Electronic Colloquium on Computational Complexity, Report No. 149 (2024)

Rate-1 Zero-Knowledge Proofs from One-Way Functions

Noor Athamnah[1], Eden Florentz – Konopnicki[1], and Ron D. Rothblum[1]

Technion, Taub Faculty of Computer Science
noor.athamnah@gmail.com
edenko@campus.technion.ac.il
rothblum@cs.technion.ac.il

Abstract. We show that every NP relation that can be verified by a bounded-depth polynomial-sized circuit, or a bounded-space polynomial-time algorithm, has a computational zero-knowledge proof (with statistical soundness) with communication that is only *additively larger* than the witness length. Our construction relies only on the minimal assumption that one-way functions exist.

In more detail, assuming one-way functions, we show that every NP relation that can be verified in NC has a zero-knowledge proof with communication $|w| + \text{poly}(\lambda, \log(|x|))$ and relations that can be verified in SC have a zero-knowledge proof with communication $|w|+|x|^{\varepsilon}\cdot\text{poly}(\lambda)$. Here $\varepsilon > 0$ is an arbitrarily small constant and λ denotes the security parameter. As an immediate corollary, we also get that *any* NP relation, with a size S verification circuit (using unbounded fan-in XOR, AND and OR gates), has a zero-knowledge proof with communication $S + \text{poly}(\lambda, \log(S))$.

Our result improves on a recent result of Nassar and Rothblum (Crypto, 2022), which achieves length $(1+\varepsilon)\cdot|w| + |x|^{\varepsilon}\cdot\text{poly}(\lambda)$ for bounded-space computations, and is also considerably simpler. Building on a work of Hazay et al. (TCC 2023), we also give a more complicated version of our result in which the parties only make a *black-box* use of the one-way function, but in this case we achieve only an inverse polynomial soundness error.

1 Introduction

Zero-knowledge proofs, introduced in the groundbreaking work of Goldwasser Micali and Rackoff [GMR89], are interactive protocols in which a powerful but untrusted prover convinces a verifier of the validity of a computational statement, in such a way, that no additional information is revealed. Different notions of zero-knowledge have been studied in the literature. In this work we focus exclusively on proofs offering *statistical soundness* and *computational zero-knowledge*, and refer to this notion whenever we say zero-knowledge proof (see Remark 4 for a discussion of related variants).

In their seminal work, Goldreich, Micali and Wigderson [GMW86] constructed a zero-knowledge proof for checking that a given graph is 3-colorable (assuming

ISSN 1433-8092

the existence of one-way functions). As 3-coloring is NP-complete, their result yielded the amazing fact that every problem in NP (i.e., every problem possessing a classical proof) also has a zero-knowledge proof.

The protocol of [GMW86], henceforth referred to as GMW, proceeds by having the prover commit to a random 3-coloring of the graph $G = (V, E)$, the verifier chooses an edge and the prover decommits to the colors of the two endpoints. In order to get soundness error $2^{-\lambda}$, this base protocol is repeated sequentially[1] $\Theta(|E| \cdot \lambda)$ times.

Thus, the overall communication in the GMW protocol is $|V| \cdot |E| \cdot \text{poly}(\lambda)$. This should be contrasted with the direct NP proof which has length $|V| \cdot \log_2(3)$.[2] Things becomes even worse when considering general NP languages – for such languages, due to the Karp reduction to 3-coloring, the GMW protocol gives communication that is (a relatively large) polynomial in the complexity of the NP verification circuit rather than the length of the raw witness. A similar overhead is incurred by other classical approaches such as Blum's [Blu86] Hamiltonicity protocol.

It is natural to wonder whether the overhead incurred by these protocols is inherent. This question has been studied in several works that show that it is possible to achieve communication that is polynomial in the witness length, rather than the size of the verification circuit, for a large subclass[3] of NP relations [IKOS09, KR08, GKR15, RRR21, NR22, HVW23]. Similarly to the original GMW protocol, the protocols in this line of work all rely on the existence of one-way functions, an assumption that is also known to be necessary for the construction of zero-knowledge proofs for NP [OW93, HN24]. A different approach, proposed by Gentry *et al.* [GGI+15], constructs zero-knowledge proofs with communication $m + \text{poly}(\lambda)$, where m is the size of the witness and λ is the security parameter, but relies on the existence of a *fully homomorphic encryption scheme* (FHE), which is (believed to be) a much stronger assumption (and is currently only known to be instantiable assuming the circular security of LWE [Gen09, BV11, MV24], or via indistinguishibility obfuscation [CLTV15]).[4]

All the aforementioned results that rely on one-way functions incur at the very least a large multiplicative blowup over the witness size. In a recent work, Nassar and Rothblum [NR22], relying only on the existence of one-way functions, showed that any bounded space NP relation, has a zero-knowledge proof with length $(1+\gamma) \cdot m + n^{\beta} \cdot \text{poly}(\lambda)$, where m is the witness length, n is the instance length, λ is the security parameter and $\gamma, \beta > 0$ are arbitrarily small constants.

[1] While it may seem natural to repeat the protocol in parallel, this is insecure, see [HLR21].

[2] For this high-level discussion, we ignore minor issues arising from rounding and efficient bit-representation of trits.

[3] The results obtained in this line of work differ, but loosely speaking, other than [GGI+15], all known results hold for NP relations that can be decided by either bounded depth circuits or by bounded space algorithms.

[4] Gentry *et al.* [GGI+15] focus on *non-interactive* zero-knowledge proofs, but note that their approach is also applicable to interactive zero-knowledge.

1.1 Our Results

As our main result, we construct zero-knowledge proofs, with communication that is *only additively larger than the witness length*, for any NP relation that can be verified either by a bounded space algorithm or by a bounded-depth circuit. Our constructions rely only on the minimal assumption of one-way functions.

Theorem 1 (Succinct Zero-Knowledge for Bounded Depth). *Assume that one-way functions exist. Let R be an* NP *relation with input size n and witness size m, that can be decided by a polynomial-size circuit with depth D and assume $n \leq \mathrm{poly}(m)$. Then, R has a zero-knowledge proof with soundness error $2^{-\lambda}$ and communication complexity $m+\mathrm{poly}(\lambda, \log(m), D)$ where λ denotes the security parameter.*

Furthermore, the prover and verifier run in time $\mathrm{poly}(n, \lambda)$, the protocol is public-coin and the number of rounds is $\mathrm{poly}(\lambda, \log(m), D)$.

Theorem 2 (Succinct Zero-Knowledge for Bounded Space). *Assume that one-way functions exist. Let R be an* NP *relation with input size n and witness size m that can be decided by a polynomial-time and space S algorithm and assume $n \leq \mathrm{poly}(m)$. Then, for every constant $\delta > 0$, the relation R has a zero-knowledge proof with soundness error $2^{-\lambda}$, and communication complexity $m + n^{\delta} \cdot \mathrm{poly}(S, \lambda)$, where λ denotes the security parameter.*

Furthermore, the prover and verifier run in time $\mathrm{poly}(m, \lambda)$, the protocol is public-coin and the number of rounds is $\mathrm{poly}(\lambda)$.

Theorems 1 and 2 improve on the result in [NR22] in that they achieve a truly *additive* overhead in communication over the raw witness length (in contrast to the the $(1+\gamma)$ multiplicative overhead in [NR22]).[5] For example, Theorem 1 implies that satisfiability of a polynomial-size formula on n-variables has a zero-knowledge proof with communication $n + \mathrm{poly}(\lambda, \log n)$, and 3-colorability of an n-vertex graph has a zero-knowledge proof with communication $n \cdot \log_2(3) + \mathrm{poly}(\lambda, \log n)$.

These results are optimal in two ways. First, in terms of assumptions, they only rely on the minimal assumption that one-way functions exist [OW93,HN24]. Second, in terms of communication, assuming the strong exponential-time hypothesis (SETH) [IP99], the witness length is a lower bound on communication (up-to additive terms), due to known limitations on so-called "laconic" provers [GH98, GVW02]. Given the above, we refer to zero-knowledge proofs with a strictly additive overhead over the witness as having *rate*-1.

The proofs of Theorems 1 and 2 are also significantly simpler than that of [NR22] (which relied on recent non-trivial results on high-rate interactive oracle proofs (IOPs) [RR20]). The key idea, on which we elaborate in Section 1.2, is a form of "hybrid zero-knowledge" and is inspired by the construction in

[5] Our result is also more general than that of [NR22] in that it holds also for bounded depth circuits, whereas [NR22] is explicitly only stated for bounded space algorithms. Nevertheless, by relying on [RR20, Remark 1.5], the techniques of [NR22] could also yield proofs with communication roughly $(1+\gamma) \cdot m$ for bounded-depth circuits.

[GGI+15] (and can be further traced back to hybrid encryption). In a nutshell, we give a simple reduction from constructing a rate-1 zero-knowledge proof, to constructing a zero-knowledge with communication that can depend polynomially on the *witness length* (rather than the size of the verification circuit). Theorems 1 and 2 then follow by combining our reduction with known zero-knowledge proofs from the literature.

It is worth pointing out some second order differences between Theorem 1 and Theorem 2, which are inherited from doubly-efficient interactive-proofs on which they rely. The additive term in the communication in Theorem 1 depends only poly-logarithmically on the input-size, whereas in Theorem 2 the dependence has the form n^{δ}. On the other hand, the round complexity in Theorem 2 is $\text{poly}(\lambda)$ whereas in Theorem 1 the number of rounds also depends poly-logarithmically on the witness size and linearly on the depth.

Circuit-size Communication for General NP *Relations.* Theorem 1 also yields a new zero-knowledge proof for *general* NP relations with communication that is only additively larger than the *size* of the verification circuit. This essentially follows from the NP completeness of SAT: any NP relation can be verified in (poly-)logarithmic depth, if the witness includes the values obtained by all of the gates in the evaluation of the verification circuit (indeed, this "extended witness" can be checked by verifying that each gate is separately satisfied by the assignment).

Corollary 3 (Succinct Zero-Knowledge for General Relations). *Assume that one-way functions exist. Let R be an* NP *relation that can be verified by a circuit of size S with unbounded fan-in XOR, AND and OR gates. Then, R has a zero-knowledge proof with soundness error $2^{-\lambda}$ and communication complexity $S + \text{poly}(\lambda, \log(S))$, where λ is the security parameter.*

Corollary 3 improves over a similar result for general circuits obtained by [IKOS09], which had a constant multiplicative overhead, and a result that can be derived from [NR22], which gives $(1 + \varepsilon)$ multiplicative overhead.

Remark 4 (On Computationally Sounds Proofs). In contrast to the statistically sound proofs considered in this work, it is well-known that (assuming the existence of collision-resistant hash functions) there exist zero-knowledge *arguments* (aka computationally sound proofs) in which the communication is *substantially smaller than the witness length* [Kil92].

Our focus however is on the case of statistical soundness. In this case, assuming reasonable hardness assumptions, the witness length poses a barrier on the communication [GH98, GVW02].

1.1.1 Zero-Knowledge with Black-Box use of the OWF

Many of the aforementioned constructions of succinct zero-knowledge proofs, including the protocols establishing Theorems 1 and 2, make a *non black-box* use of the one-way function. This means that the implementation of the prover and

the verifier depends on the actual *code* of the one-way function. This is in contrast to a black-box construction in which it suffices for the parties to receive oracle access to the one-way function (in other words, the one-way function is merely used as a sub-routine). Non black-box constructions are usually considered less efficient than their black-box counterparts and it is therefore desirable to construct protocols that avoid such a non black-box use of the one-way function. Such constructions are also more modular, enabling applications that may not be possible otherwise (see [KRV24] for a recent example).

The MPC-in-the-head framework of Ishai *et al.* [IKOS09] gives an alternate approach that enables a black-box use of the one-way function. In particular, a very recent work by Hazay, Venkitasubramaniam and Weiss [HVW23] builds on this framework to construct *black-box* zero-knowledge proofs with communication roughly $(1+\varepsilon)\cdot m$, thereby matching the non black-box result of [NR22]. A downside of their result, compared to [NR22], is that they only achieve a constant soundness error (which cannot be reduced by repetition unless we blowup the communication).

Our second set of results are zero-knowledge proofs with a black-box use of the one-way function, that improve on the result of [HVW23] in two ways. Our main improvement is that we obtain proof length that is only additively larger than m – i.e., a rate-1 zero-knowledge proof (improving on the $(1+\varepsilon)$ multiplicative overhead in [HVW23]). The second improvement is that we obtain soundness error that is polynomial in the (reciprocal of) the security parameter, thereby improving on the constant error achieved in [HVW23]. Our construction also avoids the use of the relatively heavy hammer of high-rate IOPs used by [HVW23] and relies on more basic tools (e.g., the doubly-efficient interactive proof of [GKR15]).

Theorem 5. *Assume that one-way functions exist. Let R be an* NP *relation with input size n and witness size m, that is computable by a (non-uniform) circuit family C of size $S = S(n)$ and depth $D = D(n)$ and assume $n \leq \mathrm{poly}(m)$. Then, for any $\varepsilon > 0$ the relation R has a zero-knowledge proof with perfect completeness, and soundness error ε, in which the verifier, prover and simulator all only make a black-box use of the one-way function. The communication complexity is $m + \mathrm{poly}\left(\frac{1}{\varepsilon}, \lambda, D, \log(S)\right)$, where λ is the security parameter.*

Furthermore, the prover and verifier run in polynomial time, the protocol is public-coin and the number of rounds is $\mathrm{poly}(D, \log(S))$.

We remark that a similar result to Theorem 5 for bounded space computations can also be obtained, see discussion in Section 1.2.2.

1.2 Techniques

In this section we give an overview of our techniques.

1.2.1 Rate-1 Zero-Knowledge: Proving Theorems 1 and 2

As mentioned above, the proofs of Theorems 1 and 2 are surprisingly simple. The key idea behind the protocols, which is inspired by [GGI+15], is to reduce

the construction of rate-1 zero-knowledge proofs, to constructing zero-knowledge proofs whose communication depends polynomially only on the witness length.

The protocol proceed as follows. Given an input x and its witness w, the prover randomly samples a short seed $s \in \{0,1\}^\lambda$ for a pseudorandom generator (PRG) G and then uses it to mask the witness. That is, the prover computes and sends $u = G(s) \oplus w$ to the verifier.

At this point, we view the PRG seed s as playing the role of a new witness, in the sense that if the verifier knew s then she could verify that x is indeed in the language. Needless to say, sending s in the clear would violate the zero-knowledge property, but we observe that it now suffices for the the prover to prove, in zero-knowledge, that it could have revealed an s that would have made the verifier accept. The key benefit is that s, which serves as the new witness, is much shorter than the original witness. Hence, we can afford to use one of the pre-existing zero-knowledge proofs that have a *polynomial overhead* in the witness. Thus, while the first message sent has length exactly $m = |w|$, the length of the messages sent afterwards is polynomial in the length of the seed s.

In more detail, given an NP relation R, the prover generates $u = G(s) \oplus w$ and we consider the relation $R'_G = \left\{ ((x,u),s) \mid (x, G(s) \oplus u) \in R \right\}$. Observe that the tuple $((x,u),s)$ is in R'_G if and only if $(x, G(s) \oplus u) \in R$. So by sending u, we have reduced the problem to one with a smaller witness.

The relation R'_G is in NP, since given s we can compute $G(s)$ in polynomial time in $|w|$, and then run the NP verifier on $(x, G(s) \oplus u)$, which can also be done in $\text{poly}(|x|)$ time. Moreover, we observe that if R can be decided in small depth then so can R'_G – this follows from the fact that, assuming one-way functions, there exists a PRG $G : \{0,1\}^\lambda \to \{0,1\}^m$ computable by depth $\log(m) \cdot \text{poly}(\lambda)$ (and size $\text{poly}(m, \lambda)$) circuits. Indeed such a PRG follows by using a stretch-doubling PRG (which can be constructed from a one-way function [HILL99]) and applying the [GGM86] tree-based construction for $\log(m)$ levels. Therefore, since the relation R'_G is verifiable in small depth, using pre-existing results from the literature[6] [GKR15], this relation has a zero-knowledge proof with a communication complexity $\text{poly}(\lambda, \log(n), D)$, where D denotes the depth of the original verification circuit.

Overall, we obtain a zero-knowledge proof for R with communication complexity that is larger than the witness length only by an additive $\text{poly}(\lambda, \log(n), D)$ factor.

For relations R that can be verified in small space we follow a similar approach, using a zero-knowledge proof for small space relations with polynomial overhead in the witness size [RRR21] and using a PRG computable in small space (such a PRG follows essentially by the textbook stretch increasing construction of a PRG, see [Gol01, Construction 3.3.2]).

[6] The main result in [GKR15] is a doubly-efficient interactive proof for bounded-depth computations, which is not zero-knowledge. We use here a corollary [GKR15, Theorem 1.6] that obtains *computationally* zero-knowledge proofs for bounded depth NP relations.

1.2.2 Rate-1 Zero-Knowledge: The Black-Box Way

A downside of the approach described in Section 1.2.1 is that the prover and verifier make a non black-box use of the one-way function. This is due to the fact that the new relation R'_G has the PRG G encoded as part of its specification circuit. All known general purpose zero-knowledge proofs need an explicit representation of the NP verification circuit (and this is inherent, see [Ros12]). Thus, a zero-knowledge proof for R'_G has to use the code of the PRG, which translates into a non black-box use of the one-way function that G is based on.

In this section we present a different approach in which the prover, verifier and simulator all make a black-box use of the one-way function. A caveat of this alternate approach is that we only get a soundness error that is polynomially related to the (reciprocal of the) security parameter, rather than an exponentially small soundness error as in Theorems 1 and 2. Nevertheless, this already significantly improves on the constant soundness error achieved by the previous black-box construction of Hazay *et al.* [HVW23].

We continue with the idea of hiding the witness by masking it with a PRG, but rather than employing an off-the-shelf zero-knowledge protocol to prove that the masked witness can be opened, we use a general interactive protocol, which is not zero-knowledge, and make it zero-knowledge by applying multi-party computation (MPC) techniques, details follow.

Following [HVW23] (although their idea is not quite articulated in the same way), our main step is constructing a form of "distributed zero-knowledge", in which we have a single prover and k verifiers. The goal is for the verifiers to each be convinced that the prover holds a valid witness, but in such a way that a subset $t < k$ of the verifiers does not learn anything else. The distributed zero-knowledge protocol can then be compiled into a standard one (i.e., with a single monolithic verifier), using the MPC-in-the head approach: the prover emulates the interaction between the k verifiers via an MPC protocol, and sends commitments to their views. The monolithic verifier can now request that some of these views be opened to check that the MPC protocol was executed correctly.

In order to get our desired communication complexity, we therefore need for the overall communication in the k-party distributed protocol to be roughly $m+\mathrm{poly}(\lambda)$, and for the MPC-in-the-head emulation to only increase this additively.

The Distributed Zero-Knowledge Protocol. We start by secret sharing the witness w to the k verifiers, where, for $i \in [k-1]$ the share is a PRG seed s_i and the remaining k-th share is set to $w \oplus (\bigoplus_{i=1}^{k-1}(G(s_i))$. Note that this is indeed a secret sharing of the witness w, since by expanding the seed and XORing, we can recover the witness. Also, the overall communication of this step is at most $m + k \cdot \mathrm{poly}(\lambda)$ as desired.

Assume that the NP relation is decidable in small depth. The prover now starts an execution of the doubly-efficient GKR protocol for bounded-depth computations [GKR15] to prove that $(x, w) \in R$. In each round in the protocol, rather than sending the next GKR prover message in the clear, the prover secret shares it between the parties. Since the GKR protocol is public-coin, the

GKR verifier's messages can be generated by some global source of randomness. Note that for circuits verifiable in NC, the communication in this part is just $k \cdot \text{polylog}(n)$.

At the end of the interactive phase, the GKR verifier needs to check some predicate on (x, w) and the interaction transcript. Since our verifier is distributed, we perform this task via an off-the-shelf (semi-honest secure) MPC protocol.

The idea so far yields a zero-knowledge proof, but hides a somewhat subtle flaw. The circuit which the parties emulate via the MPC protocol needs to fully expand the k PRG seeds, recover the witness and then check that the witness is valid. This means that the size of the circuit is at least $k \cdot m$, which increases the complexity of the MPC protocol beyond what we can afford.

Holography to the Rescue. To resolve this difficulty, we recall an extremely useful property of the GKR protocol (as well as related protocols in the literature) – it is a *holographic proof* [BFLS91, GR17]. Namely, the GKR verifier does not need full access to its input w but rather only to compute a single point in the low degree extension of w.[7] Moreover, the desired point depends only on the verifier's randomness.

Given this, rather than applying the MPC protocol on the full shares of w, we have each of the k verifiers compute its local contribution to the low degree extension at the desired point. Here we crucially use the fact that both the secret-sharing and low-degree extension are computed as linear[8] functions, and so the sum of contributions of the shares is indeed equal to the low degree extension of w at the desired point.

Thus, the MPC protocol only needs to recombine these small shares and then run the GKR verification step.

Compiling into a Monolithic Verifier. We now compile the distributed protocol to one with a single monolithic verifier. The prover simply sends commitments to all of the shares of the witness and the messages that were sent to the k parties, and then runs the MPC "in the head". To maintain short communication, for the last share of w, which has length m, we use a commitment scheme with only additive overhead (which can be achieved similarly to our original construction by XORing with the output of a PRG).

After the prover simulates the MPC, it sends commitments to the parties views. The verifier then chooses t parties at random and asks the prover to decommit to everything concerning these parties. Assuming the MPC protocol

[7] Recall that a low degree extension of a string $w \in \{0,1\}^m$ is a low degree multivariate polynomial that agrees with w on a prescribed sub-domain (see Section 2.2 for details) For our purposes it will only be important that the low-degree extension is a *linear* function.

[8] It is important here that these procedures are linear over the same field. To do this, we employ the GKR protocol over a characteristic 2 fields, in which case the additive secret sharing can be via an XOR (a linear function over such fields). Also, the secret sharing is not quite linear, because the k-th share is a PRG seed that first needs to be expanded, but this suffices for our approach.

has perfect correctness, the only way for the prover to cheat is by providing one of the players with an incorrect view or a pair of players with inconsistent views. The former case is caught with probability at least t/k and the latter with probability at least $\frac{t\cdot(t-1)}{k\cdot(k-1)}$. So the last thing for the verifier to do is to check the validity of all decommitments and that all revealed parties behaved consistently with the protocol.

In order to get a polynomially small soundness error, we set $k = \text{poly}(\lambda)$ and use as our MPC protocol the semi-honest GMW protocol [GMW87] in the OT-hybrid model, which offers perfect semi-honest security against $t = k - 1$ parties. This yields a soundness error of $O(1/k)$.

We remark that an analogous result for bounded space computations can also be obtained by replacing the [GKR15] that we used with the [RRR21] protocol for bounded space computations, which is also holographic.

Remark 6 (On Negligible Soundness via Malicious MPC). Following [IKOS09], it is natural to try to improve the above and obtain a negligible soundness error by relying on an MPC protocol with *malicious security*. Recall that in malicious MPC the computation is robust even if a constant fraction of the parties are corrupt. The idea would then be for the verifier to request to a open a constant fraction of the parties views such that either she will identify one of the corrupt parties and reject or, if all the opened views are consistent, the computation should be correct.

The reason why this attempt fails is that malicious MPC robustness holds at the condition that the function has the same output no matter the input of the corrupted parties. In the case where the function being computed by the parties is the NP verification of the relation, this attempt would work, since no matter what the witness is, the function should reject. However, we apply the MPC on a much simple function (which recover the low degree extension at just a single point) and changing the input of even just one party can change the result of the computation.

1.3 Open Questions

The main open question left by our work is constructing rate-1 zero-knowledge proofs for *all* NP relations. By the aforementioned result of [GGI+15], such proofs are known to exist assuming (full-fledged) FHE, but the question is whether a similar result can be established from a weaker assumption; ideally, just from the minimal assumption of one-way functions. We remark that, using our hybrid zero-knowledge approach, such a rate-1 zero-knowledge proof would follow from the existence of a zero-knowledge proof for NP that has an arbitrary polynomial dependence on the witness length (but does not scale linearly with the size of the verification circuit).

A second question left open by our work is whether we can construct succinct zero-knowledge proofs that use the one-way function as a black-box, but achieve a negligible soundness error (in contrast to the inverse polynomial soundness error achieved by our construction). We remark that [IKOS09], building on [DI06],

give such a result with communication $O(|C|)+\text{poly}(\lambda, \log(|C|))$, where C is the size of the NP verification circuit. This falls short of our goal of additive overhead over the NP *witness*. Actually, to the best of our knowledge it is not even known how to construct such protocols (i.e., with black box use of the one-way function and negligible soundness error) with communication $\text{poly}(m) + \text{poly}(\lambda, \log(m))$ (even for NP relations that are decidable in NC).

1.4 Organization

Preliminaries are in Section 2. In Section 3 we construct the succinct zero-knowledge proofs that establish Theorems 1 and 2. The constructions that make a blackbox use of the one-way function, proving Theorem 5, are in Section 4.

2 Preliminaries

For an NP relation R, we denote by L_R the language $L_R = \{x : \exists w, \text{ s.t. } (x, w) \in R\}$. Throughout this work we use n to denote the instance size $|x|$, and m to denote the witness size $|w|$.

2.1 Computational Indistinguishably

Definition 7. *Let $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}, E = \{E_\lambda\}_{\lambda\in\mathbb{N}}$ be two distribution ensembles indexed by a security parameter λ. We say that the ensembles are* computationally indistinguishable*, denoted $D \overset{c}{\approx} E$, if for any family of polynomial size circuits $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, the following quantity is a negligible function in λ:*

$$\left| \Pr_{x\leftarrow D_\lambda} [C_\lambda(x) = 1] - \Pr_{x\leftarrow E_\lambda} [C_\lambda(x) = 1] \right|.$$

Fact 8 (Computational Data-Processing Inequality) *If the distributions D and E are computationally indistinguishable, and A is a PPT algorithm, then $A(D)$ and $A(E)$ are also computationally indistinguishable.*

2.2 Interactive Proofs

Definition 9 (Interactive proof). *A pair of interactive machines (P, V) is called an* interactive proof system *for a language L, if V is a probabilistic polynomial-time machines, and the following conditions hold for every security parameter $\lambda \in \mathbb{N}$:*

- ***Completeness:*** *For every $x \in L$, V accepts with probability 1 when interacting with P on common input $(x, 1^\lambda)$.*
- ***Soundness:*** *For every $x \notin L$, and every prover P^*, V accepts with probability at most $\varepsilon(\lambda)$ when interacting with P^* on common input $(x, 1^\lambda)$.*

We say that an interactive proof has an efficient prover *if P can be implemented in (probabilistic) polynomial-time. In the context of an interactive proof for an* NP relation*, we allow the prover access to* an NP witness.

We remark that all proofs that we construct in this work will have an efficient prover.

The Interactive Proof-System of [GKR15]. Our construction will build on the interactive proof-system of [GKR15]. This protocol relies on the *multi-linear extension encoding*, which we recall next. Let $\mathbb{F}$ be a finite field and d an integer. The multi-linear extension of a function $f : \{0,1\}^d \to \mathbb{F}$ is the unique multi-linear polynomial (over $\mathbb{F}$) such that $\hat{f}(x) = f(x)$, for all $x \in \{0,1\}^d$. A multi-linear extension of a string $w \in \{0,1\}^d$ can be defined by viewing the string as the truth-table of a function $f_w : \{0,1\}^{\log(d)} \to \{0,1\}$. The multi-linear extension can be explicitly written as:

$$\hat{f}(x) = \sum_{h \in \{0,1\}^d} f(h) \cdot I(x,h)$$

where

$$I(x,h) = \prod_{j \in [d]} \big(x_j \cdot h_j + (1 - x_j) \cdot (1 - h_j)\big).$$

This formula also directly shows that the multi-linear extension at a given point $x \in \mathbb{F}^d$ can be computed in time $2^d \cdot \mathrm{poly}(\log(\mathbb{F}))$.

Theorem 10 (Follows from [GKR15, Theorem 1.5]). *Let L be a language computable by a (non-uniform) circuit family C of size $S = S(n)$ and depth $D = D(n)$. Let $\mathbb{F} = \mathbb{F}(n)$ be a constructible field ensemble. Then, there exists a two phase public-coin interactive proof $(P, V_{interactive}, V_{post})$ with the following properties*

1. *In the interactive phase $(P, V_{interactive})$, P gets as input (C,x) and $V_{interactive}$ gets only $S = |C|$. The prover P runs in time $\mathrm{poly}(S)$, and $V_{interactive}$ runs in time $D \cdot \mathrm{poly}(\log(S), \log(|\mathbb{F}|))$. Denote by* transcript *all messages sent between the parties. The communication complexity of the interactive phase is $\mathrm{poly}(D, \log(S), \log(|\mathbb{F}|))$.*
2. *From* transcript *and the circuit C we can derive $z \in \mathbb{F}^d$, $\alpha \in \mathbb{F}$ and $\langle C \rangle \in \{0,1\}^{\mathrm{poly}(D, \log(S), \log(|\mathbb{F}|))}$ in time $\mathrm{poly}(S)$.*
3. *V_{post} gets as input $\big(\text{transcript}, \langle C \rangle, \hat{x}(z)\big)$ and either accepts or rejects. V_{post} performs a test on $(transcript, \langle C \rangle)$ and checks the claim $\hat{x}(z) = \alpha$. V_{post} runs in time $\mathrm{poly}(D, \log(S), \log(|\mathbb{F}|))$.*

The interactive protocol obtained by first running the interactive phase, then having the verifies derive $\langle C \rangle, \hat{x}(z)$ and finally running V_{post}, has perfect completeness and soundness error $O\left(\frac{D \log S}{|\mathbb{F}|}\right)$.

We remark that [GKR15, Theorem 1.5] does not separate the proof-system into two phases as above. However, such a separation follows easily using the fact that the GKR protocol is *holographic*, meaning that the verifier's only needs to preprocess some queries to the low degree extension of the input prior to the interaction, and subsequently runs in $\text{poly}(D, \log(S))$ time.

2.3 Zero-Knowledge Proofs

Next, we recall the definition of zero-knowledge proofs. For sake of simplicity we focus on the basic standalone definition but note that our constructions also achieve the stronger notion of *auxilary-input zero-knowledge*.

Definition 11 (Zero-knowledge proofs). *Let (P, V) be an interactive proof system for an* NP *relation R with security parameter λ. The proof-system (P, V) is* computational zero-knowledge *if for every polynomial-time interactive machine $\hat{V}$ there exists a probabilistic polynomial-time machine Sim, called the simulator, such that for every ensemble $(x, w) = (x_\lambda, w_\lambda)_\lambda$, with $(x_\lambda, w_\lambda) \in R$ the following distribution ensembles are computationally indistinguishable:*

- $\left\{View_{\hat{V}}^{P(w)}\left(x, 1^\lambda\right)\right\}_{\lambda \in \mathbb{N}}$, *and*
- $\left\{Sim\left(x, 1^\lambda\right)\right\}_{\lambda \in \mathbb{N}}$.

Succinct Zero-Knowledge Proofs. Next, we state two prior works obtaining succinct zero-knowledge proofs for bounded depth and bounded space computations. In contrast to our results, these prior works have a large multiplicative overhead over the witness length.

Theorem 12 ([GKR15, Theorem 1.6]). *Assume one-way functions exist, and let $\lambda = \lambda(n) \geq \log(n)$ be a security parameter. Let L be a language in* NP/poly*, whose relation R can be computed on inputs of length n with witnesses of length $m = m(n)$ by Boolean circuits of size* $\text{poly}(n)$ *and depth $d(n)$. Then L has a zero-knowledge interactive proof:*

1. *The prover runs in time* $\text{poly}(n, \lambda)$ *(given an* NP *witness), the verifier runs in time* $\text{poly}(n, \lambda)$ *and number of rounds is* $\text{poly}(\lambda, d(n))$.
2. *The protocol has perfect completeness and soundness error* $2^{-\lambda}$.
3. *The protocol is public-coin, with communication complexity* $m \cdot \text{poly}(\lambda, d(n))$.

Remark 13. The theorem statement in [GKR15] (i.e., [GKR15, Theorem 1.6]) does not explicitly state the number of rounds, but it can be inferred in a straightforward manner from the protocol. Additionally, the stated soundness error there is $\frac{1}{2}$, but the protocol can be repeated λ times (sequentially) to achieve a soundness error of $2^{-\lambda}$.

Theorem 14 ([RRR21, Theorem 2]). *Assume one-way functions exist, and let $\delta > 0$ be a constant. Let R be an* NP *relation, with instance length n, and witness length m that can be verified by a* $\text{poly}(m)$*-time and space $S = S(m)$*

Turing Machine, where $n \leq \mathrm{poly}(m)$*. Then, the relation* R *has a public-coin zero-knowledge interactive proof with perfect completeness, constant soundness error, and communication complexity* $(m + S(m)) \cdot m^{\varepsilon} \cdot \mathrm{poly}(\lambda)$*. The (honest) prover, given a valid witness, runs in time* $\mathrm{poly}(m, \lambda)$*. The verifier runs in time* $\mathrm{poly}(m, \lambda)$*.*

2.4 Pseudorandom Generator

Definition 15 (Pseudorandom generator). *A pseudorandom generator (PRG) is a deterministic polynomial-time algorithm* G *satisfying the following two conditions:*

1. ***Expansion:*** *There exists a function* $\ell : \mathbb{N} \to \mathbb{N}$ *such that* $\ell(\lambda) > \lambda$ *for all* $\lambda \in \mathbb{N}$*, and* $|G(s)| = \ell(|s|)$ *for all* $s \in \{0,1\}^*$*.*
2. ***Pseudorandomness:*** *The ensembles* $\{G(U_\lambda)\}_\lambda$ *and* $\{U_{\ell(\lambda)}\}_\lambda$ *are computationally indistinguishable.*

Proposition 16 *Assuming one-way functions exist, for every polynomial* $\ell = \ell(\lambda)$*, there exist a PRG* $G : \{0,1\}^\lambda \to \{0,1\}^\ell$ *computable by circuits of size* $\mathrm{poly}(\lambda, \ell)$ *and depth* $\mathrm{poly}(\lambda) \cdot \log(\ell)$*.*

We emphasize that here (as well as in Theorem 17 below) poly refers to a fixed polynomial that is independent of ℓ.

Theorem 16 follows from the tree based PRF construction of Goldreich *et al.* [GGM86] (see also [Gol01, Construction 3.6.5]), where we simply output the $\log(\ell)$-th layer of the tree (where the root is at layer 0).

Proposition 17 *Assuming one-way functions exist, for every polynomial* $\ell = \ell(\lambda)$*, there exists a PRG* $G : \{0,1\}^\lambda \to \{0,1\}^\ell$ *computable by a time* $\mathrm{poly}(\lambda, \ell)$ *and space* $\mathrm{poly}(\lambda) + \log(\ell)$ *Turing machine.*

Theorem 17 follows from the standard stretch-increasing PRG construction (see [Gol01, Construction 3.3.2]).

2.5 Commitment Scheme

Next, we define commitment schemes. We focus on non-interactive statistically binding commitments in the common random string (CRS) model, which can be constructed from one-way functions.

Definition 18 (Commitment scheme). *A commitment scheme in the CRS model is a tuple of probabilistic polynomial-time algorithms* (Gen, Com, Ver) *with the following semantics:*

1. $crs \leftarrow Gen(1^\lambda)$*, where* crs *is referred to as the common reference string.*
2. *For any string* $m \in \{0,1\}^* : (com, dec) \leftarrow Com(crs, m)$*.*
3. *For any* $com, dec, m \in \{0,1\}^* : \{0,1\} \leftarrow Ver(crs, com, m, dec)$*.*

The scheme must satisfy the following requirements:

1. ***Correctness:*** *Ver always accepts in an honest execution, i.e, for any string m and any security parameter λ,*

$$\Pr_{\substack{crs\leftarrow Gen(1^\lambda)\\(com,dec)\leftarrow Com(crs,m)}}\left[Ver(crs,com,m,dec)=1\right]=1.$$

2. ***Hiding****: For any two strings $m_1, m_2 \in \{0,1\}^*$ and any common reference string crs, the distribution of the commitment of m_1 and m_2 are computationally indistinguishable, i.e, if we denote by Com_c only the commitment part of Com then: $\{Com_c(crs,m_1)\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \{Com_c(1^\lambda,crs,m_2)\}_{\lambda\in N}$.*
3. ***Binding****: For every $\lambda \in \mathbb{N}$, with probability at least $1-2^{-\lambda}$ over the common reference string, any commitment com^* has at most one value m that can be accepted by Ver, i.e,*

$$\Pr_{crs\leftarrow Gen(1^\lambda)}\left[\exists m_1,m_2,dec_1,dec_2\in\{0,1\}^*:\ \begin{array}{c} m_0\neq m_1,\\ Ver(crs,com^*,m_1,dec_1)=1,\\ Ver(crs,com^*,m_2,dec_2)=1\end{array}\right]<2^{-\lambda}.$$

Theorem 19 ([Nao91, HILL99]). *Assuming the existence of a one-way function, there exists a commitment scheme in the CRS model. Furthermore, the commitment scheme only makes a black-box use of the one-way function.*

Fact 20 *Let D a distribution over strings of length λ, f a function and com a commitment scheme. Then $(D, com(f(D)))$ and $(D, com(0^\lambda))$ are computationally indistinguishable.*

2.6 Multi-Party Computation

We consider the following multi party computation model: n parties wish to evaluate a function defined jointly on their n private inputs. While there are many variations of this model, we focus on the one where the output of all of the parties should be the same (aka "secure function evaluation"). The communication between parties is synchronous and all pairwise communication channels are secure. Additionally, following [IKOS09],we also allow an OT-channel between every two parties. In each round, each party can perform local computations on all its view (input and all messages seen up to that round), send messages to any other party and read all its incoming messages. A protocol in this setting, is a specification for each of the n parties.

For this setting we define the notion of privacy and robustness as given by [IKOS09]:

Definition 21 (Correctness). *Given a deterministic n-party functionality $f(w_1,...,w_n)$ (where input w_i belongs to party i), we say that Π realizes f with perfect correctness if for all inputs $w_1,\ldots,w_n$, the probability that the output of some party is different from the output of f is 0, where the probability is over the randomness of all of the parties.*

Definition 22 (t-Privacy). *Let $1 \leq t < n$. We say that Π realizes f with perfect t-privacy if there exists a PPT simulator Sim such that for any inputs $w_1, \ldots, w_n$, and every set of corrupted parties $T \subset [n]$, where $|T| \leq t$, the joint views of parties in T (which includes their inputs, randomness and received messages) is distributed identically to $Sim(T, (w_i)_{i \in T}, f(w_1, \ldots, w_n))$.*

We will rely on the classical construction of a secure MPC protocol against $t \leq n-1$ corruptions, which has perfect semi-honest security in the OT-hybrid model, due to Goldreich, Micali and Wigderson [GMW87].

Theorem 23 ([GMW87]). *For any n-input functionality f, computable by a circuit of size S, there is an n-party protocol in the OT-hybrid model with perfect correctness and perfect $(n-1)$-privacy. The parties run in time* $\mathrm{poly}(S, n)$.

3 Succinct Zero-Knowledge Proofs

In this section we prove our main results: zero-knowledge proofs for any NP relation that can be verified either in bounded space or by a bounded depth circuits. We start with a technical definition, which, for any NP relation R gives a related relation R' with a shorter witness (but while increasing the length of the input and complexity of verifying the relation). For an NP relation R, recall that we use n to denote the input length, m to denote the witness length and λ to denote the security parameter.

Definition 24. *Let R be an* NP *relation, and G be a PRG, then we define the* NP *relation* $R'_G \triangleq \left\{ \left((x, u), s \right) : \left(x, G(s) \oplus u \right) \in R \right\}$.

Our main technical lemma shows how to covert a zero-knowledge proof for R'_G to one for R (where we benefit if the protocol for R'_G mainly depends on the witness length).

Lemma 25. *Let R be an* NP *relation with input size n and witness size m, G be a PRG, and λ a security parameter. If R'_G has a zero-knowledge proof with communication complexity $cc(m, n, \lambda)$ and soundness error ε, then R has a zero-knowledge proof with communication complexity $m + cc(m, n, \lambda)$ and soundness error ε.*

Before proving Lemma 25, we first show how to use it to derive our main results.

Deriving Theorems 1 and 2 from Lemma 25. Let R be an NP relation by depth $D = D(n)$ polynomial-sized circuits. Assuming the existence of one-way function, by Theorem 16, there exists a PRG $G : \{0,1\}^\lambda \to \{0,1\}^m$ computable by depth $\log(m) \cdot \mathrm{poly}(\lambda)$ and size $\mathrm{poly}(m, \lambda)$ circuits. This implies that the relation R'_G can be decided by a depth $D + \log(m) \cdot \mathrm{poly}(\lambda)$ and size $\mathrm{poly}(n, m, \lambda)$ circuit. By Theorem 12 (and once again using the assumption that one-way functions exist), we have that R'_G has a zero-knowledge proof with communication complexity

$cc(m, n, \lambda) = \text{poly}(\log(m), \lambda, D)$, and soundness error $2^{-\lambda}$ (and a polynomial-time prover and verifier). Theorem 1 now follows directly from Lemma 25.

Theorem 2 follows similarly, by combining the small space PRG of Theorem 17 with the succinct zero-knowledge proof for bounded space computations of Theorem 14.

3.1 Proof of Lemma 25

Let R be an NP relation. The zero-knowledge proof for R, which establishes Lemma 25, is presented in Fig. 1.

Common Input: $x \in \{0,1\}^n$ and security parameter 1^λ.
Prover's Additional Input: witness $w \in \{0,1\}^m$, such that $(x, w) \in R$.

The Protocol:

1. P generates a random PRG seed $s \in \{0,1\}^\lambda$.
2. P sends $u = G(s) \oplus w$ to V.
3. P and V emulate (P', V') with (x, u) as the common input, s as the witness and wrt security parameter λ.

Fig. 1. Succinct Zero-Knowledge Proof for NP Relation R

Let R'_G be the related NP relation (see Definition 24) and assume that (P', V') is a zero-knowledge proof for R'_G.

Completeness. Let $(x, w) \in R$. For any $s \in \{0,1\}^\lambda$, by construction, it holds that $\big((x, u), s\big) \in R'_G$, where $u = G(s) \oplus w$. Thus, the protocol (P', V') is run on a YES instance. Perfect completeness now follows immediately from the perfect completeness of (P', V').

Soundness. Let $x \notin L_R$ and let P^* be a cheating prover strategy. Without loss of generality we assume that P^* is deterministic. We denote P^*'s first message in the protocol by u^*. Assume toward a contradiction that $(x, u^*) \in L'_R$. By definition, there exists an s s.t. $(x, u^* \oplus G(s)) \in R$, but that contradicts our assumption that $x \notin L_R$. Therefore $(x, u^*) \notin L'_R$ and so, the protocol (P', V') is run on a NO instance. By the soundness of the latter protocol, the verifier accepts with probability at most ε.

Complexity. The prover sends u to V, where $u = G(s) \oplus w$ so $|u| = |w|$. Then, the parties emulate (P', V') to prove that $((x, u), s) \in R'$. The communication complexity for R'_G from the given zero-knowledge protocol is $cc(m, n, \lambda)$. Thus, overall we get communication complexity $|w| + cc(m, n, \lambda)$. In addition, assuming P' and V' are polynomial-time, then so are P and V.

Computational Zero-knowledge. Computational zero-knowledge follows from the computational zero-knowledge of (P', V') and the pseudorandomness of G, details follow.

Given a malicious verifier $\hat{V}$ we show how to simulate its view. We note that after P sends the first message, the parties run the zero-knowledge protocol for R'_G, hence we can view the behavior of $\hat{V}$ from that point on as the behavior of a malicious verifier in the (P', V') protocol. We denote this residual cheating verifier behavior by $\hat{V}'$. Since (P', V') is zero-knowledge, $\hat{V}'$ has a simulator S' that can simulate its view. We use S' to construct an S for $\hat{V}$:

$\underline{S(x, 1^\lambda)}$:

1. Choose $u^* \in \{0,1\}^m$.
2. Run S' on input $\big((x, u^*), 1^\lambda\big)$ and output $(x, u^*, S'(x, u^*))$.

Claim. For every ensemble $(x, w) \in R$ it holds that $\left\{View_{\hat{V}}^{P(w)}(x, 1^\lambda)\right\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \left\{S(x, 1^\lambda)\right\}_{\lambda\in\mathbb{N}}$.

Proof. The proof is via a hybrid argument. Define the following hybrid distributions (to avoid cluttering the notation we omit the 1^λ from all distributions):

$$H_0 := \Big(x, u, View_{\hat{V}'}\big(x, u)\big)\Big),$$
$$H_1 := \Big(x, u, S'\big(x, u)\big)\Big),$$
$$H_2 := \Big(x, u^*, S'\big(x, u^*\big)\Big),$$

where $s \in_R \{0,1\}^\lambda$ is a random seed, $u = G(s) \oplus w$ and $u^* \in_R \{0,1\}^m$. Note that $H_0 = View_{\hat{V}}^{P(w)}(x)$ and that $H_2 = S(x)$ and so it suffices to show that H_0 and H_2 are both computationally indistinguishable from H_1.

$H_0 \overset{c}{\approx} H_1$*:* Assume towards a contradiction that the distributions are computationally distinguishable. Then, there exists a distinguisher D that distinguishes between H_0 and H_1 with non-negligible advantage δ. By an averaging argument, there is some $s = (s_\lambda)_{\lambda\in\mathbb{N}}$ such that D has a distinguishing δ advantage conditioned on choosing s as the PRG seed. We hardwire this choice of s into the distinguisher D as non-uniform advice and denote the resulting distinguisher by D_s. We use D_s to build a distinguisher D' between $View_{\hat{V}'}((x, u))$ and $S'(x, u)$ (recall that $u = G(s) \oplus w$ with the aforementioned s) in contradiction to the zero-knowledge property of (P', V'). Since (x, u) is the input of the protocol (P', V') they already exists in the view of $\hat{V}'$, so D' will take them from the view, concatenate everything to $(x, u, View_{\hat{V}'(z)}(x, u))$, then use D and achieve the same distinguishing probability δ.

$H_1 \overset{c}{\approx} H_2$*:* Assume towards a contradiction that there exists a non-uniform distinguisher D that distinguishes between the hybrids for some (x, w). We build D' that distinguishes between U_n and $G(U_{|\lambda|})$. We give D' the non-uniform advice (x, w). Given as input $r \in \{0,1\}^m$, the distinguisher D' runs D on input $(x, w \oplus r, S'(x, w \oplus r))$ and outputs the result. If r is sampled from U_n, then $w \oplus r$ will also be a random element of U_n and thus $(x, w \oplus r, S'(x, w \oplus r))$ will be of the same distribution as $(x, u^*, S'(x, u^*))$. On the other hand, if r is sampled from $G(U_{|\lambda|})$ then $(x, w \oplus r, S'(x, w \oplus r))$ is the same distribution as $(x, u, S'(x, u))$. So D' will be able to distinguish with the same probability as D, in contradiction to the pseudorandomness of G.

4 Zero-Knowledge with Black-Box use of the OWF

Recall that the proof of Theorems 1 and 2 relies on a protocol in which the prover and verifier make a non black-box use of the one-way function (see Lemma 25 for details). In this section, we prove Theorem 5 which gives a different construction that only makes black-box use of the one-way function. A caveat however is that here we only achieve an inverse polynomial soundness error, whereas Theorem 1 and Theorem 2 had an exponentially small error.

As mentioned in the introduction, the proof of Theorem 5 is inspired by, and improves upon the aforementioned work of Hazay, Venkitasubramaniam and Weiss [HVW23]. Similarly to their work, we utilize the MPC-in-the-head [IKOS09] techniques in order to avoid the non black-box use of the one way function.

4.1 Proof of Theorem 5

Let R be an NP relation. We denote by n the instance length and m the witness length, we denote with $S = S(n)$ the size of the verification circuit and $D = D(n)$ its depth. Let λ be a security parameter and ε the desired soundness error. To construct the protocol establishing Theorem 5, we will use the following ingredients, all of which either exist unconditionally or can be constructed (via a fully black-box construction) from a one-way function:

- A pseudorandom generator (PRG) $G : \{0,1\}^\lambda \to \{0,1\}^m$ (e.g., the one from Theorem 17, but any PRG with a similar stretch would do – the depth bound is not needed) with security parameter $\lambda' = \lambda + \log(3/\varepsilon)$
- The non-interactive statistically binding CRS commitment scheme from Theorem 19, which we denote by *com*.
- The interactive protocol from Theorem 10, denoted $(P_{GKR}, V_{interactive}, V_{post})$. We denote the number or rounds in the interactive part by $r = O(D \cdot \log(S))$. We choose a field with characteristic 2 and size $\Theta\left(\frac{D \cdot \log(S)}{\varepsilon}\right)$, where all operations below will be done over this field.
- An MPC protocol from Theorem 23 with perfect security and $(k-1)$-privacy with k parties, where $k = \Theta(1/\varepsilon)$.

Using these components, the zero-knowledge proof for R that establishes Theorem 5 is presented in Fig. 2.

We proceed to show that the protocol satisfies the desired properties.

Complexity. First, the verifier sends to P a reference string of size $\text{poly}(\lambda)$. Then, P sends to V commitments to all random seeds and w_{com}. Each commitment is of size $\text{poly}(\lambda)$, and $|w_{com}| = m$. The prover and verifier run the interactive phase that has communication complexity $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$, and since it is secret shared among k parties, we have communication $k \cdot \text{poly}(D, \log(S), \log(|\mathbb{F}|))$.

The prover P then sends commitments to the views of all k parties in the MPC. The MPC's input for each party consists of the multi-linear extension at point z, which has size $\log(\mathbb{F})$, and b_j (where $b_j = (m_{i,j})_{i\in[r]}$, the messages from Step Item 4), each of size $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$. The size of the circuit computed by the MPC is $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$, as derived from the complexity of V_{post} in Theorem 10.

By Theorem 23, the parties run in time polynomial in the size of the input and the circuit, so the size of the view is at most $\text{poly}(k, D, \log(S), \log(\mathbb{F}))$. The size of the commitment to the view is there $\text{poly}(\lambda, k, D, \log(S), \log(|\mathbb{F}|))$. Finally, P sends the $O(k)$ decommitments of size $\text{poly}(\lambda)$ each.

Overall, the communication complexity is therefore $m + \text{poly}(\lambda, k, D, \log(S), \log(|\mathbb{F}|))$. The communication complexity stated in the theorem statement now follows by taking $k = \Theta(\frac{1}{\varepsilon})$ and $\mathbb{F}$ as mentioned above. In addition since the commitment, PRG, GKR protocol and the MPC protocol are computable in polynomial-time, then so are P and V.

Completeness. Let $(x, w) \in R$. If (P, V) follow the protocol specification, the input of V_{post} in Step 5c is: $\left(coins_V, \left(\bigoplus_{j\in[k]} b_j\right)\right)$, where by construction $\bigoplus_{j\in[k]} b_j = \left(m_i\right)_{i\in[r]}$. Hence, $\left(coins_V, \left(\bigoplus_{j\in[k]} b_j\right)\right)$ is the transcript of the interaction between $(P_{GKR}, V_{interactive})$. Also note that $\langle C\rangle$ is the circuit "hash" for the GKR protocol. Lastly, $\bigoplus_{j\in[k]} a_j = \left(\bigoplus_{j\in[k-1]} \widehat{G(s_j)}[z]\right) \oplus \widehat{w_s}[z] = \widehat{w}[z]$, where the last equality stems from the fact that the low degree extension is a linear function and the addition is done bit-wise over a field of characteristic two.

Thus, V_{post}'s input in Step 5c is a valid run of the GKR protocol and by its perfect completeness, the verifier will accepts. Hence, from the (perfect) completeness of the MPC protocol, the run of the MPC protocol will be an accepting one. Since P behaved according to the protocol, P should be able to open all the commitments correctly, and all checks in Step 8 will pass and therefore V accepts.

Soundness. Let $x \notin L_R$ and let P^* be a cheating prover strategy. Without loss of generality we assume that P^* is deterministic.

Common Input: $x \in \{0,1\}^n$ and security parameter 1^λ.
Prover's Additional Input: witness $w \in \{0,1\}^m$, such that $(x, w) \in R$.

The Protocol:

1. V generates a reference string for the commitment scheme and sends it to P using security parameter $\lambda' = \lambda + \log(3/\varepsilon)$. All commitments in the protocol are done using the commitment scheme *com* with respect to this reference string, which we omit to avoid cluttering the notation.
2. P generates k random PRG seeds $s_1, \ldots, s_k \in \{0,1\}^\lambda$.
3. P sends commitments $\{com(s_i)\}_{i\in[k]}$ to all the seeds. In addition, it sends $w_{com} = w_s \oplus G(s_k)$, where $w_s = (w \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1}))$. (The pair $(w_{com}, com(s_k))$ should be interpreted as a commitment to w_s).
4. P and V emulate the interactive phase of the GKR protocol (see Theorem 10) on input (C_x, w) (where C_x denotes the circuit that computes the relation R with x hardcoded). However, in every round $i \in [r]$, whenever P_{GKR} wants to send a message m_i, the prover does not forward the message directly, but rather generates an additive secret sharing of the message s.t $m_{i,1} \oplus \cdots \oplus m_{i,k} = m_i$, and sends to V commitments to $m_{i,1}, \ldots, m_{i,k}$.
   (We denote the coins sent from V to P in this stage as $coins_V$).
5. (a) P derives $z \in \mathbb{F}^d$ and $\langle C \rangle$ (He can do it from Item 4 as explained in Theorem 10).
   (b) P computes the multi-linear extension of $G(s_1), \ldots, G(s_{k-1})$ and w_s at the point z. That is, for every $j \in [k-1]$, it computes $a_j = \widehat{G(s_j)}[z]$, and additionally computes $a_k = \widehat{w_s}[z]$ (see Section 2.2 for details).
   (c) P executes ("in its head") the k-party MPC protocol with the following inputs. For party $j \in [k-1]$, the input is $input_j = (a_j, b_j)$, where $b_j := (m_{i,j})_{i\in[r]}$. For the last party $input_k = (a_k, b_k)$, where $b_k := (m_{i,k})_{i\in[r]}$. The MPC is executed relative to the functionality $V_{post}\Big(\big(coins_V, \bigoplus_{j\in[k]} (b_j)\big), \langle C \rangle, \bigoplus_{j\in[k]} a_j\Big)$. Where $b_j = (m_{i,j})_{i\in[r]}$.
   (d) P sends commitments to the views of the k parties in the MPC protocol.
6. V randomly chooses a subset of size $t = k - 1$ of the parties, denoted by $T_q = [k] \backslash \{q\}$, and sends it to P.
7. For every $j \in T_q$, the prover P decommits to everything related to j, namely s_j, $(m_{i,j})_{i\in[r]}$, and the view of party j.
8. V verifies that (1) all inputs of the parties in T_q were computed correctly, (2) all their views are consistent (3) all parties properly followed the specification of the MPC protocol (4) all of the parties accepted. If all tests pass then V accepts, otherwise it rejects.

Fig. 2. Succinct Zero-Knowledge Proof for NP Relation R

By Definition 18, since we used security parameter $\lambda' \geq \log(3/\varepsilon)$, with probability at least $1 - \varepsilon/3$ the reference string generated in Step 1 produces a perfectly binding commitment. We continue the analysis under the assumption that the CRS is indeed perfectly binding, while noting that this can only increase the soundness error by $\varepsilon/3$.

Consider the following possible behaviors of P^*:

1. It produces an invalid decommitment.
2. The behavior of one of the parties in the MPC protocol transcript, that is defined by the commitment (since they are perfectly binding), does not follow the protocol specification.
3. A pair of views is inconsistent (i.e., messages sent by one party are not received correctly by the other parties).

In the first case the verifier when checking the decommitments. In the second case, with probability $1 - \frac{1}{k}$, the verifier V will choose the relevant party and reject. In the third case, with probability at least $1 - \frac{2}{k}$ the relevant pair of parties is selected and the verifier rejects.

Additionally, if P calculates $\langle C \rangle$ incorrectly then this either does not change the outcome of the MPC or it changes (at least) one of the parties' behavior, or creates an inconsistency between the views of two parties, then once again with probability $1 - \frac{2}{k}$, V will choose the relevant party/parties and reject. Thus, we can continue the analysis assuming the MPC protocol computes the intended function on the defined inputs while adding at most $\frac{2}{k}$ to the soundness error.

Assuming all commitments can be opened and in one way, and assuming P simulates the MPC protocol correctly and on the inputs derived from the opening of the commitments as defined in the protocol, then from the perfect correctness of the MPC protocol, we get that the MPC protocol calculates the output of $V_{post}(coins_V, \{m_i\}_{i\in[r]}), \langle C \rangle, \hat{w}_*[z]$. For some w_* (derived from the unique de-commitment and recombining of the messages in Item 5b), and $(coins_V, \{m_i\}_{i\in[r]}), \langle C \rangle$ derived of a possible run of the GKR protocol. Since $x \notin L_R$, for any such w_* it holds that $(x, w_*) \notin R$ and so the circuit C_x does not accept w_*. By fixing the field size to be $\Theta(\frac{D \cdot \text{polylog}(S)}{\varepsilon})$ for a sufficiently large constant in the Θ-notation, by Theorem 10, the GKR protocol has a soundness error of $\varepsilon/3$.

Overall, the probability that V accepts is at most $\frac{\varepsilon}{3} + \frac{2}{k} + \frac{\varepsilon}{3}$. By choosing $k := \lceil \frac{6}{\varepsilon} \rceil$ we can get the desired soundness error of ε.

Computational Zero-knowledge. Let V^* be a malicious verifier, which we assume without loss of generality to be deterministic. For a given input $(x, w) \in R$, we denote V^*'s first message on input x (which should specify a reference string for the commitment) by ref. By Definition 18, the commitment is hiding when using *any* reference string, in particular ref. For clarity of notation we therefore omit ref below, but note that all commitments are done relative to this fixed reference string.

Our proof of zero-knowledge follows the outline of the textbook proof of zero-knowledge of the 3-coloring protocol [Gol01, Section 4.4.2.3]. In particular, we

will present a simulator S that is allowed to output a special abort symbol $\perp$ and analyze it via two key propositions:

- In Proposition 26 we show that the probability that $S(x)$ outputs $\perp$ is at most $1 - O(\frac{1}{k})$.
- In Proposition 27 we show that conditioned on not outputting $\perp$, the output of $S(x)$ is computationally indistinguishable from the verifier's view in a real execution of the protocol.

These two properties, combined with rejection sampling, yield the desired simulator (see [Gol01, Definition 4.3.2] for details). The base simulator (which is allowed to abort) is presented in Fig. 3.

Proposition 26. *The probability that S outputs $\perp$ is at most $1 - O(1/k)$.*

Proof. Recall that V^* is deterministic. We assume without loss of generality that V^* always specifies a valid set T_{q^*} (i.e $T_{q^*} \subseteq [k]$ is a subset of size $k-1$) in Step 6 (since otherwise we can just interpret its message as some fixed T_{q^*}).

We view two strings $m = (s, \alpha, \beta), m' = (s', \alpha', \beta')$, where s, s' represent some choice of seeds for the PRG (in Step 3), α, α' two randomness choices for the secret sharing (in Step 4) and β, β' two randomness choices for the MPC simulator (in Step 5d). These randomness choices, together with a choice of $q \in [k]$ (in Step 5c) and randomness of the commitments, define all the randomness of the simulator.

Denote by $\Pr[V^*_{q^*}(m, q)]$ the probability, taken over the randomness only of the commitment, that the verifier V^* requests q^* given simulator behavior corresponding to randomness (m, q). For any different choices (m, q) and (m', q'), due to the hiding property of the commitment, the difference between the probabilities of the verifier making the choice q^* for these two interactions is negligible (otherwise there exist two distinct messages that we can distinguish between using V^*).

Thus, for every polynomial p_1 and every choice q^* it holds that $|\Pr[V^*_{q^*}(m, q)] - \Pr[V^*_{q^*}(m', q')]| < \frac{1}{p_1(n)}$. Using this inequality, we prove the claim.

The simple idea is that if the choice of the verifier for T_{q^*} is made regardless (up to negligible probability) of the messages sent by the prover, then with probability close to $\frac{1}{k}$ the verifier will choose the same set and there will be no abort. The rigorous proof that follows is with elementary manipulations over the probabilities.

The Simulator for V^*

Input: main input $x \in L$ and security parameter 1^λ.

1. The simulator S starts emulating V^* on input x and obtain in response a reference string ref. All commitments in the protocol are done using this reference string, which we omit similarly to the protocol.
2. S chooses $\tilde{w} \in \{0,1\}^m$ and generates k random PRG seeds $s_1, \ldots, s_k \in \{0,1\}^\lambda$.
3. S sends to V^* commitments to all random seeds $\{s_i\}_{i\in[k]}$. In addition, it sends $\tilde{w}_{com} = \tilde{w}_s \oplus G(s_k)$, where $\tilde{w}_s = (\tilde{w} \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1}))$.
4. S emulates with V^* the interactive phase of the GKR protocol as follows: In every round $i \in [r]$, the simulator S randomly chooses $\tilde{m}_i$, and secret shares the message to k shares s.t $\tilde{m}_{i,1} \oplus \cdots \oplus \tilde{m}_{i,k} = \tilde{m}_i$, and sends to V^* commitments to $\tilde{m}_{i,1} \ldots \tilde{m}_{i,k}$. We denote the coins sent from V to P in this stage by $coins_{V^*}$, and the entire interaction in this stage (the commitments to all shares as well as the verifier's coins) by $\widetilde{\mathsf{tr}_{\mathsf{GKR}}}$.
5. (a) Based on the interaction, S computes $z \in \mathbb{F}^d$ and $\langle C \rangle$ as described in Theorem 10.
   (b) S computes the multi-linear extension of $G(s_1), \ldots, G(s_{k-1})$ and $\tilde{w}_s$ at the point z. That is, for every $j \in [k-1]$, it computes $a_j = \widehat{G(s_j)}[z]$, and additionally computes $a_k = \widehat{\tilde{w}_s}[z]$.
   (c) S chooses a random subset $T_q = [k] \backslash \{q\}$ of $k-1$ parties.
   (d) S computes the inputs for the selected parties as in the protocol, and runs the MPC simulator denoted S_{MPC} on the selected parties' inputs. The MPC simulation is executed wrt the function $V_{post}\Big(\big(coins_{V^*}, \bigoplus_{j\in[k]} b_j\big), \langle C \rangle, \bigoplus_{j\in[k]} a_j\Big)$, where $b_j = (\tilde{m}_{i,j})_{i\in[r]}$. We denote S_{MPC}'s output for party $i \in T_q$ by $\widetilde{view}_i$.
   (e) S sets the view of the remaining party q, to a default value, $\widetilde{view}_q = 0^{|view|}$, and sends to V^* commitments to the $k-1$ views generated by S_{MPC} and the view of the remainder party q and all communication channels. Denote these commitments by $\widetilde{com(view_i)}_{i\in[k]}$.
6. V^* responds with a set T_{q*}.
7. If $T_q \neq T_{q*}$, then S outputs $\perp$ and terminates. Otherwise, S outputs $\big(x, ref, \bar{c}, \tilde{w}_{com}, \widetilde{\mathsf{tr}_{\mathsf{GKR}}}, \{\widetilde{com(view_i)}\}_{i\in[k]}, T_q, \{dec_S(i), dec_m(i), dec_v(i)s\}_{i\in T}\big)$, where (1) $\bar{c} = com(s_1), \ldots, com(s_k)$, (2) $dec_s(i)$, $dec_m(i)$, $dec_v(i)$ respectively the decommitments to s_i, $\tilde{m}_i$ and party i's view.

Fig. 3. Zero-Knowledge Simulator

We note that all randomness choices are independent, hence using elementary manipulations:

$$\begin{aligned}
\Pr[S=\perp] &= \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta,q}\left[\sum_{q^*\neq q}\Pr\left[V^*_{q^*}(m_{\bar{s},\alpha,\beta},q)\right]\right] \\
&\leq \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta,q}\left[\sum_{q^*\neq q}\left(\Pr[V^*_{q^*}(\bar{0},0)]+\frac{1}{2k^2}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta}\left[\mathop{\mathbb{E}}_{q}\left[\sum_{q^*\neq q}\Pr[V^*_{q^*}(\bar{0},0)]\right]\right]+\frac{1}{2k} \\
&= \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta}\left[\mathop{\mathbb{E}}_{q^*}\left[\sum_{q\neq q^*}\Pr\left[V^*_{q^*}(\bar{0},0)\right]\right]\right]+\frac{1}{2k} \\
&= \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta}\left[(k-1)\mathop{\mathbb{E}}_{q^*}\left[\Pr[V^*_{q^*}(\bar{0},0)]\right]\right]+\frac{1}{2k} \\
&= \mathop{\mathbb{E}}_{\bar{s},\alpha,\beta}\left[\frac{(k-1)}{k}\right]+\frac{1}{2k} \\
&= 1-\frac{1}{2k},
\end{aligned}$$

and the proposition follows.

Denote by $\bar{S}(x)$ the distribution of $S(x)$ conditioned on $S(x)\neq\perp$ (i.e., conditioned on $T_q = T_{q^*}$).

Proposition 27. *The ensembles $\bar{S}(x)$ and $\{View^{P(w)}_{V_*}(x,\lambda)\}_{x\in L}$ are computationally indistinguishable.*

Proof. Let $T_q \subseteq [k]$ denote the set of parties that the verifier selects, both with respect to the simulator and the prover (note that T_q depends on the previous messages that the prover/simulator sent). For $x \in L$, both $\bar{S}(x)$ and $View^{P(w)}_{V_*}(x,\lambda)_{x\in L}$ are sequences of the following form:

$$\left(x, ref, \bar{c}, w_{com}, \mathsf{tr}_{\mathsf{GKR}}, \left\{com(view_i)\right\}_{i\in[k]}, T_q, \left\{dec_S(i), dec_m(i), dec_v(i)\right\}_{i\in T}\right).$$

Since we fixed (x,w) and ref we omit them from the notation when analyzing these two distributions.

We define for each subset $T_q = [k]\backslash q$ two random variables describing, respectively, the output of $\bar{S}$ and the view of V^* in a real interaction, in the case that the verifier's request equals T_q:

1. Let $\mu_q(x)$ denote the output of $\bar{S}(x)$ conditioned on having the verifier's request in Step 6 equal T_q when interacting with V^*. Let $p_q(x)$ denote the probability that the verifier requests the set T_q when interacting with $\bar{S}(x)$.

2. Let $\nu_q(x)$ denote $View^{P(w)}_{V^*}(x)$ conditioned on $View^{P(w)}_{V^*}(x)$ having the verifier's request in Step 6 in the protocol (i.e., when interacting with $P(x,w)$) equal T_q. Let $f_q(x)$ denote the probability that V^* selects T_q when interacting with $P(x,w)$ in the protocol.

Assume toward a contradiction that the two ensembles in the statement of the claim are distinguishible. Then one of the following cases must occur.

Case 1: There exists $q \in [k]$ such that $|p_q - f_q|$ is non-negligible. To show that Case 1 leads to a contradiction, we first argue that the part of the interaction up to the opening of the commitment is computationally indistinguishible. This is is captured by the following claim.

Claim. It holds that $(\bar{c}, \tilde{w}_{com}, \widetilde{\mathsf{tr}_{\mathsf{GKR}}}, com(\widetilde{view}_i)_{i\in[k]}) \overset{c}{\approx} (\bar{c}, w_{com}, \mathsf{tr}_{\mathsf{GKR}}, com(view_i)_{i\in[k]})$, where $\bar{c} = (com(s_1), \ldots, com(s_k))$.

Section 4.1 follows in a straightforward manner from hiding property of the commitment scheme and so we defer its proof to Section 4.1.1.

Assuming Case 1 occurs, we can build a (non-uniform) distinguisher between the two distributions by feeding V^* with the distribution and seeing whether it outputs q, the distinguisher output 1 if q was chosen and 0 otherwise thus case 1 leads to contradiction.

Case 2: If we are not in Case 1, then, loosely speaking, for every q it holds that

$$|p_q(x) - f_q(x)| \leq \frac{1}{\text{poly}(|x|)}.$$

Since we assumed the ensembles are distinguisible, by an averaging argument, there exists some $q \in [k]$ for which p_q and f_q are close and yet the distinguisher is able to distinguish even conditioned on this value of q. Formally, there exists a probabilistic polynomial-time algorithm A, a polynomial $p(\cdot)$, and an infinite sequence of integers such that for each integer n (in the sequence) there exists an $x, |x| = n$ and a set of parties T_q such that the following conditions hold[9]:

1. $f_q(x) > \frac{1}{2\cdot p(n)}$,
2. $|p_q(x) - f_q(x)| < \frac{1}{8\cdot p(n)^2}$,
3. $|\Pr[A(\mu_q(x)) = 1] - \Pr[A(\nu_q(x)) = 1]| > \frac{1}{2\cdot p(n)}$.

We proceed to show that Case 2 leads to a contradiction to the following claim,

[9] The conditions follows from the fact that A distinguishes the two distributions and that Case 1 does not hold. From an averaging argument, there exist a player $q \in [k]$ s.t $\left|f_q(x)\cdot \Pr[A(\mu_q(x))] - p_q(x)\cdot \Pr[A(\nu_q(x))]\right| \geq \frac{1}{p(n)}$. Now for Item 2 we use the fact that Case 1 does not hold, using a suitably large polynomial. Now we conclude that $\left|f_q(x)\cdot \Pr[A(\mu_q(x))] - f_q(x)\cdot \Pr[A(\nu_q(x))]\right| \geq \frac{1}{2\cdot p(n)}$ and Items 1 and 3 follow.

Claim. Let $T_q = [k] \backslash \{q\}$ be a fixed set of parties, denote:

$$H_0 = \left(\bar{c}, \tilde{w}_{com}, \widetilde{\mathsf{tr}_{\mathsf{GKR}}}, \left\{com(\widetilde{view}_i)\right\}_{i\in[T_q]}, com(\widetilde{view}_q), T_q, \left\{dec_S(i), dec_m(i), dec_v(i)\right\}_{i\in T_q}\right)$$
$$H_1 = \left(\bar{c}, w_{com}, \mathsf{tr}_{\mathsf{GKR}}, \left\{com(view_i)\right\}_{i\in[T_q]}, com(view_q), T_q, \left\{dec_S(i), dec_m(i), dec_v(i)\right\}_{i\in T_q}\right)$$

then, $H_0 \overset{c}{\approx} H_1$.

We yet again defer the proof of the claim to Section 4.1.2 and proceed directly to showing why it leads to a contradiction. Namely, we use A to construct a distinguisher A' that distinguishes between H_0 and H_1 thereby contradicting Section 4.1. Consider A' that emulates the simulator and checks if T_q was chosen by V^*. If so A' runs A on its input (which is sampled either from H_0 or H_1). Otherwise A' outputs 0.

We proceed to show that A' indeed distinguishes between these two distributions:

$$\begin{aligned}
\left|\Pr[A'(H_0)] - \Pr[A'(H_1)]\right| &= \left|f_q(x)\cdot\Pr[A(\mu_q(x))] - p_q(x)\cdot\Pr[A(\nu_q(x))]\right| \\
&\geq f_q(x)\cdot\left|\Pr[A(\mu_q(x))] - \Pr[A(\nu_q(x))]\right| - \Pr[A(\nu_q(x))]\cdot\left|p_q(x) - f_q(x)\right| \\
&\geq f_q(x)\cdot\left|\Pr[A(\mu_q(x))] - \Pr[A(\nu_q(x))]\right| - \left|p_q(x) - f_q(x)\right| \\
&> \frac{1}{2\cdot p(n)}\cdot\frac{1}{2\cdot p(n)} - \frac{1}{8\cdot p(n)^2} \\
&= \frac{1}{8\cdot p(n)^2},
\end{aligned}$$

where the first inequality follows from the (reverse) triangle inequality and the third inequality from the above distance bound on p_q vs. f_q. Thus, A' distinguishes between H_0 and H_1 with non-negligible probability, in contradiction to Section 4.1.

This concludes the proof of Proposition 27.

4.1.1 Proof of Section 4.1

The proof is via a hybrid argument. Define:

$$H_0 := \left(\bar{c}, \tilde{w}_{com}, \widetilde{\mathsf{tr}_{\mathsf{GKR}}}, com(\widetilde{view}_i)_{i\in[k]})\right),$$
$$H_1 := \left(\bar{c}, w_{com}, \widetilde{\mathsf{tr}_{\mathsf{GKR}}}, com(\widetilde{view}_i)_{i\in[k]}\right),$$
$$H_2 := \left(\bar{c}, w_{com}, \mathsf{tr}_{\mathsf{GKR}}, com(view_i)_{i\in[k]}\right).$$

We show that H_0 and H_2 are both indistinguishable from H_1, from which the claim follows.

$H_0 \overset{c}{\approx} H_1$*:* Assume towards a contradiction that the distributions are computationally distinguishable. Then, since the only difference between the hybrids lies in $\tilde{w}_{com}$ versus w_{com}, there exists a distinguisher D that distinguishes between $(\tilde{w}_{com})$ and w_{com}. (recall $w_{com} = w_s \oplus G(s_k)$, where $w_s = (w \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1}))$, and $\tilde{w}_{com} = \tilde{w}_s \oplus G(s_k)$, where $\tilde{w}_s = (\tilde{w} \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1}))$. We construct D' that distinguishes between U_n and $G(U_{|\lambda|})$. We give D' the non-uniform advice (x, w).

The distinguisher D', given an input $r \in \{0,1\}^m$, chooses $s_1, \ldots, s_{k-1}$, and runs D on the input $\hat{w}_{com} = r \oplus (w \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1}))$ and outputs the result. If r is sampled from U_n, then $\hat{w}_{com}$ will also be a random element of U_n and thus $\hat{w}_{com}$ will be of the same distribution as $\tilde{w}_{com}$. On the other hand, if r is sampled from $G(U_{|\lambda|})$ then $\hat{w}_{com}$ is the same distribution as w_{com}. So D' will be able to distinguish with the same probability as D, in contradiction to the pseudorandomness of G.

$H_1 \overset{c}{\approx} H_2$*:* Due to the hiding property of the commitment, for any two strings $m_1, m_2 \in \{0,1\}^*$ and any common reference string ref, the distribution of the commitment of m_1 and m_2 are computationally indistinguishable. Thus the commitments of the simulator are computationally indistinguishable from the commitments to in the real interation.

4.1.2 Proof of Section 4.1

We first show that the inputs and views of the parties selected in the set T_q are computationally indistinguishable in the two cases: that is, when V^* interacts with P vs. its inteaction with S.

For simplicity of notation we will assume without loss of generality that $q = 1$ and we use T to denote the selected set $T = \{2, \ldots, k\}$. Thus, we need to show that

$$\left((s_i)_{i\in T}, \tilde{w}_s, (\tilde{m}_i)_{i\in T}, (\widetilde{view}_i)_{i\in[T]}\right) \overset{c}{\approx} \left((s_i)_{i\in t}, w_s, (m_i)_{i\in T}, (view_i)_{i\in[T]}\right),$$

where recall that for $i \in T$:

- m_i is the share for party i of the GKR message in the real interaction (Fig. 2, Step 4) and $\tilde{m}_i$ is is the corresponding share of a random message, in the simulation (Fig. 3, Step 4).
- $\widetilde{input}_i$ is the input to party i in the MPC protocol, derived by the simulator in Fig. 3, Step 5d. That is, $\widetilde{input}_i = \left(\widehat{G(s_i)}(z), \tilde{m}_i\right)$, for $i \neq k$ and $\widetilde{input}_k = \left(\widehat{\tilde{w}_s}(z), \tilde{m}_k\right)$.
- $\widetilde{view}_i$ consists of the input for party i followed by the output of S_{MPC} for the party i. Thus, $\left\{(\widetilde{view}_i)\right\}_{i\in[T]} = \left((\widetilde{input}_i)_{i\in T}, S_{MPC}\left(\widetilde{input}_i\right)_{i\in T}\right)$.
- $input_i$ is the input to party i in the MPC protocol in the real interation, Fig. 2, Step 5c. Thus, $input_i = \left(s(\widehat{G(s_i)}(z), m_i\right)$ for $i \neq k$, and $input_k = \left(\widehat{w_s}(z), m_k\right)$.

– $view_i$ is $input_i$ followed by the view of that party in the MPC protocol (all in the real interaction).

We first show the distribution of the inputs and views of the parties in T are computationally indistinguishable in the two cases. We then conclude that the commitment and decommitment to those distributions are also computationally indistinguishable. Finally, we show we can add to those distributions the commitment of the remaining party and $coins_{V^*}$ sent by V^* in Step 4 and the claim follows.

The proof is via a hybrid argument. Consider the following hybrid distributions:

$$
\begin{aligned}
H_0 &:= \left((s_i)_{i\in T}, \tilde{w}_s, (\tilde{m}_i)_{i\in T}, \left\{\widetilde{(view_i)}\right\}_{i\in[T]}\right) \\
&= \left((s_i)_{i\in T}, \tilde{w}_s, (\tilde{m}_i)_{i\in T}, (\widetilde{(input_i)}_{i\in T}, S_{MPC}(\widetilde{(input_i)}_{i\in T})\right) \\
H_1 &:= \left((s_i)_{i\in T}, w_s, (\tilde{m}_i)_{i\in T}, (\overline{(input_i)}_{i\in T}, S_{MPC}(\overline{(input_i)}_{i\in T})\right), \\
&\quad \text{where for } i \neq k, \overline{(input_i)} = \left(\widehat{G(s_i)}(z)), \tilde{m}_i\right), \text{and } \overline{(input_k)} = \left(\widehat{w_s}(z)), \tilde{m}_k\right) \\
H_2 &:= \left((s_i)_{i\in T}, w_s, (m_i)_{i\in T}, ((input_i)_{i\in T}, S_{MPC}((input_i)_{i\in T})\right), \\
H_3 &:= \left((s_i)_{i\in T}, w_s, (m_i)_{i\in T}, \left\{(view_i)\right\}_{i\in[T]}\right).
\end{aligned}
$$

$H_0 \overset{c}{\approx} H_1$*:* We first show that $(s_2, \ldots, s_{k-1}, \tilde{w}_s) \overset{c}{\approx} (s_2, \ldots, s_{k-1}, w_s)$. Assume there exists a distinguisher D that distinguishes between $(s_2, \ldots, s_{k-1}, \tilde{w}_s)$ and $(s_2, \ldots, s_{k-1}, w_s)$ with non-negligible advantage. Recall that $\tilde{w}_s = \big(\tilde{w} \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1})\big)$ and $w_s = \big(w \oplus G(s_1) \oplus \cdots \oplus G(s_{k-1})\big)$.

We construct D' that distinguishes between U_n and $G(U_{|\lambda|})$. We give D' the non-uniform advice (x, w). The distinguisher D', given as input $r \in \{0,1\}^m$, generates $s_2, \ldots, s_{k-1} \in \{0,1\}^{\{|\lambda|\}}$, then computes $G(s_2) \ldots G(s_{k-1})$ and runs D on input $(s_2 \ldots, s_{k-1}, \bar{w}_s)$, where $\bar{w}_s = r \oplus (w \oplus G(s_2) \oplus \cdots \oplus G(s_{k-1}))$ and outputs the result. If r is sampled from U_n, then $\bar{w}_s$ will also be a random element of U_n and thus $\bar{w}_s$ will be of the same distribution as $\tilde{w}_s$. On the other hand, if r is sampled from $G(U_{|\lambda|})$ then $\bar{w}_s$ is the same distribution as w_s. So D' will be able to distinguish with the same probability as D, in contradiction to the pseudorandomness of G.

Observe that H_0 and H_1 are obtained from the two distributions above via the same procedure. Namely, $(\widetilde{input_i})_{i\in T}$ and $(\overline{input_i})_{i\in T}$ are computed by applying G on the seeds and computing the multi-linear extension at the point z, followed by, either $\widehat{\tilde{w}_s}(z), (\tilde{m}_i)_{i\in T}$ or $\widehat{w_s}(z), (\tilde{m}_i)_{i\in T}$. Then, in the same way run S_{MPC} on $(\widetilde{input_i})_{i\in T}$ and $(\overline{input_i})_{i\in T}$). Hence, using Theorem 8, since G, the multi-linear extension and S_{MPC} are PPT algorithms, we conclude that $H_0 \overset{c}{\approx} H_1$.

$H_1 \equiv H_2$*:* Recall $(m_i)_{i\in[k]}$ is the distribution of the additive secret shaing of the GKR messages as in Fig. 2, Step 4, whereas $(\tilde{m}_i)_{i\in[k]}$ is the distribution of a secret sharing of a random message in Fig. 3, Step 4.

Since the restriction of an additive secret sharing to any set of $k-1$ shares is uniformly random, it follows that $(m_i)_{i\in T}$ is distributed identically to $(\tilde{m}_i)_{i\in T}$.

Thus, since the rest of the distributions are simply computed in the same manner for both distributions similarly to the previous case, we conclude $H_1 \equiv H_2$.

$H_2 \equiv H_3$ By the $(k-1)$-privacy of the MPC protocol (see Definition 22 and Theorem 23), it holds that $\{(view_i)\}_{i\in[T]}$ is distributed identically to $S_{MPC}((input_i)_{i\in T})$. Thus, since the input and the rest of the hybrids are identical in the two cases, we have that $H_2 \equiv H_3$.

Thus, we conclude that $H_0 \overset{c}{\approx} H_3$. Denote:

$$C_0 := \left(com(s_2), \ldots \tilde{w}_{com}, com(s_k), com(\tilde{m}_2) \ldots com(\tilde{m}_k), \left\{com(\widetilde{view}_i)\right\}_{i\in[T]}, \overline{dec}\right)$$

$$C_1 := \left(com(s_2), \ldots w_{com}, com(s_k), com(m_2) \ldots com(m_k), \left\{com(view_i)\right\}_{i\in[T]}, \overline{dec}\right),$$

where $\overline{dec} = \{dec_S(i), dec_m(i), dec_v(i)\}_{i\in T}$ and $dec_s(i)$, $dec_m(i)$, $dec_v(i)$ are the decommitments to s_i, $\tilde{m}_i$ and party i's view, respectively.

Observe that C_0 (resp., C_1) is computed from H_0 (resp., H_3) by the same procedure – namely, committing to the seeds, $(\tilde{m}_i)_{i\in T}$ (resp., $(m_i)_{i\in T}$) and $(\widetilde{view}_i)_{i\in T}$ (resp., $(view_i)_{i\in T}$), generating a random seed s_k and computing $\tilde{w}_{com} = \tilde{w}_s \oplus G(s_k)$ (resp., $w_{com} = w_s \oplus G(s_k)$). Thus, by Theorem 8, we conclude that $C_0 \overset{c}{\approx} C_1$.

Finally, from Theorem 20 we can add the commitment to the input of the remaining party $j = 1$, i.e.:

$$(C_0, com(s_1), com(\tilde{m}_1), com(\widetilde{view}_1)) \overset{c}{\approx} (C_1, com(s_1), com(m_1), com(view_1)).$$

The remaining difference between the claim and what we have proved are $\mathsf{tr}_{\mathsf{GKR}}$ and $\widetilde{\mathsf{tr}_{\mathsf{GKR}}}$. Recall that $\mathsf{tr}_{\mathsf{GKR}} = (com(m_1), \ldots, com(m_k), coins_{V^*})$, and $\widetilde{\mathsf{tr}_{\mathsf{GKR}}} = (com(\tilde{m}_1), \ldots, com(\tilde{m}_k), coins_{V^*})$, where $coins_{V^*}$ are the strings sent by V^* in step 4. From Section 4.1, the interaction up to the opening of the commitment is computationally indistinguishable between the interaction with P and $\bar{S}$. Therefore, $coins_{V^*}$ are also indistinguishable between the interaction of V^* with P and $\bar{S}$. Hence, we can add $coins_{V^*}$ to the distributions, and the claim follows.

Acknowledgements

We thank Yuval Ishai for useful discussions, and the anonymous reviewers insightful comments.

Noor Athamnah and Ron Rothblum are funded by the European Union (ERC, FASTPROOF, 101041208). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

BFLS91. László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31. ACM, 1991. 8

Blu86. Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986. 2

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011. 2

CLTV15. Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2015. 2

DI06. Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 501–520. Springer, 2006. 9

Gen09. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, USA, 2009. 2

GGI+15. Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interactive zero-knowledge proofs. *J. Cryptol.*, 28(4):820–843, 2015. 2, 4, 5, 9

GGM86. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986. 6, 13

GH98. Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998. 3, 4

GKR15. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for Muggles. *J. ACM*, 62(4):27:1–27:64, 2015. 2, 5, 6, 7, 9, 11, 12

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

GMW86. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 1986. 1, 2

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on*

Theory of Computing, 1987, New York, New York, USA, pages 218–229. ACM, 1987. 9, 15

Gol01. Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. 6, 13, 21, 22

GR17. Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 39:1–39:43. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 8

GVW02. Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Comput. Complex.*, 11(1-2):1–53, 2002. 3, 4

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. 6, 14

HLR21. Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 750–760. ACM, 2021. 2

HN24. Shuichi Hirahara and Mikito Nanashima. One-way functions and zero knowledge, 2024. 2, 3

HVW23. Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, and Mor Weiss. Beyond MPC-in-the-head: Black-box constructions of short zero-knowledge proofs. In Guy N. Rothblum and Hoeteck Wee, editors, *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part I*, volume 14369 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2023. 2, 5, 7, 18

IKOS09. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009. 2, 4, 5, 9, 14, 18

IP99. Russell Impagliazzo and Ramamohan Paturi. Complexity of k-SAT. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240. IEEE Computer Society, 1999. 3

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732. ACM, 1992. 4

KR08. Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008. 2

KRV24. Or Keret, Ron D. Rothblum, and Prashant Nalini Vasudevan. Doubly-efficient batch verification in statistical zero-knowledge. *IACR Cryptol. ePrint Arch.*, page 781, 2024. 5

MV24. Daniele Micciancio and Vinod Vaikuntanathan. Sok: Learning with errors, circular security, and fully homomorphic encryption. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part IV*, volume 14604 of *Lecture Notes in Computer Science*, pages 291–321. Springer, 2024. 2

Nao91. Moni Naor. Bit commitment using pseudorandomness. *J. Cryptol.*, 4(2):151–158, 1991. 14

NR22. Shafik Nassar and Ron D. Rothblum. Succinct interactive oracle proofs: Applications and limitations. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 504–532. Springer, 2022. 2, 3, 4, 5

OW93. Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*, pages 3–17. IEEE Computer Society, 1993. 2, 3

Ros12. Mike Rosulek. Must you know the code of f to securely compute f? In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 87–104. Springer, 2012. 7

RR20. Noga Ron-Zewi and Ron D. Rothblum. Local proofs approaching the witness length. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 846–857. IEEE, 2020. 3

RRR21. Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021. 2, 6, 9, 12

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il