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

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] University of Massachusetts, Amherst
mukul@cs.umass.edu

Abstract. We present a transformation from NIZK with *inefficient provers* in the uniform random string (URS) model to ZAPs (two message witness indistinguishable proofs) with *inefficient provers*. While such a transformation was known for the case where the prover is efficient, the security proof breaks down if the prover is inefficient. Our transformation is obtained via new applications of Nisan-Wigderson *designs*, a combinatorial object originally introduced in the derandomization literature.

We observe that our transformation is applicable both in the setting of super-polynomial provers/poly-time adversaries, as well as a new fine-grained setting, where the prover is polynomial time and the verifier/simulator/zero knowledge distinguisher are in a lower complexity class, such as NC^1. We also present NC^1-fine-grained NIZK in the URS model for all of NP from the worst-case assumption $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$.

Our techniques yield the following applications:

1. ZAPs for AM from Minicrypt assumptions (with super-polynomial time provers),
2. NC^1-fine-grained ZAPs for NP from worst-case assumptions,
3. Protocols achieving an "offline" notion of NIZK (oNIZK) in the standard (no-CRS) model with uniform soundness in both the super-polynomial setting (from Minicrypt assumptions) and the NC^1-fine-grained setting (from worst-case assumptions). The oNIZK notion is sufficient for use in indistinguishability-based proofs.

1 Introduction

A long and important line of research has been dedicated to understanding the necessary and sufficient assumptions for the existence of computational zero

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

knowledge (CZK) proofs (with potentially unbounded provers) for a language $\mathcal{L}$ [13, 60, 49]. This line of research culminated with the work of Ong and Vadhan [59] which fully resolved the question by proving that a language in NP has a CZK 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] (sometimes called the "public parameters" setting)—are also well-understood. Pass and shelat [61], 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.

While the NIZK of Pass and shelat [61] indeed minimizes interaction and assumptions, it critically utilizes trusted setup to generate a structured CRS sampled from a particular distribution. In contrast, motivated by concerns of subversion of public parameters [12] and considerations from the blockchain community [16, 17, 18], a recent line of research has focused on "transparent" setup that does not require a trusted party, but simply access to a shared source of public randomness such the NIST randomness beacon, or a uniform random string (URS).[5] In the URS model, it is well known that NIZK with unbounded provers follows from one-way permutations (OWP) [34]. However, even agreeing upon a genuinely random string to implement the URS model may be infeasible in some cases.

We investigate what can be proven with "zero-knowledge" in a truly trust-free setting, with *minimal interaction and assumptions*. In particular, we extend the above line of work on minimizing assumptions to other types of "zero knowledge" primitives, such as *ZAPs* (two message witness indistinguishable (WI) proofs), non-interactive witness indistinguishable proofs (NIWI), and, ultimately, a type of NIZK with uniform soundness (and no URS/CRS).

Our primary goal is to understand the relationship between ZAPs and zero-knowledge primitives that can be constructed from minimal assumptions in the *inefficient prover* setting. Once we construct ZAPs, we will show that NIWI and a type of NIZK with uniform soundness can also be constructed (note that while these implications are already known in the efficient-prover setting [8, 10], hurdles are introduced by removing this constraint). Ultimately, we are interested in obtaining constructions of ZAPs from *Minicrypt* [45] assumptions only[6]. To further motivate our focus on the *inefficient prover* setting, note that

[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.

[5] Note that recent work on transparent or trustless (succinct) proofs, typically assumes existence of a public random oracle. We will will only consider (at most) *short* public random strings in this work.

[6] 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

barriers are known for constructions of ZAPs from Minicrypt assumptions when the prover is required to be efficient. Indeed, efficient-prover ZAPs are known to be equivalent to efficient-prover NIZK in the URS model [32] (assuming one-way functions exist), and efficient-prover NIZKs, in turn, are only known to be achievable from Cryptomania [45] primitives such as (enhanced) trapdoor permutations. (See Section 1.2 for details.)

Because of this dichotomy, 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 setting of super-polynomial provers/polynomial adversary, 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). Our main technical contribution is a new transformation from inefficient prover NIZK in the URS model to inefficient prover ZAPs. A single transformation works both for the unbounded prover and fine-grained settings. Our transformation is obtained via new applications of Nisan-Wigderson *designs*, a combinatorial object originally introduced in the derandomization literature [58]. We also show that fine-grained NIZK in the URS model is achievable from worst-case assumptions ($\oplus L/\text{poly} \not\subseteq \mathsf{NC}^1$). Given the well-known construction of unbounded prover NIZK in the URS model from one-way permutations (via the hidden bits model), we obtain (1) super-poly prover ZAPs for AM from Minicrypt assumptions and (2) fine-grained ZAPs for NP from worst-case assumptions.

Technical hurdles introduced by inefficient provers. When dealing with inefficient provers, one must proceed with care, as many "folklore" results no longer hold. We make the following surprising observation (discussed in more detail in Section 1.1): While it is known that NIZKs in the uniform random string (URS) model imply ZAPs for the case of *efficient provers* [32], the transformation of [32] fails when the NIZK prover is inefficient. Briefly, this occurs because the *reduction* from the zero knowledge of the underlying NIZK to the witness indistinguishability of the ZAP does not have the computational power to run the *honest* prover's algorithm. Furthermore, as we will explain in Section 1.1, the honest proofs cannot simply be pre-computed and hardwired into the reduction. Instead, we must develop new techniques for the inefficient prover case.

Our notions of Zero Knowledge: the "fine-grained" setting. We introduce fine-grained analogues of zero knowledge and witness indistinguishability. 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) to what can be computed

original Impagliazzo and Rudich separation [46] For the same reason, we consider Collision-Resistant Hashing to be in Minicrypt.

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 such a notion of simulation 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 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 [29]) and a commitment-based ZK protocol (e.g. Blum-Hamiltonicity). We therefore focus on fine-grained ZAPs and NIZK.

NIZK imply ZAPs for inefficient provers. Our main contribution is to prove that NIZK in the URS model implies ZAPs, even in the case of *inefficient provers*. Specifically, we show the following:

Theorem 1 (Informal). *Assuming the existence of an NIZK proof system for a language $\mathcal{L} \in \mathsf{AM}$ with provers running in time T in the URS model, there exists a ZAP for $\mathcal{L}$ with provers running in time* $\mathrm{poly}(T, n)$*, where n is security parameter.*

Our proof surprisingly leverages a type of *design*—a combinatorial object that was used in the derandomization of BPP by Nisan and Wigderson [58]. To the best of our knowledge, this is a novel application of designs to the cryptographic setting.

We also briefly discuss here the notion of a "witness" for an AM language and the meaning of witness indistinguishability. Recall that a language is in AM iff it has an AM protocol ($\mathsf{Prover}, \mathsf{Verifier}$) and so AM languages are inherently tied to protocols. Therefore, similarly to tying witnesses for NP languages to a specific verification algorithm, the notion of a "witness" for an AM language will be tied to the protocol. Specifically, we assume that there is an AM-protocol for a language $\mathcal{L}$. Given the first message r from the verifier, we can consider the Circuit-SAT problem w.r.t. the first message r and the verifier's circuit.

[7] Note that this is very different from other fine-grained flavors of zero knowledge such as "knowledge tightness" or "precise zero knowledge" [37, 36, 57, 30, 31] 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).

Specifically, a witness w is a Prover's message that causes the verifier to output 1, when the first message r is fixed. Thus, witness-indistinguishability means that if there are two possible Prover messages w_1, w_2 that can be sent in response to r and such that the verifier accepts both, then the transcript of the ZAP should be indistinguishable when the Prover uses witness w_1 or w_2.

As a concrete example, consider the Goldwasser-Sipser (GS) protocol [40] for proving lower bounds on the size of NP sets. The verifier sends a random hash value and the prover responds with an element in the set that hashes to that value. WI is meaningful if there are multiple elements in the set that hash to the target value, since it guarantees that the verifier cannot distinguish which pre-image was used.[8]

Since it is well-known that NIZK with inefficient provers in the URS model can be constructed from one-way permutations (OWP) (see e.g. [61]), our result immediately yields ZAPs with subexponential provers from the Minicrypt assumption of OWP.

Theorem 2 (Informal). *Assuming the existence of one-way permutations, if* $\mathcal{L} \in \mathsf{AM}$ *with prover run-time* T*, then there exists a ZAP for* $\mathcal{L}$ *with prover run-time* $\mathrm{poly}(T, \mathsf{subexp}(n))$.

Extending to the fine-grained setting. Next, we observe that our same transformation can be applied to obtain *fine-grained* ZAPs from fine-grained NIZK in the URS model. 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 $\mathcal{F}$. 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. We thus obtain the following:

Theorem 3 (Informal). *Assuming the existence of non-adaptive* NC^1*-fine-grained NIZK proof systems for* NP *in the URS model, there exist* NC^1*-fine-grained ZAPs for* NP.

We next show how to construct NC^1-fine-grained NIZK in the URS model for all of NP, assuming the *worst-case assumption* that $\oplus L/\mathrm{poly} \not\subseteq \mathsf{NC}^1$. Our result begins by converting the NIZK construction of [3] that works for languages $\mathcal{L}$ with randomized encodings from the CRS model to the URS model. [9] Since randomized encodings are known for the class $\oplus L/\mathrm{poly}$, this

[8] We note that the GS protocol is used to prove that MA is contained in AM (by proving that the set of accepting coins of the verifier is sufficiently large). Recall that MA is like NP except the verifier can be randomized. It is not difficult to observe that our notion under the above transformation yields proofs for MA where witnesses that make the randomized verifier accept w.h.p. are indistinguishable.

[9] Recently, [33] constructed one-way permutations in the fine-grained setting. However, their results cannot be extended in straight-forward manner to construct fine-grained NIZKs and therefore are unlikely to lead to simpler constructions

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 [29]. 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 4 (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. [8] 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.

The constructions of NIWI from ZAPs and derandomization techniques go through in the inefficient-prover setting, since parallel repetition of 2-message protocols retains WI even in the inefficient prover setting (though this is not necessarily true for protocols with more than 2-messages).

We are not able to convert NIWI into fully standard NIZK with uniform soundness. The reason is that the transformation from NIWI to NIZK with uniform soundness in the no-CRS model employs the well-known FLS paradigm [34]. In this paradigm, the ZK simulator runs the honest prover with a trapdoor witness. However, in our case, the simulator cannot run the honest prover as it does not have enough computational power. Fortunately, we are able to show that if the simulator is given non-uniform advice that *does not depend on the statement being proved* then the simulator can perfectly simulate the honest prover's output on the trapdoor witness. Thus, we introduce offline NIZK (oNIZK), which 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). We note that this notion is sufficient for indistinguishability-based applications. We next state our results for the oNIZK setting:

without using other techniques. For more discussion on this, we refer the interested readers to Section 1.2.

Theorem 5 (Informal). *Assuming the existence of one-way permutations, appropriate derandomization assumptions,*[10] *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.*

Theorem 6 (Informal). *Assuming that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$, appropriate derandomization assumptions as above, and the existence of uniform collision resistant hash functions, there exist NC^1-fine-grained 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

[10] Specifically, the existence of efficient 1/2-hitting set generators (HSG) against co-nondeterministic uniform algorithms [8].

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 to prove soundness of 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 arguing witness indistinguishability. The reason can be seen from the sketch of the hybrid argument above, where we 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 $i-1$ proofs and run the honest prover with witness w_1 for the last $n'-i$ 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 and therefore 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. Specifically, our construction leverages the notion of a *design*, introduced by Nisan and Wigderson in their seminal work [58]. A *design* with parameters (l, n, c, n') is a set of n' 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. [58]). 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 ($i' \in [n'], i' \neq i$) is 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 $T_{i'}$ is 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 $T_{i'}$ is 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.

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 [3] 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 [47] with probability 1/2 (which follows from the fact that randomized encodings of [47] 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 $\oplus L/\operatorname{poly}$).

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 $\oplus L/\operatorname{poly} \neq \mathsf{NC}^1$, 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 $\oplus L/\operatorname{poly}$ (i.e. the decision problem $\mathcal{L}_{det}$ which, given a commitment com outputs 1 if it is a commitment to 1 is in $\oplus L/\operatorname{poly}$). (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 $\oplus L/\operatorname{poly}$ is closed under negation, disjunction and conjunction (see [11]), we can use the statistical-zero knowledge NIZK in the URS model for languages in $\oplus L/\operatorname{poly}$ 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 $\oplus L/\operatorname{poly}$ and since $\oplus L/\operatorname{poly}$ is closed under negation/conjunction/disjunction, we have that $\mathcal{L}_{\mathcal{C}} \in \oplus L/\operatorname{poly}$. Moreover, given $com_1, \ldots, com_z$, we can simulate a proof in NC^1 (using the simulator for the NIZK for languages in $\oplus L/\operatorname{poly}$), indicating that the NIZK provides zero knowledge against NC^1.

1.2 Related Work

Zero Knowledge Zero knowledge (ZK) proofs were introduced by Goldwasser, Micali, and Rackoff [39]. 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 – [37] which showed ZK proofs exist for all languages in NP, and [38] which showed that interaction is crucial for achieving zero knowledge property in case of non-trivial languages. Specifically, [38] 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) [21] 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 [32] 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. [21], 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 [34], whereas Lapidot and Shamir [55], constructed publicly verifiable NIZK from one-way permutations in URS model, Groth et al. [42] constructed NIZK from DLIN assumption in URS model. Recently, Peikert and Shiehian [62] constructed NIZK from LWE assumption in URS model.

NIZKs have also been studied in other models [15, 27, 26], and models which consider preprocessing along with other assumptions such as one-way encryption schemes exist [28], lattices (LWE) [54], and DDH/CDH [53]. Few of the other works on NIZKs include [14, 61, 44, 19, 25, 41, 1]. For more details on NIZK related research, we refer the interested readers to [65]

The notion of witness indistinguishable proofs was introduced by [35]. As discussed earlier, Dwork and Naor [32] 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. [8] gave a construction of NIWI based on derandomization assumptions and certified trapdoor permutations (by derandomizing the verifier of [32] construction). Groth et al. [43] constructed first non-interactive ZAP from DLIN assumption, whereas Bitansky and Paneth [20] 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 [51, 52], and quasi-polynomial hardness of LWE [4, 50].

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 [29] 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. [6, 7] 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 [23] initiated the study of fine-grained secure computation by constructing a verifiable computation protocol secure against NC^1 adversaries based on worst-case assumptions. LaVigne et al. [56] constructed a fine-grained key-exchange protocol.

Comparison with Egashira et al. [33] Recently, Egashira et al. [33] constructed one-way permutations, hash-proof systems, and trapdoor one-way functions, all of which can be computed in NC^1 and are secure against adversaries in NC^1, from the same assumptions that we consider in this work ($\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$). Their results do not directly extend to construct NC^1-fine-grained NIZK systems in the URS model, as (1) to the best of our knowledge it is not known how to construct NIZK in URS model from trapdoor one-way functions, and (2) their one-way permutation does not directly allow instantiation of the hidden bits model [34], which could then be used to construct NC^1-fine-grained NIZK in the URS model. Specifically, the domain/range of their OWP includes only full rank matrices and does not include *all* strings of a given length. Furthermore, whether a given string is contained in the domain/range cannot be determined by a NC^1 circuit (assuming $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$) and strings that are not in the range can have multiple pre-images. So to implement the hidden bits model, a prover would need to prove that a string is or is not contained in the domain/range, without compromising the one-wayness of unopened bits, which would itself require a NC^1-fine-grained NIZK proof system in the URS model. In contrast, our construction of NC^1-fine-grained NIZK in the URS model is direct and does not require fine-grained OWP nor implementing a fine-grained hidden bits model.

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 $\{\mathcal{D}_n^0\}_{n\in\mathbb{N}}, \{\mathcal{D}_n^0\}_{n\in\mathbb{N}}$, with support $\{0,1\}^{\ell(n)}$, are indistinguishable by $\mathcal{F}$ if*

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

We refer the interested reader to the full version of this paper [5], for additional definitions of fine-grained pseudorandom generator (PRG), as well as the standard definitions of witness indistinguishability (WI), and non-interactive witness indistinguishability (NIWI).

Definition 2 ($\mathcal{G}$-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,*

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

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 3 (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 4 (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* $|x|$*)*

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

A useful property of oNIZK is the following: Let $\mathcal{D}_{\mathsf{yes}}$ be a distribution over statements $x \in \mathcal{L}$ and let $\mathcal{D}_{\mathsf{no}}$ be a distribution over statements $x \in \overline{\mathcal{L}}$. If $\mathcal{D}_{\mathsf{yes}}$ and $\mathcal{D}_{\mathsf{no}}$ are computationally indistinguishable by polynomial-sized circuits then the following two distribution ensembles are computationally indistinguishable by polynomial-sized circuits (when the distinguishing gap is a function of $|x|$)

$$\{(x, (\mathsf{URS}, \pi) \leftarrow \mathsf{Sim}(x)) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}, x \leftarrow \mathcal{D}_{\mathsf{yes}}\}$$
$$\{(x', (\mathsf{URS}', \pi') \leftarrow \mathsf{Sim}(x')) : \mathsf{Sim} \leftarrow \mathcal{D}_{\mathsf{Sim}}, x' \leftarrow \mathcal{D}_{\mathsf{no}}\}.$$

The above allows a typical usage of oNIZK in hybrid style proofs: In the first hybrid, one can leave the statement the same and switch from proofs generated by the honest prover to proofs generated by the simulator, in the second step, one can switch the statement from a true statement to a false statement.

For more details on the relationship between Definition 4 and the notions of *witness hiding* (WH) and *weak zero knowledge* (WZK), see [5].

Definition 5 (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 6 (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 7 (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 $C' \in \mathsf{NC}^1$

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

2.2 Fine-Grained Witness Indistinguishability

Definition 8 ($\mathcal{F}$-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 9 (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, ρ, 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 10 ($\mathcal{F}$-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 Fine-Grained NIWI

Definition 11 ($\mathcal{F}$-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 12 (Non-Interactive Proofs with uniform soundness). *A pair of algorithms* (Prover, Verifier) *is called a non-interactive proof system with uniform soundness $T := T(|x|)$, for a language $\mathcal{L}$ if the algorithm* 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 13 (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 14 (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 15 (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 ZAPs from NIZK

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 [58].

Definition 16 (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 for $l := n^2$, constant $c \in \mathbb{N}$, and $n' := n^c$ [58].

Let $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ be a non-adaptive oNIZK in the URS model with inefficient prover for language $\mathcal{L}$ 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 $\mathsf{st} \in \mathcal{L}$, 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 indices 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 7. *Assume* $\Pi = (\mathsf{Prover}^{NIZK}, \mathsf{Verifier}^{NIZK})$ *is a non-adaptive oNIZK proof system for language* $\mathcal{L}$ *with an inefficient prover in the URS model. Then the above construction is a ZAP for language* $\mathcal{L}$ *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 distribution with statement st and witness w_1.

Hybrid H^{w_2}: This is the real 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:

Theorem 8. *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.*

We present the results related to ZAPs, NIWI and oNIZK for AM or NP with polynomial security in the full version of this paper [5].

4 Fine-Grained NIZK and ZAPs for NP

This section is focuses on constructing NC^1-fine-grained zero-knowledge non-interactive proofs for NP. Our general approach is to bootstrap a *statistical* NIZK for languages in $\oplus\mathsf{L}/\operatorname{poly}$ to a fine-grained NIZK for all of NP. The NISZK protocol we bootstrap is a variant of NISZK protocol from [3], in turn constructed from the randomized encodings of [47, 48], adapted to work in the URS setting. Next we repurpose the randomized encodings to construct a perfectly binding commitment scheme which is (a) hiding for NC^1, yet (b) extractable in $\oplus\mathsf{L}/\operatorname{poly}$. Finally, to prove a circuit is satisfiable, the prover simply commits to a witness and the ensuing circuit evaluation and appends a NISZK that the commitments indeed open to a satisfying evaluation (which, when using such a special commitment scheme, is a $\oplus\mathsf{L}/\operatorname{poly}$ statement). The fine-grained ZAP follows from the fine-grained NIZK by Theorem 8.

4.1 Background on Randomized Encodings of [47, 48]

We begin by reviewing some of the ingredients we require from the work of Ishai and Kushilevitz [47, 48]. Our exposition in this subsection follows that of [2].

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, where G_x is the subgraph of G specified momentarily. 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, $x_1, \ldots, x_n = x$. Specifically, the (i, j) entry of $A(x)$ contains the value of $\phi_{i,j}(x)$, where $\phi_{i,j}(x)$ is equal to either a constant function 1 or some literal, such as x_k or $\bar{x}_k$. We constrain ϕ such that if (i, j) is not an edge, the entry is necessarily 0. 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 4.1).

$$\begin{pmatrix} 1 & r_1^{(1)} & r_2^{(1)} & \cdot & \cdot & r_{\ell-2}^{(1)} \\ 0 & 1 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1 & r_{\binom{\ell-1}{2}}^{(1)} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_{1,2}(x) & \phi_{1,3}(x) & \cdot & \cdot & \cdot & \phi_{1,\ell}(x) \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 1 & \phi_{\ell-1,\ell}(x) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & r_1^{(2)} \\ 0 & 1 & 0 & 0 & 0 & r_2^{(2)} \\ 0 & 0 & 1 & 0 & 0 & \cdot \\ 0 & 0 & 0 & 1 & 0 & \cdot \\ 0 & 0 & 0 & 0 & 1 & r_{\ell-2}^{(2)} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 4.1. The matrices $R_1(r^{(1)})$, $A(x)$, and $R_2(r^{(2)})$.

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 4.1). We will also need the following facts. Note that in all that follows, we consider all arithmetic over $\mathbb{F}_2$, including determinants.

Fact 1 **([2])** *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 $R_1(r^{(1)}) M R_2(r^{(2)}) = M'$.*

Lemma 1 **([2]).** *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 and M_1 as matrices that are all 0 except for the lower diagonal which is 1, and the top right entry which is 1 (resp. 0) in M_1 (resp. M_0).

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

Lemma 2. *Assuming $\oplus L/\mathrm{poly} \not\subseteq \mathsf{NC}^1$, the distributions $R_1(r^{(1)}) M_0 R_2(r^{(2)})$ and $R_1(r^{(1)}) M_1 R_2(r^{(2)})$ cannot be distinguished by NC^1 circuits, where $r^{(1)}, r^{(2)}$ are chosen at random.*

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

Due to properties of the randomized encoding construction of [47], we can construct a statistical NIZK protocol in the uniform *random* string (URS) model for languages in $\oplus L/\operatorname{poly}$. Our protocol is heavily based on the protocol of Applebaum and Raykov [3], which gave a NISZK construction in the common *reference* string (CRS) 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 $M = L(x)$ (as defined in Section 4.1), 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.
  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." Otherwise, reveal $r^{(1)}, r^{(2)}$ of the correct form, such that $R_1(r_i^{(1)})MR_2(r_i^{(2)}) = M_i'$.
  3. 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 URS for languages $\mathcal{L} \in \oplus L/\operatorname{poly}$. Moreover, the NIZK simulator can be instantiated by sampling a NC^1 circuit Sim from an efficiently samplable distribution $\mathcal{D}_{\mathsf{Sim}}$.*

We present the proof of Lemma 3 in the full version of this paper [5].

4.3 $\mathcal{G}$-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 2):

- 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 $\mathcal{F}$.

For our purposes, we will consider $\mathcal{G}$ to be the class $\oplus L/\operatorname{poly}$ 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 $\mathbb{F}_2$.

Lemma 4. *The languages $\mathcal{L}_{det}$ and $\overline{\mathcal{L}_{det}}$ are contained in $\oplus L/\operatorname{poly}$.*

Toda [64] 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 follows that evaluating the determinant in $\mathbb{F}_2$ can be done in $\oplus\mathsf{L}/\mathit{poly}$.
Construction of $\oplus L/\operatorname{poly}$-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 $\oplus L/\operatorname{poly}$-extractable, NC^1-Fine-Grained Commitment Scheme follow from Lemma 4 and from the assumption that $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$, as shown by [29].

4.4 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 $\oplus L/\operatorname{poly}$-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 $com_1 \in \overline{\mathcal{L}_{det}}$. Additionally, recall that $\mathcal{L}_{det}$ (and therefore also $\overline{\mathcal{L}_{det}}$) is contained in $\oplus L/\operatorname{poly}$. 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:

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

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

Due to closure of $\oplus L/\operatorname{poly}$ w.r.t. negation, conjunction and disjunction [11], we have that $\mathcal{L}_{\mathcal{C}} \in \oplus L/\operatorname{poly}$.

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 $\oplus L/\operatorname{poly}$-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 4.2 to prove that $(com_1, \ldots, com_z) \in \mathcal{L}_{\mathcal{C}}$.

Theorem 9. *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 given an NP language, L, with a canonical polynomial size verification circuit $V(x, w)$, the prover can simply prove that the circuit $V_x(\cdot) := V(x, \cdot)$ is satisfiable. Because each bit of V_x is computable in NC^0, the NIZK verifier can generate V_x independently of the prover.

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 $(com_1, \dots, com_z)$.
- Let Sim be the zero knowledge simulator defined in Section 4.2 for languages in $\oplus L/\operatorname{poly}$.
- Sim' runs the simulator Sim on input statement $(com_1, \dots, com_z)$ 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 $\oplus L/\operatorname{poly}$-extractable, NC^1-Fine-Grained Commitment Scheme, a contradiction.

We require an alternative construction of NC^1-fine-grained NIZK in the URS model (deferred to the full version [5]), to construct NC^1-fine-grained oNIZK with uniform soundness in the standard model. We use either construction above together with Theorem 8 to obtain the following:

Theorem 10. *Assuming that* $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$*, there exist* NC^1*-fine-grained ZAPs for* NP*.*

4.5 NC^1-Fine-Grained NIWI for NP

We use the transformation of Barak et al. [9, 8] 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 11. *Assuming that* $\oplus L/\operatorname{poly} \not\subseteq \mathsf{NC}^1$*, the existence of efficient 1/2-HSG against co-nondeterministic uniform algorithms, there exist* NC^1*-fine-grained NIWI for* NP*.*

4.6 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) $(com_1, \dots, \mathsf{com}_z)$ 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.

Acknowledgments

We thank Tal Malkin for helpful discussions and suggestions to improve this work. The first author is supported in part 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 and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0085. 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, DAPRA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. The second and third authors are 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 and 70NANB19H126 from the U.S. Department of Commerce, National Institute of Standards and Technology.

References

1. Ananth, P., Deshpande, A., Kalai, Y.T., Lysyanskaya, A.: Fully homomorphic NIZK and NIWI proofs. Cryptology ePrint Archive, Report 2019/732 (2019), https://eprint.iacr.org/2019/732
2. Applebaum, B.: Cryptography in Constant Parallel Time. Information Security and Cryptography, Springer (2014)
3. Applebaum, B., Raykov, P.: On the relationship between statistical zero-knowledge and statistical randomized encodings. In: Robshaw and Katz [63], pp. 449–477
4. Badrinarayan, S., Fernando, R., Jain, A., Khurana, D., Sahai, A.: Statistical ZAP arguments. Cryptology ePrint Archive, Report 2019/780 (2019), https://eprint.iacr.org/2019/780
5. Ball, M., Dachman-Soled, D., Kulkarni, M.: New techniques for zero-knowledge: Leveraging inefficient provers to reduce assumptions and interaction. Cryptology ePrint Archive, Report 2019/1464 (2019), https://eprint.iacr.org/2019/1464
6. Ball, M., Rosen, A., Sabin, M., Vasudevan, P.N.: Average-case fine-grained hardness. In: Hatami, H., McKenzie, P., King, V. (eds.) 49th ACM STOC. pp. 483–496. ACM Press (Jun 2017)
7. Ball, M., Rosen, A., Sabin, M., Vasudevan, P.N.: Proofs of work from worst-case assumptions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 789–819. Springer, Heidelberg (Aug 2018)
8. Barak, B., Ong, S.J., Vadhan, S.: Derandomization in cryptography. SIAM Journal on Computing 37(2), 380–400 (2007)
9. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg (Aug 2003)
10. Barak, B., Pass, R.: On the possibility of one-message weak zero-knowledge. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 121–132. Springer, Heidelberg (Feb 2004)

11. Beimel, A., Gál, A.: On arithmetic branching programs. J. Comput. Syst. Sci. 59(2), 195–220 (1999)
12. Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: Security in the face of parameter subversion. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 777–804. Springer, Heidelberg (Dec 2016)
13. Bellare, M., Micali, S., Ostrovsky, R.: Perfect zero-knowledge in constant rounds. In: 22nd ACM STOC. pp. 482–493. ACM Press (May 1990)
14. Bellare, M., Yung, M.: Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. Journal of Cryptology 9(3), 149–166 (Jun 1996)
15. Ben-Or, M., Gutfreund, D.: Trading help for interaction in statistical zero-knowledge proofs. Journal of Cryptology 16(2), 95–116 (Mar 2003)
16. Ben-Sasson, E., Bentov, I., Chiesa, A., Gabizon, A., Genkin, D., Hamilis, M., Pergament, E., Riabzev, M., Silberstein, M., Tromer, E., Virza, M.: Computational integrity with a public random string from quasi-linear PCPs. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 551–579. Springer, Heidelberg (Apr / May 2017)
17. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046 (2018), https://eprint.iacr.org/2018/046
18. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable zero knowledge with no trusted setup. In: Boldyreva and Micciancio [22], pp. 701–732
19. Bitansky, N., Lin, H.: One-message zero knowledge and non-malleable commitments. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part I. LNCS, vol. 11239, pp. 209–234. Springer, Heidelberg (Nov 2018)
20. Bitansky, N., Paneth, O.: ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 401–427. Springer, Heidelberg (Mar 2015)
21. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC. pp. 103–112. ACM Press (May 1988)
22. Boldyreva, A., Micciancio, D. (eds.): CRYPTO 2019, Part III, LNCS, vol. 11694. Springer, Heidelberg (Aug 2019)
23. Campanelli, M., Gennaro, R.: Fine-grained secure computation. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 66–97. Springer, Heidelberg (Nov 2018)
24. Canetti, R. (ed.): TCC 2008, LNCS, vol. 4948. Springer, Heidelberg (Mar 2008)
25. Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G.N., Rothblum, R.D., Wichs, D.: Fiat-Shamir: from practice to theory. In: Charikar, M., Cohen, E. (eds.) 51st ACM STOC. pp. 1082–1090. ACM Press (Jun 2019)
26. Chailloux, A., Ciocan, D.F., Kerenidis, I., Vadhan, S.P.: Interactive and noninteractive zero knowledge are equivalent in the help model. In: Canetti [24], pp. 501–534
27. Ciocan, D.F., Vadhan, S.: Interactive and noninteractive zero knowledge coincide in the help model. Cryptology ePrint Archive, Report 2007/389 (2007), http://eprint.iacr.org/2007/389
28. De Santis, A., Micali, S., Persiano, G.: Non-interactive zero-knowledge with preprocessing. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 269–282. Springer, Heidelberg (Aug 1990)

29. Degwekar, A., Vaikuntanathan, V., Vasudevan, P.N.: Fine-grained cryptography. In: Robshaw and Katz [63], pp. 533–562
30. Ding, N., Gu, D.: Precise time and space simulatable zero-knowledge. In: ProvSec. Lecture Notes in Computer Science, vol. 6980, pp. 16–33. Springer (2011)
31. Ding, N., Gu, D.: On constant-round precise zero-knowledge. In: ICICS. Lecture Notes in Computer Science, vol. 7618, pp. 178–190. Springer (2012)
32. Dwork, C., Naor, M.: Zaps and their applications. SIAM Journal on Computing 36(6), 1513–1543 (2007)
33. Egashira, S., Wang, Y., Tanaka, K.: Fine-grained cryptography revisited. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 637–666. Springer, Heidelberg (Dec 2019)
34. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM Journal on Computing 29(1), 1–28 (1999)
35. Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 526–544. Springer, Heidelberg (Aug 1990)
36. Goldreich, O.: The Foundations of Cryptography - Volume 1: Basic Techniques. Cambridge University Press (2001)
37. Goldreich, O., Micali, S., Wigderson, A.: 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)
38. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. Journal of Cryptology 7(1), 1–32 (Dec 1994)
39. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on computing 18(1), 186–208 (1989)
40. Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. In: 18th ACM STOC. pp. 59–68. ACM Press (May 1986)
41. Goyal, V., Jain, A., Sahai, A.: Simultaneous amplification: The case of non-interactive zero-knowledge. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 608–637. Springer, Heidelberg (Aug 2019)
42. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (Aug 2006)
43. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. Journal of the ACM (JACM) 59(3), 11 (2012)
44. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (Apr 2008)
45. Impagliazzo, R.: A personal view of average-case complexity. In: Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference. pp. 134–147. IEEE (1995)
46. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: 21st ACM STOC. pp. 44–61. ACM Press (May 1989)
47. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: 41st FOCS. pp. 294–304. IEEE Computer Society Press (Nov 2000)
48. Ishai, Y., Kushilevitz, E.: Perfect constant-round secure computation via perfect randomizing polynomials. In: Widmayer, P., Ruiz, F.T., Bueno, R.M., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 244–256. Springer, Heidelberg (Jul 2002)

49. Itoh, T., Ohta, Y., Shizuya, H.: Language dependent secure bit commitment. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 188–201. Springer, Heidelberg (Aug 1994)
50. Jain, A., Jin, Z.: Statistical zap arguments from quasi-polynomial LWE. Cryptology ePrint Archive, Report 2019/839 (2019), https://eprint.iacr.org/2019/839
51. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 158–189. Springer, Heidelberg (Aug 2017)
52. Kalai, Y.T., Khurana, D., Sahai, A.: Statistical witness indistinguishability (and more) in two messages. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 34–65. Springer, Heidelberg (Apr / May 2018)
53. Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 622–651. Springer, Heidelberg (May 2019)
54. Kim, S., Wu, D.J.: Multi-theorem preprocessing NIZKs from lattices. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 733–765. Springer, Heidelberg (Aug 2018)
55. Lapidot, D., Shamir, A.: Publicly verifiable non-interactive zero-knowledge proofs. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO'90. LNCS, vol. 537, pp. 353–365. Springer, Heidelberg (Aug 1991)
56. LaVigne, R., Lincoln, A., Williams, V.V.: Public-key cryptography in the fine-grained setting. In: Boldyreva and Micciancio [22], pp. 605–635
57. Micali, S., Pass, R.: Local zero knowledge. In: STOC. pp. 306–315. ACM (2006)
58. Nisan, N., Wigderson, A.: Hardness vs. randomness (extended abstract). In: 29th FOCS. pp. 2–11. IEEE Computer Society Press (Oct 1988)
59. Ong, S.J., Vadhan, S.P.: An equivalence between zero knowledge and commitments. In: Canetti [24], pp. 482–500
60. Ostrovsky, R., Wigderson, A.: 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. pp. 3–17 (1993)
61. Pass, R., shelat, A.: Unconditional characterizations of non-interactive zero-knowledge. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 118–134. Springer, Heidelberg (Aug 2005)
62. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 89–114. Springer, Heidelberg (Aug 2019)
63. Robshaw, M., Katz, J. (eds.): CRYPTO 2016, Part III, LNCS, vol. 9816. Springer, Heidelberg (Aug 2016)
64. Toda, S.: Counting problems computationally equivalent to. SIAM J. Computing 13, 423–439 (1984)
65. Wu, H., Wang, F.: A survey of noninteractive zero knowledge proof system and its applications (2014), https://doi.org/10.1155/2014/560484