambda$.
- For $i \in [\lambda]$, set $y_i = G(s_i)$.
- Sample each of $\mathsf{URS}_0, \ldots, \mathsf{URS}_z$ uniformly at random.
- For each wire i of $\mathcal{C}'$, sample a commitment to 0 and a commitment to 1: (com_i^0, com_i^1), together .
- For each public wire i of $\mathcal{C}'$, produce the decommitments $\pi_{in,i}^0, \pi_{in,i}^1$ proving that $com_i^0 \in \mathcal{L}_{com_0}$ and that $com_i^1 \in \mathcal{L}_{com_1}$, respectively.
- For the output wire z of $\mathcal{C}'$, compute an honest proof π_{out} that $com_z^1 \in \mathcal{L}_{com_1}$.
- For each gate with input wires i, j and output wire k of $\mathcal{C}'$, compute 4 honest proofs $[\pi_{gate,i,j,k}^{b_1,b_2}]_{b_1,b_2 \in \{0,1\}}$ proving that $com_i^{b_1}, com_j^{b_2}, com_k^{1-b_1 \wedge b_2} \in \mathcal{L}_{gate}$, for $b_1, b_2 \in \{0,1\}$.
- **Hardwired Values:** A satisfying assignment $s_1, \ldots, s_\lambda$ for $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$ and $[com_i^0, com_i^1]_{i \in [z]}$, $(\pi_{in,i}^0, \pi_{in,i}^1)$, π_{out}, $[\pi_{gate,i,j,k}^{b_1,b_2}]_{i,j,k,b_1,b_2}$.
- **Circuit Evaluation:** On input $\mathsf{desc}(\mathcal{C})$, choose the appropriate public inputs corresponding to that input. Additionally, chose the private inputs corresponding to the satisfying assignment y. Let $b_{in}(i)$ denote the value of the i-th public input wire. Assume there are a total of z'' input wires. Using these, compute the values of all wires of $\mathcal{C}'$. Let $b(i)$ denote the value of the i-th wire of $\mathcal{C}'$. Output commitments $[com_i^{b(j)}]_{i \in [z]}$ and proofs $[\pi_{in,i}^{b_{in}(i)}]_{i \in [z'']}$, $[\pi_{gate,i,j,k}^{b(i),b(j)}]_{i,j,k}$.

Note that the outputted distribution is identical to an honest proof with witness corresponding to a satisfying assignment of $\mathcal{C}_{G,y_1,\ldots,y_\lambda}$. Thus, by the witness indistinguishability property of the proof system, the simulated proof is indistinguishable from the real proof.

Moreover, note that we still have soundness against computationally unbounded adversaries, since a random string $y_1, \ldots, y_\lambda$ will have the property that each y_i is in the image of G with probability at most $1/2^\lambda$.

Note the same construction/proof as above holds for the case of $\mathcal{F}$-fine-grained oNIZK, (with $\mathcal{F}$-fine-grained commitments replacing regular commitments and $\mathcal{F}$-fine-grained strech-1 PRGs replacing regular stretch-1 PRGs) as long as the distribution defined above is a distribution over circuits contained in $\mathcal{F}$, whenever D is contained in $\mathcal{F}$. This holds when instantiating $\mathcal{F}$ with the class NC^1 since, $\mathcal{C}' \in \mathsf{NC}^1$. Moreover, since NC^1-fine-grained commitments and strech-1 PRGs exist under the assumption that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ (see [DVV16]), we obtain the following theorem:

Theorem 14. *Assume $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ and assume that $\Pi = (\mathsf{Prover}^{ZAP}, \mathsf{Verifier}^{ZAP})$ is a NC^1-fine-grained ZAP system. Then the above construction is a NC^1-fine-grained oNIZK in the URS model.*

4 ZAPs, NIWI and oNIZK for AM or NP with Polynomial Security

Using Theorem 11 and the fact that non-adaptive (standard) NIZK with unbounded provers in the URS model for AM can be constructed from one-way permutations, we obtain:

Theorem 15 (Informal). *Assuming the existence of one-way permutations, there exist NIWI for AM with inefficient provers.*

Using known results [BOV03, BOV07] on transformation of ZAPs to NIWI proof systems via the derandomization assumption of the existence of hitting set generators (HSG) against co-nondeterministic uniform algorithms, we obtain:

Theorem 16 (Informal). *Assuming the existence of one-way permutations and efficient 1/2-HSG against co-nondeterministic uniform algorithms, there exist NIWI for NP with inefficient provers.*

Finally, we observe that the only non-polynomial-time computations performed by the NIWI prover are inversions of OWP, and that the hardness of the OWP is tunable to any 2^{n^ϵ} for any constant $0 < \epsilon < 1$, since the input length of the OWP is independent of all other parameters. Therefore, we can use known techniques to obtain oNIZK in the standard model that guarantees soundness against uniform adversaries. Specifically, the soundness against uniform adversaries comes from the hardness of finding a collision in a keyless collision resistant hash function. Assuming subexponential hardness of such collision resistant hash functions, we can tune the hardness of finding such a collision t obe any 2^{n^c}, for any constant c. We therefore obtain the following:

Theorem 17 (Informal). *Assuming the existence of one-way permutations, efficient 1/2-HSG against co-nondeterministic uniform algorithms, and subexponentially-hard uniform collision resistant hash functions, then for any constant $0 < \epsilon < 1$ and constant $c \geq 1$, there exist oNIZK in the standard model for NP with honest provers running in uniform time 2^{n^ϵ} and soundness against uniform adversaries running in time $2^{n^{\epsilon'}}$, where n is security parameter.*

5 Fine-Grained ZAPs, NIWI and oNIZK for NP

5.1 Background on Randomized Encodings of [IK00]

Our exposition in this subsection follows that of [App14].

Let $BP = (G, \phi, s, t)$ be a mod-2 BP of size ℓ, computing a Boolean function $f : \{0,1\}^n \to \{0,1\}$; that is, $f(x) = 1$ if and only if the number of paths from s to t in G_x equals 1 modulo 2. Fix some topological ordering of the vertices of G, where the source vertex s is labeled 1 and the terminal vertex t is labeled ℓ. Let $A(x)$ be the $\ell \times \ell$ adjacency matrix of G_x viewed as a formal matrix whose entries are degree-1 polynomials in the input variables x. Specifically, the (i, j) entry of $A(x)$ contains the value of $\phi(i, j)$ on x if (i, j) is an edge in G, and 0 otherwise. (Hence, $A(x)$ contains the constant 0 on and below the main diagonal, and degree-1 polynomials in the input variables above the main diagonal.) Define $L(x)$ as the submatrix of $A(x) - I$ obtained by deleting column s and row t (i.e., the first column and the last row). As before, each entry of $L(x)$ is a degree-1 polynomial in a single input variable x_i; moreover, $L(x)$ contains the constant $-1 = 1$ mod 2 in each entry of its second diagonal (the one below the main diagonal) and the constant 0 below this diagonal (see Figure 5.1).

Let $r^{(1)}$ and $r^{(2)}$ be vectors of $\mathbb{F}_2$ of length $\sum_{i=1}^{\ell-2} i = \binom{\ell-1}{2}$ and $\ell - 2$, respectively. Let $R_1(r^{(1)})$ be an $(\ell - 1) \times (\ell - 1)$ matrix with 1's on the main diagonal, 0's below it, and $r^{(1)}$'s elements in the remaining $\binom{\ell-1}{2}$ entries above the diagonal (a unique element of $r^{(1)}$ is assigned to each matrix entry). Let $R_2(r^{(2)}$ be an $(\ell - 1) \times (\ell - 1)$ matrix with 1's on the main diagonal, $r^{(2)}$'s elements in the rightmost column, and 0's in each of the remaining entries (see Figure 5.1).

$$\begin{pmatrix} 1 & r_1^{(1)} & r_2^{(1)} & \cdot & \cdot & r_{\ell-2}^{(1)} \\ 0 & 1 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1 & r_{\binom{\ell-1}{2}}^{(1)} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ 0 & 1 & * & * & * & * \\ 0 & 0 & 1 & * & * & * \\ 0 & 0 & 0 & 1 & * & * \\ 0 & 0 & 0 & 0 & 1 & * \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & r_1^{(2)} \\ 0 & 1 & 0 & 0 & 0 & r_2^{(2)} \\ 0 & 0 & 1 & 0 & 0 & \cdot \\ 0 & 0 & 0 & 1 & 0 & \cdot \\ 0 & 0 & 0 & 0 & 1 & r_{\ell-2}^{(2)} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 1. **

Fact 1 ([App14]) *Let M, M' be $(\ell-1)\times(\ell-1)$ matrices that contain the constant $-1 = 1 \mod 2$ in each entry of their second diagonal and the constant 0 below this diagonal. Then, $\det(M) = \det(M')$ if and only if there exist $r^{(1)}$ and $r^{(2)}$ such that $R_1(r^{(1)})MR_2(r^{(2)}) = M'$.*

Lemma 1 ([App14]). *Let BP be a mod-2 branching program computing the Boolean function f. Define a function $\hat{f}(x,(r^{(1)},r^{(2)})) := R_1(r^{(1)})L(x)R_2(r^{(2)})$. Then $\hat{f}$ is a perfect randomized encoding of f.*

Define

$$M_0 := \begin{pmatrix} 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \end{pmatrix} \quad M_1 := \begin{pmatrix} 0&0&0&0&0&1 \\ 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \end{pmatrix}$$

Lemma 2. *Assuming $\oplus L/\text{poly} \not\subseteq \mathsf{NC}^1$, the distributions $R_1(r^{(1)})M_0R_2(r^{(2)})$ and $R_1(r^{(1)})M_1R_2(r^{(2)})$ cannot be distinguished by NC^1 circuits, where $r^{(1)}, r^{(2)}$ are chosen at random.*

5.2 Statistical NIZK protocol in the URS model for $\oplus L/\text{poly}$

Due to properties of the randomized encoding construction of [IK00], we can construct a statistical NIZK protocol in the common *random* string model for languages in $\oplus L/\text{poly}$. Our protocol is heavily based on the protocol of Applebaum and Raykov [AR16], which gave a NISZK construction in the common *reference* string model for languages that have (statistical) randomized encodings. Our protocol is described next:

- **URS Generation:** The URS consists of λ random strings, each from $\{0,1\}^t = \{0,1\}^{\binom{\ell-1}{2}+\ell-1}$.
- **Prover:** On input statement matrix M, the prover does the following:
  1. For $i \in [\lambda]$, use the i-th block of t bits to populate the upper-triangular entries of a matrix M_i' that has -1's on its second diagonal and 0's below it.
  2. For $i \in [\lambda]$, if $\det(M_i') = 0$, reveal $r_i^{(1)}, r_i^{(2)}$ of the correct form such that $R_1(r_i^{(1)})M_0R_2(r_i^{(2)}) = M_i'$, where M_0 is a determinant 0 matrix of "canonical form."
  3. Otherwise, reveal $r^{(1)}, r^{(2)}$ of the correct form, such that $R_1(r_i^{(1)})MR_2(r_i^{(2)}) = M_i'$.
  4. Output $\pi = [(r_i^{(1)}, r_i^{(2)})]_{i\in[\lambda]}$.
- **Verifier:** On input $(\mathsf{URS}, M, \pi = [(r_i^{(1)}, r_i^{(2)})]_{i\in[\lambda]})$, the verifier checks that for all $i \in [\lambda]$, either $M_i' = R_1(r_i^{(1)})M_0R_2(r_i^{(2)})$ or $M_i' = R_1(r_i^{(1)})MR_2(r_i^{(2)})$.

Lemma 3. *The protocol above is a NIZK proof system with statistical soundness and statistical zero knowledge in the common* random *string model for languages $\mathcal{L} \in \oplus P/\text{poly}$. Moreover, the NIZK simulator can be instantiated by sampling a NC^1 circuit Sim from an efficiently samplable distribution $\mathcal{D}_{\mathsf{Sim}}$.*

The only way that soundness fails, is if all λ instances M'_i obtained from the URS have determinant 0. This occurs with probability at most $2^{-\lambda}$, due to the fact that the randomized encodings of [IK00] are "balanced" (i.e. an equal number of strings correspond to an encoding of 0 vs. 1).

Statistical Zero knowledge. We define the following randomized circuit $\mathsf{Sim} \in \mathsf{NC}^1$. Sim takes as input the instance, represented by M, and a sufficiently long string of random coins and does as follows:

- Sample a random set $\mathcal{S} \subseteq [\lambda]$ of indeces, where each $i \in [\lambda]$ is placed in $\mathcal{S}$ with independent probability $1/2$.
- Sample random bitsrings $(r_i^{(1)}, r_i^{(2)})$ of the correct length for $i \in [\lambda]$.
- For $i \in \mathcal{S}$, Sim sets the corresponding t bit string of the URS to consist of the entries of $R_1(r_i^{(1)})M_0R_2(r_i^{(2)})$ and reveals $(r_i^{(1)}, r_i^{(2)})$.
- For $i \notin \mathcal{S}$, Sim sets the corresponding t bit string of the URS to consist of the entries of $R_1(r_i^{(1)})MR_2(r_i^{(2)})$ and reveals $(r_i^{(1)}, r_i^{(2)})$.

5.3 $\mathcal{G}$-extractable, $\mathcal{F}$-Fine-Grained Commitments for NC^1

$\mathcal{G}$-extractable, $\mathcal{F}$-Fine-Grained Commitments are are commitments that are perfectly binding and have the following properties (see also Definition 3 for a formal description):

- The commitments can be computed and opened in class $\mathcal{F}$.
- Given a commitment, the committed value can be extracted in class $\mathcal{G}$.
- The hiding property of the commitment holds against adversaries in class $\mathcal{F}$.

For our purposes, we will consider $\mathcal{G}$ to be the class $\oplus L/\text{ poly}$ and the class $\mathcal{F}$ to be the class NC^1.

Define the following languages $\mathcal{L}_{det}$, $\overline{\mathcal{L}_{det}}$. $\mathcal{L}_{det}$ is the set of $\ell - 1 \times \ell - 1$ matrices M with -1 on the second diagonal, 0's below the second diagonal, 0 or 1 elements on the diagonal and above such that M has determinant 1 over $\mathbb{F}_2$. $\overline{\mathcal{L}_{det}}$ is the set of $\ell - 1 \times \ell - 1$ matrices M with -1 on the second diagonal, 0's below the second diagonal, 0 or 1 elements on the diagonal and above such that M has determinant 0 over $\mathbb{F}_2$.

Claim. The languages $\mathcal{L}_{det}$ and $\overline{\mathcal{L}_{det}}$ are contained in $\oplus L/\text{ poly}$.

Toda [Tod84] showed that the determinant is complete for $\#\mathsf{L}$ by demonstrating NC^1-computable projection from the determinant to counting paths in acyclic graphs. It is not difficult to see that $\oplus\mathsf{L}/poly$.

Construction of $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme: To commit to a 1, choose random $(r^{(1)}, r^{(2)})$ of appropriate length and output $R_1(r^{(1)})M_0R_2(r^{(2)})$. To commit to a 0, choose random $(r^{(1)}, r^{(2)})$ of appropriate length and output $R_1(r^{(1)})M_1R_2(r^{(2)})$.

The required properties of the $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme follow from Claim 5.3 and from the assumption that $\oplus L/\text{ poly} \not\subseteq \mathsf{NC}^1$, as shown by [DVV16].

5.4 NC^1-Fine-Grained NIZK for Circuit SAT

Assume $\mathcal{C}$ is represented as a circuit consisting of NAND gates and assume it has z number of wires. The value of each wire is committed (using the $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme from the previous section) as $com_1, \ldots, com_z$. Recall that com_i commits to 1 iff $com_1 \in \mathcal{L}_{det}$ and com_1 commits to 0 iff $com_1 \notin \mathcal{L}_{det}$. Additionally, recall that $\mathcal{L}_{det}$ (and therefore also $\overline{\mathcal{L}_{det}}$) is contained in $\oplus L/\text{ poly}$. The language $\mathcal{L}_{\mathcal{C}}$ consists of strings $com_1, \ldots, com_z$ which satisfy all of the following:

- $com_z \in \mathcal{L}_{det}$
- For each gate G_ℓ with with input wires i, j and output wire k:

$$\left(com_i \in \overline{\mathcal{L}_{det}} \wedge com_k \in \mathcal{L}_{det}\right) \vee \left(com_j \in \overline{\mathcal{L}_{det}} \wedge com_k \in \mathcal{L}_{det}\right) \vee \left(com_i \in \mathcal{L}_{det} \wedge com_j \in \mathcal{L}_{det} \wedge com_k \in \overline{\mathcal{L}_{det}}\right).$$

We denote this as $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$.

Due to closure of $\oplus L/$ poly w.r.t. negation, conjunction and disjunction [BG99], we have that $\mathcal{L}_{\mathcal{C}} \in \oplus L/\text{ poly}$.

Construction of NC^1*-Fine-Grained NIZK for Circuit SAT.* Given a circuit-SAT instance with circuit C, commit to the witness w using the above type of commitment (i.e. the witness corresponds to the values of all wires in the circuit C and the commitment is a wire-by-wire commitment to those values as above). We have shown above that the following language $\mathcal{L}_{\mathcal{C}}$ is then in $\oplus L/\operatorname{poly}$ $\mathcal{L}_{\mathcal{C}}$: $\{(com_1, \ldots, com_z)$: $com_1, \ldots, com_z$ are commitments to $w = w_1, \ldots, w_z$ and w is a circuit-SAT witness for $C\}$.

Now, applying the argument system from before to proving statement $(com_1, \ldots, com_z)$ is contained in language $\mathcal{L}_{\mathcal{C}}$ yields a fine-grained NIZK in the URS model for circuit SAT.

In more detail, the construction proceeds as follows: The Prover commits to witness $w = w_1, \ldots, w_z$ using a $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme, yielding $(com_1, \ldots, com_z)$. The Prover then runs the statistical NIZK protocol given above in Section 5.2 to prove that $(com_1, \ldots, com_z) \in \mathcal{L}_{\mathcal{C}}$.

Theorem 18. *The construction above is a* NC^1*-fine-grained NIZK proof system for the circuit SAT language.*

Note that the above implies a NC^1-fine-grained NIZK proof system for all of NP. This is because the prover can perform the Karp reduction, transforming the statement to be proved to a Circuit SAT statement. The prover can then include the tableau of the computation performed to transform the instance in the proof, and the NC^1 verifier can first verify that the Karp reduction was computed correctly, and then verify the underlying proof.

To argue zero knowledge of the NIZK against a NC^1 distinguisher, we define the following randomized circuit $\mathsf{Sim}' \in \mathsf{NC}^1$. Sim' takes as input the instance, represented by NAND circuit $\mathcal{C}$ consisting of z number of wires, and a sufficiently long string of random coins and does as follows:

- Generate z commitments to garbage $(com_1, \ldots, com_z)$.
- Let Sim be the zero knowledge simulator defined in Section 5.2 for languages in $\oplus L/\operatorname{poly}$.
- Sim' runs the simulator Sim on input statement $(com_1, \ldots, com_z)$ and language $\mathcal{L}_{\mathcal{C}}$.
- Sim' outputs whatever Sim outputs.

Note that $\mathsf{Sim}' \in \mathsf{NC}^1$, since $\mathsf{Sim} \in \mathsf{NC}^1$. If a NC^1 adversary can distinguish simulated and real proofs, then we can use the adversary to break the hiding property of the $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme, a contradiction.

Alternative Construction. The following alternative construction of a NC^1-fine-grained NIZK proof system for the Circuit SAT language will be useful for some of our applications. In the alternate construction, the Prover still produces a single set of commitments $(com_1, \ldots, com_z)$ to the wire values of the circuit $\mathcal{C}$, but we include a separate URS for proving that the satisfying assignment is valid for each individual gate. Details follow.

Given a circuit $\mathcal{C}$ with t gates and a common random string split into $t+1$ sections $\mathsf{URS}_0, \ldots, \mathsf{URS}_t$, witness $w = w_1, \ldots, w_z$ (which consists of the values of each of the z wires of $\mathcal{C}$ corresponding to a satisfying assignment. The Prover does as follows: Commit to the values of the wires, using the $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme, producing commitments $com_1, \ldots, com_z$. Use URS_0 to prove that $com_z \in \mathcal{L}_{det}$. For the ℓ-th gate, with input wires i, j and output wire k, use URS_ℓ to prove that $(com_i, com_j, com_k) \in \mathcal{L}_{gate}$. Note that $\mathcal{L}_{det}$ and $\mathcal{L}_{gate}$ are both contained in $\oplus L/$ poly. Therefore, to prove each of the $z+1$ statements, the NIZK in the URS model given in Section 5.2 can be used.

The simulator Sim' is almost exactly as above: On input circuit $\mathcal{C}$, Sim' still proceeds by randomly generating a commitments $(com_1, \ldots, com_z)$ to garbage. We also now will now invoke the simulator $\mathsf{Sim} \in \mathsf{NC}^1$ given in Section 5.2, Sim' now has $z+1$ number of statements to be proven about different subsets of the commitments $(com_1, \ldots, com_z)$, each proof with a separate URS. In parallel for the $z+1$ statements, Sim' invokes Sim with the corresponding i-th statement. Each invocation of Sim outputs a URS_i and a proof π_i. Sim' then concatenates the URS's and proofs and outputs the result as the final simulated URS and proof.

Since $\mathsf{Sim}' \in \mathsf{NC}^1$, if a NC^1 adversary can distinguish simulated and real proofs, then we can use the adversary to break the hiding property of the $\oplus L/$ poly-extractable, NC^1-Fine-Grained Commitment Scheme, a contradiction.

5.5 NC^1-Fine-Grained ZAPs for Circuit SAT

We use either the first or alternate construction above together with Theorem 12 to obtain the following:

Theorem 19. *Assuming that $\oplus P/\text{ poly} \not\subseteq \mathsf{NC}^1$, there exist NC^1-fine-grained ZAPs for NP.*

5.6 NC^1-Fine-Grained NIWI for NP

We use the transformation of Barak et al. [BOV03, BOV07] from ZAPs to NIWI, that relies on the existence of hitting set generators (HSG) against co-nondeterministic uniform algorithms. Note that this transformation retains statistical soundness (due to the properties of the HSG) and retains its witness indistinguishability against NC^1 adversaries. However, the verifier may no longer be in NC^1, since the verifier must evaluate the HSG in order to check that the prover is using the correct URS for each of the sub-proofs. To remedy this situation, the prover evaluates the HSG and then sends a tableau of the computation (which can be verified in AC^0) to the verifier, who can then verify that the URS being used is indeed consistent with the output of the HSG.

Theorem 20. *Assuming that $\oplus P/\text{ poly} \not\subseteq \mathsf{NC}^1$, the existence of efficient 1/2-HSG against co-nondeterministic uniform algorithms, there exist NC^1-fine-grained NIWI for NP.*

5.7 NC^1-Fine-Grained oNIZK with uniform soundness

We now assume existence of a uniform collision resistant hash function h. Let $\mathcal{C}_h$ be the circuit that takes two inputs x_1, x_2 and outputs 1 if $x_1 \neq x_2$ and $h(x_1) = h(x_2)$. On input circuit SAT circuit $\mathcal{C}$, the prover now proves circuit satisfiability of the circuit $\mathcal{C}'$, where $\mathcal{C}'$ is defined as follows: $\mathcal{C}'$ takes public input $\mathsf{desc}(\mathcal{C})$, which is a description of the circuit $\mathcal{C}$, and private input x. $\mathcal{C}'$ outputs 1 on input $(\mathsf{desc}(\mathcal{C}), x)$ if and only if x is a satisfying assignment for $\mathcal{C}$ or x is a satisfying assignment for $\mathcal{C}_h$. Note that $\mathcal{C}'$ is a NC^1 circuit.

On input statement $\mathcal{C}$, the Prover uses the NIWI based on the alternate construction of the NC^1-fine-grained NIZK proof system with statistical soundness for the Circuit SAT language to prove that (1) $(com_1, \ldots, \mathsf{com}_z)$ is a satisfying assignment for $\mathcal{C}'$ and (2) The commitments corresponding to the public input decommit to values that are consistent with $\mathsf{desc}(\mathcal{C})$.

The verifier runs the verifier of the NIWI to verify the proof for the statements (1) and (2) above.

To prove zero knowledge with offline simulation (oNIZK), we must show a distribution $\mathcal{D}_{\mathsf{Sim}}$ over NC^1 circuits such that a circuit drawn from this distribution, evaluated on input statement $\mathcal{C}$ produces a distribution over proofs that is indistinguishable from real proofs for a NC^1 circuit.

A draw from $\mathcal{D}_{\mathsf{Sim}}$ is defined as follows:

- Sample colliding inputs x_1, x_2 for h.
- For each wire i of $\mathcal{C}'$, sample a commitment to 0 and a commitment to 1: (com_i^0, com_i^1).
- For each public wire i of $\mathcal{C}'$, compute honest proofs $\pi_{in,i}^0, \pi_{in,i}^1$ proving that $com_i^0 \in \overline{\mathcal{L}_{det}}$ and that $com_i^1 \in \mathcal{L}_{det}$, respectively.
- For the output wire z of $\mathcal{C}'$, compute an honest proof π_{out} that $com_z^1 \in \mathcal{L}_{det}$.
- For each gate with input wires i, j and output wire k of $\mathcal{C}'$, compute 4 honest proofs $[\pi_{gate,i,j,k}^{b_1,b_2}]_{b_1,b_2 \in \{0,1\}}$ proving that $com_i^{b_1}, com_j^{b_2}, com_k^{1-b_1 \wedge b_2} \in \mathcal{L}_{gate}$, for $b_1, b_2 \in \{0,1\}$.
- **Hardwired Values:** A satisfying assignment y (using colliding inputs x_1, x_2) for $\mathcal{C}_h$ and $[com_i^0, com_i^1]_{i \in [z]}$, $(\pi_{in,i}^0, \pi_{in,i}^1)$, π_{out}, $[\pi_{gate,i,j,k}^{b_1,b_2}]_{i,j,k,b_1,b_2}$.
- **Circuit Evaluation:** On input $\mathsf{desc}(\mathcal{C})$, choose the appropriate public inputs corresponding to that input. Additionally, chose the private inputs corresponding to the satisfying assignment y. Let $b_{in}(i)$ denote the value of the i-th public input wire. Assume there are a total of z' input wires. Using these, compute the values of all wires of $\mathcal{C}'$ (this can be done in NC^1, since $\mathcal{C}'$ is a NC^1 circuit). Let $b(i)$ denote the value of the i-th wire of $\mathcal{C}'$. Output commitments $[com_i^{b(j)}]_{i \in [z]}$ and proofs $[\pi_{in,i}^{b_{in}(i)}]_{i \in [z']}$, $[\pi_{gate,i,j,k}^{b(i),b(j)}]_{i,j,k}$.

Note that the outputted distribution is indentical to an honest proof with witness corresponding to a satisfying assignment of $\mathcal{C}_h$. Thus, by the witness indistinguishability property of the proof system, the simulated proof is indistinguishable from the real proof.

Moreover, note that by the collision resistance of h, soundness still holds against uniform, poly-time provers.

References

ADKL19. Prabhanjan Ananth, Apoorvaa Deshpande, Yael Tauman Kalai, and Anna Lysyanskaya. Fully homomorphic nizk and niwi proofs. Cryptology ePrint Archive, Report 2019/732, 2019. https://eprint.iacr.org/2019/732.

App14. Benny Applebaum. *Cryptography in Constant Parallel Time*. Information Security and Cryptography. Springer, 2014.

AR16. Benny Applebaum and Pavel Raykov. On the relationship between statistical zero-knowledge and statistical randomized encodings. In Robshaw and Katz [RK16], pages 449–477.

BFJ+19. Saikrishna Badrinarayan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical zap arguments. Cryptology ePrint Archive, Report 2019/780, 2019. https://eprint.iacr.org/2019/780.

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.

BG99. Amos Beimel and Anna Gál. On arithmetic branching programs. *J. Comput. Syst. Sci.*, 59(2):195–220, 1999.

BG03. Michael Ben-Or and Danny Gutfreund. Trading help for interaction in statistical zero-knowledge proofs. *Journal of Cryptology*, 16(2):95–116, March 2003.

BL18. Nir Bitansky and Huijia Lin. One-message zero knowledge and non-malleable commitments. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 209–234. Springer, Heidelberg, November 2018.

BMO90. Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *22nd ACM STOC*, pages 482–493. ACM Press, May 1990.

BOV03. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 299–315. Springer, Heidelberg, August 2003.

BOV07. Boaz Barak, Shien Jin Ong, and Salil Vadhan. Derandomization in cryptography. *SIAM Journal on Computing*, 37(2):380–400, 2007.

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.

BRSV17. Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 483–496. ACM Press, June 2017.

BRSV18. Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of work from worst-case assumptions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 789–819. Springer, Heidelberg, August 2018.

BY96. Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *Journal of Cryptology*, 9(3):149–166, June 1996.

Can08. Ran Canetti, editor. *TCC 2008*, volume 4948 of *LNCS*. Springer, Heidelberg, March 2008.

CCH+19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019.

CCKV08. André Chailloux, Dragos Florin Ciocan, Iordanis Kerenidis, and Salil P. Vadhan. Interactive and noninteractive zero knowledge are equivalent in the help model. In Canetti [Can08], pages 501–534.

CG18. Matteo Campanelli and Rosario Gennaro. Fine-grained secure computation. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 66–97. Springer, Heidelberg, November 2018.

CV07. Dragos Florin Ciocan and Salil Vadhan. Interactive and noninteractive zero knowledge coincide in the help model. Cryptology ePrint Archive, Report 2007/389, 2007. http://eprint.iacr.org/2007/389.

DG11. Ning Ding and Dawu Gu. Precise time and space simulatable zero-knowledge. In *ProvSec*, volume 6980 of *Lecture Notes in Computer Science*, pages 16–33. Springer, 2011.

DG12. Ning Ding and Dawu Gu. On constant-round precise zero-knowledge. In *ICICS*, volume 7618 of *Lecture Notes in Computer Science*, pages 178–190. Springer, 2012.

DMP90. Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge with preprocessing. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 269–282. Springer, Heidelberg, August 1990.

DN07. Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM Journal on Computing*, 36(6):1513–1543, 2007.

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.

DVV16. Akshay Degwekar, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Fine-grained cryptography. In Robshaw and Katz [RK16], pages 533–562.

FLS99. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

FS90. Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 526–544. Springer, Heidelberg, August 1990.

GJS19. Vipul Goyal, Aayush Jain, and Amit Sahai. Simultaneous amplification: The case of non-interactive zero-knowledge. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 608–637. Springer, Heidelberg, August 2019.

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

GO94. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

Gol01. Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001.

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.

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM (JACM)*, 59(3):11, 2012.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

IK00. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000.

Imp95. Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995.

IOS94. Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. Language dependent secure bit commitment. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 188–201. Springer, Heidelberg, August 1994.

IR89. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

JJ19. Abhishek Jain and Zhengzhong Jin. Statistical zap arguments from quasi-polynomial lwe. Cryptology ePrint Archive, Report 2019/839, 2019. https://eprint.iacr.org/2019/839.

JKKR17. Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Heidelberg, August 2017.

KKS18. Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 34–65. Springer, Heidelberg, April / May 2018.

KNYY19. Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing NIZKs from diffie-hellman assumptions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 622–651. Springer, Heidelberg, May 2019.

KW18. Sam Kim and David J. Wu. Multi-theorem preprocessing NIZKs from lattices. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 733–765. Springer, Heidelberg, August 2018.

LLW19. Rio LaVigne, Andrea Lincoln, and Virginia Vassilevska Williams. Public-key cryptography in the fine-grained setting. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 605–635. Springer, Heidelberg, August 2019.

LS91. Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 353–365. Springer, Heidelberg, August 1991.

LY94. Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *STOC*, pages 734–740. ACM, 1994.

MP06. Silvio Micali and Rafael Pass. Local zero knowledge. In *STOC*, pages 306–315. ACM, 2006.

NW88. Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *29th FOCS*, pages 2–11. IEEE Computer Society Press, October 1988.

OS17. Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Computational Complexity Conference*, volume 79 of *LIPIcs*, pages 18:1–18:49. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

OV08. Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In Canetti [Can08], pages 482–500.

OW93. Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*, pages 3–17, 1993.

Ps05. Rafael Pass and Abhi shelat. Unconditional characterizations of non-interactive zero-knowledge. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 118–134. Springer, Heidelberg, August 2005.

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.

RK16. Matthew Robshaw and Jonathan Katz, editors. *CRYPTO 2016, Part III*, volume 9816 of *LNCS*. Springer, Heidelberg, August 2016.

Tod84. Seinosuke Toda. Counting problems computationally equivalent to. *SIAM J. Computing*, 13:423–439, 1984.

WW14. Huixin Wu and Feng Wang. A survey of noninteractive zero knowledge proof system and its applications, 2014.